



# Azure Guidebook:

## Building a detection and response strategy



# Contents

|  |    |
|--|----|
| Introduction   | 3  |
| So you've got some workloads in Azure ...                                  | 3  |
| Sources of security signal in Microsoft Azure                              | 4  |
| Azure Security Center (ASC) .....  | 5  |
| Azure Active Directory (AD) Identity Protection .....                      | 6  |
| Microsoft Defender for Identity .....                                      | 7  |
| Azure Sentinel .....   | 8  |
| Microsoft Cloud Application Security (MCAS) .....                          | 10 |
| Sources of categorical log data in Azure                                   | 11 |
| Activity Logs .....  | 11 |
| Resource Logs .....  | 12 |
| Azure Active Directory (AD) Logs .....                                     | 13 |
| Getting the most out of Azure  | 15 |
| Which security-specific alerts should I enable? .....                      | 15 |
| Which detection use cases should I start with? .....                       | 15 |
| What should I enable to detect suspicious authentication attempts? .....   | 16 |
| What should I enable to detect suspicious management plane activity? ..... | 17 |
| What data should my analysts use to enhance Azure alerts? .....            | 18 |
| And one bonus tip .....  | 19 |
| How to start building a detection and response strategy in Azure           | 22 |
| Three things to remember   | 23 |

## Introduction

### **So your organization uses lots of Microsoft tools, and now you've got a few (or a lot of) workloads running in Microsoft Azure.**

#### **Now what?**

As great as it would be to live in a world where security monitoring is a matter of flipping a big switch that says "SECURITY" in bold letters ... we definitely don't live in that world. Because when it comes to any cloud provider, including Azure, getting the right signal to security analysts isn't as cut and dry as you'd hope.

But that doesn't mean it's impossible, either.

You just have to understand which sources of security signal to enable and how to make use of them in a way that empowers your SOC analysts to make quick, sound decisions.

That's exactly where we're hoping we can help.

We know that sorting through Azure's hundreds of services (and the alerts and logs associated with

them) is no easy feat, especially if you're still getting to know the platform.

#### **After reading this guide, you'll have a better understanding of:**

- The available sources of logging and alert data in Azure;
- How (and why and when) to use those logs;
- Other types of logging you'll need to pair with those security signals to set your analysts up for success; and
- A few of the lessons we've learned setting up Azure security signal (Hint: You can use these to inform and tweak your own security monitoring activities!).

---

## So you've got some workloads in Azure ...

### **... and you need to get some security signal to protect them.**

Many of our customers use Azure as their primary infrastructure provider. Over time we've seen various approaches to logging and alerting in Azure and we've got some opinions on which log and alerting sources are most useful.

Let's start with understanding the various security services in Azure. It can be overwhelming. So let's start with a TL;DR about how each security-specific alert source works, along with a couple points you should consider when deciding whether that product is pulling the data *your* SOC analysts need to do their jobs.

## Sources of security signal in Microsoft Azure

Before we dive into the details of each data source, here's a quick reference for what context you'll find from each Azure alert or log type.

| Microsoft data source                              | What it is                                   | What it does  |
|--|--|---|
| <b>Azure Security Center (ASC)</b>                 | Security-specific alerting                   | Alerts on management and data plane activity in Azure   |
| <b>Azure Active Directory Identity Protection</b>  | Security-specific alerting                   | Flags "risky sign-ins" and "risky users" in Azure and O365 Applications                                       |
| <b>Microsoft Defender for Identity</b>             | Security-specific alerting                   | Alerts on suspicious activity related to on-premise Domain Controllers and all things Active Directory        |
| <b>Azure Sentinel</b>                              | Cloud-native SIEM                            | With the ability to ingest log and alert signal from both inside and outside of your Azure environment        |
| <b>Microsoft Cloud Application Security (MCAS)</b> | Cloud Access Security Broker (CASB) solution | Generates security specific alerts, enforces usage policies and automates remediation actions                 |
| <b>Azure Activity Logs</b>                         | A type of platform log                       | Logs management plane activities in Azure   |
| <b>Azure Resource Logs</b>                         | A type of platform log                       | Logs data plane activities in Azure   |
| <b>Azure Active Directory Audit Logs</b>           | A type of platform log                       | Tracks identity and access management changes as well as changes to Azure Enterprise Applications             |
| <b>Azure Active Directory Sign-in Logs</b>         | A type of platform log                       | Tracks metadata about sign-in events by security principals   |
| <b>Azure Active Directory Provisioning Logs</b>    | A type of platform log                       | Tracks actions related to granting or revoking access to SaaS applications for a given Azure AD user or group |



## Azure Security Center (ASC)

### TL;DR:

A single pane of glass for alerts

### What is it?

ASC offers detections for both the management and data plane in Azure. It uses Azure Defender agents that are deployed to specific resources within your Azure environment. While enabling Azure Defender in your environment adds additional cost, it also offers a comprehensive and predictable source of alerting that you can't easily replicate with activity and resource logs alone.

### How does it work?

Think of Azure Security Center as the console where you can view alerts generated by Azure Defender agents as well as alerts from non-Azure resources. Azure Defender agents are deployed to Azure-hosted resources with a single on/off slider for a given resource type in a given subscription. You can also use ASC to ingest data and security alerts from [on-prem hosts](#), [AWS Security Hub](#) and [GCP Security Command Center](#) if you have a hybrid environment.

### What type of signal and detection does it offer?

ASC supports [detection use cases](#) for both management and data plane activity. Most of these alerts detect activity patterns that are “anomalous” in some way although there are more categorical detections present.

ASC is a foundational alert source for Azure environments because it covers the core resource types that exist in most environments regardless of how they're architected – like Azure Storage, Azure Resource Manager and Azure App Service.

### EXPEL PRO TIP

When you're deciding which Azure Defender agents to enable across the subscriptions and resources in your environment, keep in mind the [costs associated with each type of agent](#). Also, remember that Azure Defender agents are deployed for each resource type in a given subscription. That means if you have two subscriptions in your Azure environment and you'd like to generate ASC alerts for all Key Vault resources in both, you'll need to [flip the Key Vaults switch](#) for each subscription individually.

You can also [auto-provision Azure Defender agents](#) through Azure Security Center if you want to make sure that all new resources of a given type are generating ASC alerts by default.



## Azure Active Directory (AD) Identity Protection

### TL;DR:

User behavior analytics (UEBA)

### What is it?

Azure AD Identity Protection focuses on “risky” authentication attempts mediated by Azure Active Directory. It generates [alerts in the form of three distinct “reports”](#): — risky users, risky sign-ins and risky detections.

### How does it work?

Unlike ASC where you just need to deploy Azure Defender agents to start generating alerts from a given resource, Azure AD Identity Protection [requires an Azure Active Directory Premium P2 license](#) in order to generate alerts with the proper context you’ll need for triage.

The P1 and free licenses for Azure AD will still generate risk events for risky sign-ins and risky users, but they won’t have the context that SOC analysts rely on to take action.

Also, licenses lower than P2 don’t allow administrators to configure risk policies that automate the remediation of risky activity.

### What type of signal and detection does it offer?

While “risk” can mean different things to different people, Microsoft provides its own list of behaviors this product is trying to detect out of the box and groups them into [user risk, sign-in risk and “other risk detections.”](#)

If you’ve got Identity Protection enabled, admins can turn on [policies that automate remediation actions](#) in response to a risk event. For example, when a risky login occurs you can automatically lock out the account or force a password reset.

At Expel, we collect all of these risk events via API and then use our own rules to prioritize which ones to escalate to our SOC analysts. With the help of our automated workflows, we enrich each Identity Protection event with historical authentication data for the target user and source IP to shortcut triage steps that would normally require our analysts to manually run queries on their own.

### ✕ EXPEL PRO TIP

For orgs with a large user base, the 50 percent increase in price per user (from \$6 to \$9, at the time we wrote this) [when switching from P1 to P2 licenses for Azure AD](#) can add up very quickly. It’s important to keep this in mind when deciding whether you want to make Azure AD Identity Protection alerts a core part of your monitoring strategy.

A decorative graphic consisting of several parallel blue diagonal lines on the left side of the page.

## Microsoft Defender for Identity

### TL;DR:

UEBA for on-prem log sources

### What is it?

Microsoft Defender for Identity is a bit of an outlier in this guide since it's not actually geared towards detecting malicious activity in Azure itself. However, we've noticed some confusion around this. Specifically, we've seen a few organizations purchase Defender for Identity licenses expecting it to provide security value for their cloud environment. It doesn't.

Microsoft Defender for Identity sits on on-premise Domain Controllers to detect internal reconnaissance activity, account compromises, lateral movement and attempts to compromise the domain controller itself.

There can be overlap between cloud and on-premise activity in hybrid environments with Active Directory Federation Services (ADFS), so the confusion over exactly which portions of an environment Microsoft Defender for Identity protects is understandable.

### How does it work?

Why are we discussing a product that monitors on-premise domain controllers and Active Directory activity in a guidebook about Azure?

In short, because it integrates natively with Microsoft Cloud App Security (MCAS) and so many organizations assume that it's directly related to "the cloud." In reality, the "relationship" to the cloud basically stops and starts with the ability to forward Microsoft Defender for Identity alerts to the MCAS portal.

If you're attempting to make sense of the Azure-related monitoring product suite, Microsoft Defender for Identity can leave you scratching your head. Some of the confusion stems from the difference between Defender for Identity and Azure AD Identity Protection.

The other point of confusion stems from the fact that Microsoft Defender for Identity previously had two other names throughout its life: Microsoft Advanced Threat Analytics and Azure Advanced Threat Protection.

Both of these prior iterations of the product were also based on sensors for on-premise domain controllers, but the latter product moved the storage of alert and event data the sensor collects to the cloud (hence the addition of Azure to the product name).

### What type of signal and detection does it offer?

Microsoft Defender for Identity overlaps with Azure AD Identity Protection and also fulfills some authentication-related detection use cases.

However, [it goes far beyond detecting suspicious authentication attempts](#) and collects a wide array of signals and events from your on-premise domain controllers to generate security alerts.

*Microsoft Defender for Identity continued ...*

We're not going to explore these alerts in detail since they aren't related to Azure, but they're divided into a few straightforward categories: reconnaissance, compromised credentials, lateral movement and domain dominance.

### **✘ EXPEL PRO TIP**

Make sure you understand exactly what Microsoft Defender for Identity is protecting. It's easy to confuse these capabilities with Azure AD Identity Protection, but they're quite different.



## **Azure Sentinel**

**TL;DR:**  
SIEM alerting

### **What is it?**

Azure Sentinel is a cloud-native SIEM that ingests data from inside and outside of your Azure environment. In addition to ingesting alerts from Azure Security Center and lower-level event data from various types of platform logs, Microsoft [built out a number of connectors](#) for third-party products.

### **How does it work?**

Under the hood, an Azure Sentinel workspace is essentially an Azure Log Analytics (ALA) workspace with the ability to connect non-Azure data sources and the ability to write Microsoft's Kusto Query Language (KQL) detection rules.

Once you've connected the data sources you want Sentinel to use, you can [query that data using KQL](#) and write queries that add logic to generate your own alerts.

### **What type of signal and detection does it offer?**

Like a traditional SIEM, you can structure your rules to respond to specific alerts or correlate specific alerts and event data for more high-fidelity detections.

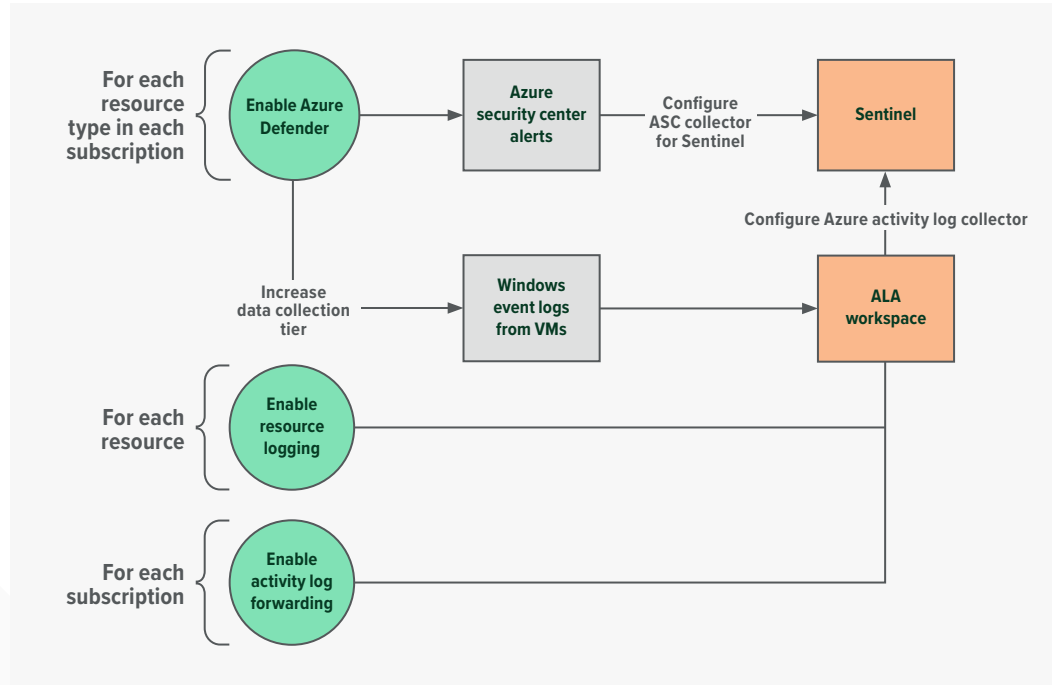
Sentinel does come with a few out-of-the-box rules, but its real power comes from the ability to write your own rules and use KQL for investigations.

### **✘ EXPEL PRO TIP**

The biggest "gotcha" related to Sentinel is the way that storage costs can balloon if you aren't careful about what you decide to forward to Sentinel.



Azure Sentinel  
continued ...



Data paths from log and alert sources via ALA to Sentinel

As an example, let's say you want to send a record of every SQL query run against an Azure-hosted SQL database. You can [enable resource logging for the database or server](#) in question and forward that to a Log Analytics Workspace (which will incur [storage and retention costs](#)). Then, you'll need to set up a connector in Azure Sentinel to ingest data from that Log Analytics Workspace (which [also adds additional storage costs](#)). You get the idea. If you're not doing all of this with a specific detection use case in mind, you can quickly end up spending a lot of money with little return on that investment.



## Microsoft Cloud Application Security (MCAS)

**TL;DR:**  
CASB for MS  
Cloud Services

### What is it?

MCAS is Microsoft's cloud access security broker (CASB) product offering. Like a lot of CASB offerings, the use cases are broad and diverse.

### How does it work?

While there's certainly security alerting functionality in MCAS, it also functions as a way to discover cloud applications and infrastructure in your environment so you can create policies to manage them. Administrators must [configure Cloud Discovery](#) by connecting some form of "traffic logs" that MCAS uses to discover which cloud apps are being used by employees.

Once admins understand how cloud applications are being used in their environment, they can create policies that define what authorized usage looks like and even automate some remediation steps for users or hosts that violate those policies.

### What type of signal and detection does it offer?

MCAS lets admins create policies that generate alerts for various patterns of user activity. It comes with a bunch of [out-of-the-box detections](#) focused on anomalous access of cloud apps and [anomalous activity patterns](#).

In addition to generating its own alerts, MCAS can also ingest alerts from other Microsoft products. While it's not a full-fledged SIEM like Azure Sentinel, you can configure it to ingest alerts from Azure AD Identity Protection and Microsoft Defender for Identity, which is a nice perk.

### EXPEL PRO TIP

[Licenses for MCAS](#) are included with a bunch of Microsoft's products. So, depending on which license you choose, different combinations of functionality are likely already available to you.

It's important to note that both ASC and MCAS can provide recommendations about misconfigurations in Azure infrastructure and SaaS applications, optionally pinned to specific compliance framework requirements. While this guide is focused on the types of security alerts that these products generate, many organizations find this security recommendation functionality extremely valuable.

## Sources of categorical log data in Azure

In addition to the products that generate security-specific alerts, Azure provides three types of [“platform logs”](#) that capture lower-level categorical events. These logs can fill in additional details for the alerts that other Azure products generate so that analysts can make quicker decisions. You can also use them to build your own security alerting in Azure Sentinel or elsewhere.

If you go down the path of building your own detections based on the logs, you can cover some interesting use cases. But go in with eyes wide open because it can be a long slog to get to the point where homebrewed detections rival what you get from some Azure products.

Regardless of whether you build your own detections or not, you’ll want to enable some subset of the platform logs so your analysts can triage security-specific alerts.

### Now let’s dive into the details of those platform logs.

#### Activity Logs

##### What are they?

Activity logs provide insight into management plane activities in Azure environments. Management plane activities refer to changes to Azure resources themselves, rather than activity that takes place on Virtual Machines (VM).

Here’s an example. If you create a new VM or [make changes to a service certificate for a VM](#) it could generate activity logs. But anything that happens at the OS level for that VM would be considered “data plane” activity and Azure Activity Log wouldn’t monitor it.

For some resource types, what’s considered “management plane” activity can be a bit fuzzy. A good example is the Azure Key Vault service. Despite the fact that the entire functional purpose of a Key Vault is to store secrets, the Activity Log records when users retrieve a secret.

##### How do you enable them?

These logs are enabled by default but if you want to query them using KQL you’ll need to send them to a Log Analytics Workspace. You can [configure this using the Activity log diagnostic settings](#). It’s also possible to store these logs as Azure Storage objects or store them in Azure Event Hub.

Any of these three options for sending activity logs to Azure Sentinel will let you enrich detection logic and do all your investigation work in one place. If you don’t need to query them with KQL or forward them to Azure Sentinel, you can keep the default configuration and view them in the Activity Log view in the Azure Portal.

Activity logs  
continued ...

You can keep the default configuration and view them in the Activity Log view in the Azure Portal.

### EXPEL PRO TIP

Each subscription has its own activity log, so be mindful of display options and the roles assigned to your Azure AD user account when you're trying to track down activity logs in your environment.

At Expel, we abstract away subscription-based logging segmentation so our analysts can simultaneously run their own KQL queries across every ALA workspace directly from the Expel Workbench™. However, this abstraction is mostly useful if your Azure environment is sending activity logs to multiple ALA workspaces.

## Resource Logs

### What are they?

Resource logs monitor data plane activity in Azure.

Using the VM example, resource logs contain information about actions “inside” the VM at the OS level. This could include data like Windows Event Logs, Linux audited events or remote connections to the VM.

For a VM, this requires installing [one of a number of agents](#) depending on where you want to send the logs. Different agents and collection methods provide different levels of details and logging use cases. It can be a bit overwhelming to choose one collection method or the other, but focusing on which information is currently missing from security specific alerts can help you decide what you need to collect. As you begin to handle those security specific alerts, you will also begin to get a clearer picture of where you need to forward resource logs so that they can be easily used during triage.

### How do you enable them?

Since data plane actions can look drastically different for different types of Azure resources, you've got to pay close attention to Microsoft's documentation for [how to configure resource logging for each type](#) and the [schemas for the resulting resource logs](#).

It shouldn't come as a surprise that these kinds of low-level, categorical events that resource logging focuses on can be noisy.

Resource logs  
continued ...

**The best way to control these costs is to only enable resource logging with specific use cases in mind.**

### **✕ EXPEL PRO TIP**

If you don't want to get a big surprise when your Azure bill arrives, you'll want to be thoughtful (and do some testing) to figure out which activity logs and security alerts are worth sending to Azure Sentinel.

Storage costs are the big driver. The best way to control these costs is to only enable resource logging with specific use cases in mind. Since resource logs are unlikely to have a one-to-one relationship with security incidents, your main use cases for resource logs in the context of security operations will likely be investigation support.



## **Azure Active Directory (AD) Logs**

### **What are they?**

Azure AD logs come in the form of reports and track two main categories of activity: security reports and activity reports. When trying to decipher what kind of monitoring is available for Azure AD activity, these two categories can be a bit confusing because "[security reports](#)" actually refers to the logging capabilities of Azure AD Identity Protection.

What's up with that? Well, it likely stems from the fact that Identity Protection is tied to the license for Azure AD and priced per user. So even though it's marketed as a separate product for security-specific alerting, it's technically bundled into the license level for Azure AD itself.

Since we're talking about platform logs right now, let's focus on logging done under the auspice of "activity reports."

When it comes to activity reports, the logs fall into three categories: [audit logs](#), [sign-in logs](#) and [provisioning logs](#).

- **Audit logs** provide insight into identity and access management changes like group membership and the creation or modification of policies (like multi-factor authentication requirements or automatic password resets). In addition to information about users, groups and the policies applied to them, audit logs record information about the addition and modification of [enterprise apps](#) to Azure AD.
- **Sign-in logs** are pretty straightforward. They cover the information you'd expect: a timestamp of user sign-ins, whether the login was successful, what application they logged into, their IP address and other common metadata.

Azure Active Directory  
(AD) Logs continued ...

- **Provisioning logs** track automated SaaS application user provisioning via Azure Active Directory. Simply put, provisioning allows you to [programmatically grant or revoke access to SaaS applications](#) for specific users and groups. While this is certainly useful for administrators to centralize identity and access management (IAM) tasks within Azure AD, we don't use these much at Expel.

### How do you enable them?

While all of the log types that make up activity reports (audit logs, sign-in logs and provisioning logs) are included in all Azure AD licenses, you need a Premium P1/P2 license for Azure AD to generate security reports.

Just like other platform logs, you can forward Azure AD audit, sign-in and provisioning logs to a number of different destinations, including a Log Analytics Workspace and Storage Containers.


### EXPEL PRO TIP

If you purchase one of the premium license tiers to get access to the security reports that Azure AD Identity Protection creates, make sure you understand what you're getting from the alerts before you write the check. Both licenses will tell you when it thinks sign-in (or a pattern of authentication attempts) are risky.

However, the P2 license generates events that contain more info about *why* a given authentication attempt or pattern of activity is considered risky. The additional context is a huge help to SOC analysts when they're trying to determine if a risky sign-in is, in fact, indicative of an account compromise.

## Getting the most out of Azure

Now that you know what logging and alert data you've got to work with, let's talk about what you can do with all that data.



### Which security-specific alerts should I enable?

This is a loaded question. The answer depends on what you want to do with the data.

You may have noticed that there are *a lot* of options when it comes to security alerting in Azure. But the alerts you get depend on what you've licensed and how you've configured it.


And the nature of those alerts vary based on licensing and configuration.

Even at a high level, we can already see the number of combinations of monitoring options you could opt for when you're building a detection and response strategy for your Azure environment.

**What your organization needs can be subjective. It depends on factors like:**

- Your threat model;
- What you're using Azure for;
- Your budget;
- Your ability to respond to alerts; and
- Your ability to tune the output from these products.

However you answer those questions, as you get started monitoring your Azure environment you'll want to lean on all of the research and development that's baked into Azure's security-specific alerting sources discussed above.



### Which detection use cases should I start with?

At a minimum you should prioritize alerting that gives you visibility into suspicious authentication attempts and suspicious management plane activities.

**From a threat modeling perspective, a large portion of attack techniques in Azure start with some form of account or credential compromise.**


Regardless of how an attacker compromises a given [security principal](#), you'll want to know when users sign-in from strange locations or when log-in attempts contain anomalous characteristics like a User-Agent string that's never been seen for that account.

*Which detection use cases should I start with? continued ...*

You'll also want to ensure that you can detect suspicious interaction with the Azure management plane in case an attacker successfully logs in with a legitimate service principal.

These two focus areas give a 1,000-foot view and a starting place that acts as a foundation for any detection and response strategy.

Now, let's look at how you can mature beyond this starting point.



## What should I enable to detect suspicious authentication attempts?

When it comes to suspicious authentication attempts, Azure AD Identity Protection is a good place to start. However, there are a couple of key points you're going to need to think through:

- Can you afford the additional cost per user associated with purchasing a P1 or P2 license for Azure Active Directory?

The per-user cost is applied to every single user you have in Active Directory, so the costs can quickly add up for larger organizations.

- Are your SOC analysts prepared to take meaningful action based on the activity that Azure AD Identity Protection generates?

Most of these detections are heuristic (versus categorical), so you'll need to dig into the lower-level sign-in and audit logs to figure out *why* Identity Protection flagged an authentication attempt as anomalous.

At Expel, we spend a lot of time automating the collection, analysis and presentation of things like historical login data for a user who has performed a "risky sign-in." Without this [additional automation](#), analysts will spend a lot of time running (and re-running) the same manual queries ad nauseum. That'll lead to burnout and you won't get consistent investigative outcomes. (For more on preventing burnout, check out our [blogs](#)).

Our analysts perform better (and are happier) when we present the data for them with useful callouts so they can focus on the critical thinking piece that ties it all together and render a verdict of what to do about each alert.

If you don't want to purchase the P1/P2 license necessary to enable Azure AD Identity Protection, you can always write your own detections based on Azure Active Directory sign-in logs.

But this is no small task and it'll require you to forward logs to either Sentinel or another data store that supports writing queries to identify unusual authentication attempts.




*What should I enable to detect suspicious authentication attempts? continued ...*

We've done this at Expel. By ingesting sign-in logs from Azure AD via API and writing our own logic for common detection use cases like impossible travel scenarios and password spraying, we've seen some impressive results. But this homebrewed strategy requires a lot of forethought, tuning, enrichment and a process to make it manageable for the SOC at scale.

This works for us because we can genericize common suspicious authentication patterns for sign-in logs outside of Azure AD (such as windows event logs and SSO products like Okta). If you were only trying to solve this problem for Azure AD log-ins, the amount of work and maintenance it requires wouldn't be feasible.

At most orgs, you can't implement a heuristic detection model that rivals the machine learning model that powers Azure AD Identity Protection. In this light, it may make sense to eat the cost of the premium licensing required to at least get yourself started.



## What should I enable to detect suspicious management plane activity?

Monitoring management plane activity is more complicated since it's not contained in a single Azure service in the same way that authentication activity is mediated by Azure Active Directory.

By definition, the management plane in Azure touches every aspect of the Azure platform. Since basically every aspect of Azure is accessible via API, separating legitimate and malicious usage of the same API endpoints can be tricky.

Activity Monitor logs are on by default and they're designed to record management plane activity. However, if you're trying to build out a library of detections using these logs alone, it can require some expert-level Azure knowledge, not to mention experience writing detection rules at scale.

If you're not quite ready to tackle that beast head on, ASC offers a quick shortcut that [covers many of the core Azure resource types](#) that form the backbone of most Azure environments.

You have to enable ASC separately for each type of resource in each subscription, so you'll want to take a more granular approach to deployment so it's aligned with how you use Azure.


For instance, if you're just hosting a web application via Azure App Service, you'd likely want to enable Azure Defender for App Service in the relevant subscription.

Additionally, you're likely to use a Key Vault to store X509 certificates, storage account keys and other secrets needed to manage and run your application. Your application will also likely need some kind of database, which uses some form of [Azure SQL service](#).

*What should I enable to detect suspicious management plane activity? continued ...*

All three of these core services have their own Azure Defender agents that you'll need to deploy in order to generate ASC alerts. Together they offer a great way to quickly generate out-of-the-box alerts for your web application and the other Azure services it uses.

The same principle applies for more complex environments. Regardless of which workloads you're hosting in Azure, an Azure Defender agent will likely cover some subset of those resources.



## What data should my analysts use to enhance Azure alerts?

Once you've got Azure security alerts flowing, you need to think critically about what you want your SOC analysts to do with the signal.

For ASC alerts, analysts need access to lower-level event data to effectively validate all of the heuristically oriented detection use cases they cover.

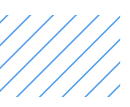
At a minimum, analysts need a way to quickly identify what "normal" behavior looks like for a given user. They also need a way to quickly zero in on suspicious authentication attempts that occur prior to a burst of suspicious management plane activity.

If analysts have both bits of context they can quickly answer two questions:

1. Was the sign-in that initiated the session suspicious or anomalous in some way?
2. Does this service principal have a history of conducting this type of management plane activity and/or interacting with the specific resources in question?

Analysts can answer the latter question by querying all platform logs associated with a given service principal or source IP address. We've found it helpful to programmatically summarize this information and present it to analysts in the form of investigative questions. We do this using Expel Workbench™.

Usually, when ASC generates alerts, it provides analysts with some description of *why* a given subset of activity is considered suspicious. It's a good start. But even with this information, analysts often still have to work backwards a bit to rule out false positives that are part and parcel of any machine learning-centric detection methodology.



## And one bonus tip

Communication is key.

The difference between a developer trying something new and an attacker doing something for the first time after they've compromised an account is, in many cases, nuanced. These nuances can make triage (and response) challenging – especially when there isn't enough context to determine if an observed activity is malicious outside of asking the user directly and them saying, “no that wasn't me and we don't have a reason to be doing that.”

Once your SOC analysts start handling ASC alerts, you'll quickly realize that post-triage communication is central to efficiently taking action on these alerts.

Let's look at a quick example of why communication is such an important part of the triage process for a large number of ASC alerts.

Suppose our analysts see an Azure Security Center alert for “Unusual operation pattern in a key vault.”

In a perfect world, analysts would be presented with:

- Some representation of the last two weeks of Activity Monitor Logs associated with the security principal indicated in the alert and;
- Aggregated metadata related to login activity for that user over the same period of time.

Armed with this additional data, it should be quite simple for analysts to answer questions like:

- Does this security principal normally interact with this key vault? Does it normally interact with this subscription?
- If it does interact with this key vault, is it normally the same kinds of actions that ASC considered anomalous?
- Were there any authentication attempts from unusual locations, using unusual user agents or any other change in metadata directly prior to the anomalous activity?
- Does this security principal have any indication of their job title in their AD object? Are they a member of any groups that would suggest they have a job role that involves interacting with core Azure infrastructure resources like a key vault?

Below is an example of the output from an automated workflow, which seeks to provide analysts with clearly formatted historical data about the kinds of actions that are common for a given user. Analysts see this info whenever there's an ASC alert that mentions an Azure AD user or service principal account name.

And one bonus tip continued ...

User listing

Last 7 Days of Azure Activity Monitor Events for [user@company.com](#)

Which Azure services has user@company.com interacted with?

| Azure Service Name    | Count |
|-----------------------|-------|
| Microsoft.DataFactory | 284   |

---

For each Azure service type, what actions does user@company.com usually take?

| Action   | Count |
|--|-------|
| Microsoft.DataFactory – Query Debug Pipeline Runs                                  | 200   |
| Microsoft.DataFactory – Get exposure control feature values for a list of features | 52    |
| Microsoft.DataFactory – Create Debug Pipeline Run Environment                      | 14    |
| Microsoft.DataFactory – Create Pipeline Run  | 10    |
| Microsoft.DataFactory – Get DataPlane access                                       | 8     |

---

When this user takes actions in Azure, what authentication method is usually associated with that activity?

| Authentication Type | Count |
|---------------------|-------|
| wia                 | 284   |

---

Which IP address is usually associated with activity sourced from user@company.com?

| IP Address | Count |
|------------|-------|
| 127.0.0.1  | 284   |

---

Which subscriptions and resource groups does user@company.com usually interact with?

| Subscription Name | Resource Group   | Count |
|-------------------|------------------|-------|
| Subscription-Name | resource-group-1 | 284   |

Expel Workbench™ Azure triage Ruxie™ workflow

All of these questions are necessary and useful to answer, but it's difficult to know for sure if this activity is unauthorized. This difficulty is especially pronounced as a managed detection and response (MDR) service provider, but even internal security teams will often need to track down the owner of that security principal and/or Azure AD application to truly determine if it was supposed to carry out those anomalous actions.

*And one bonus tip continued ...*

The SOC needs to be able to describe these three things in a way that's as simple and painless as possible for whoever is responsible for verifying whether the activity is authorized:

- The source and destination of these actions so they can find the correct employee/business unit to verify the activity;
- The activity that was considered anomalous so that the security principal owner can quickly say whether those actions were expected or not; and
- The potential impact of this activity if it were not authorized so that the security team can seek verification with the proper urgency.

If one of our customers comes back to us and tells us the suspicious activity isn't authorized, we re-classify the investigation as an incident and begin determining when the account compromise occurred and every action that was taken after that point. The end goal of this work is to paint a picture of what the attacker's goals may have been and the business impact.

Regardless of the exact nature of that post-compromise activity, it's critical that we quickly identify that a service principal was compromised in order to kick start our scoping process.

Clear and concise communication with the owner of that service principal or user account is key.

# How to start building a detection and response strategy in Azure

As you may have already pieced together, we're suggesting a phased approach to building a detection and response strategy for Azure.

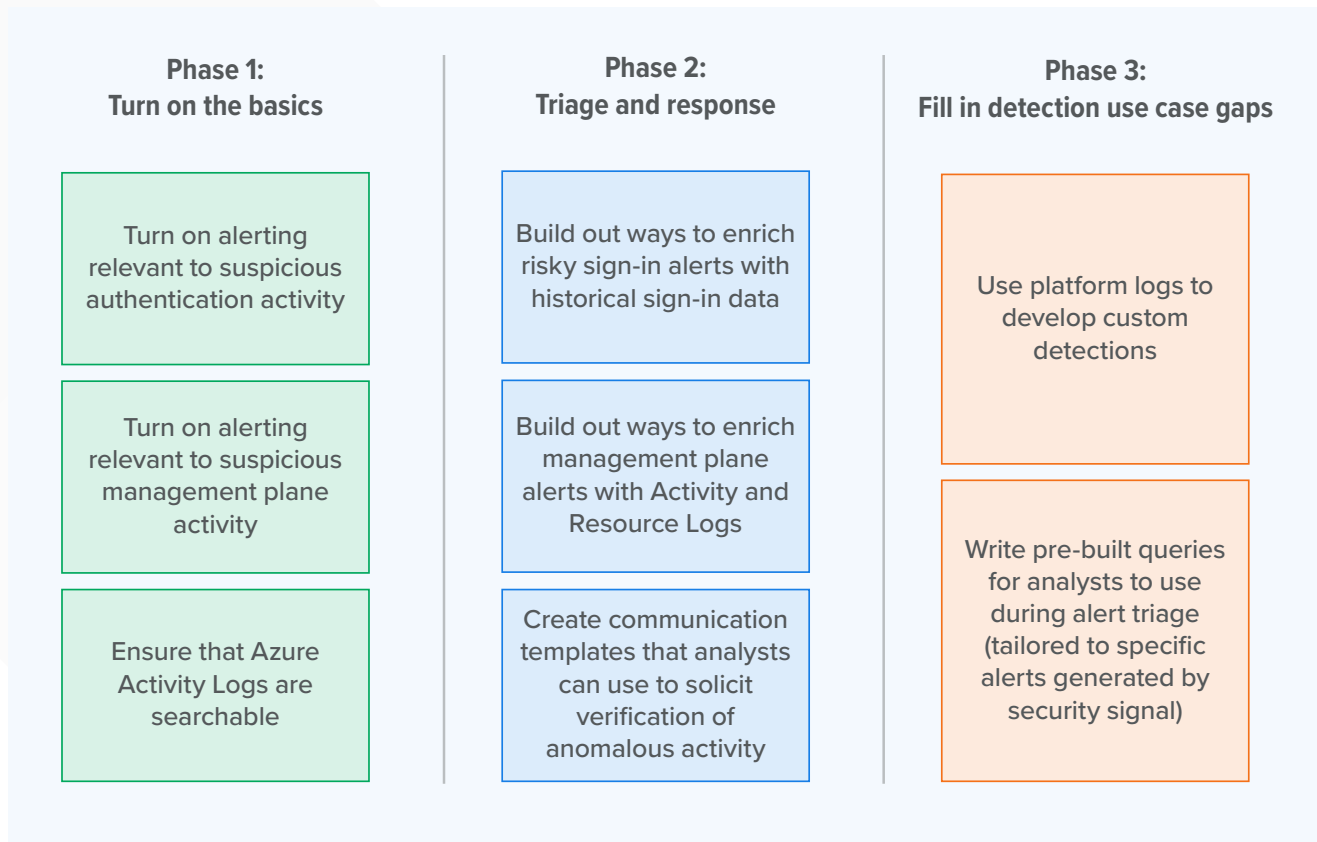
As with all things in security, a mature detection posture is built over time and evolves as you get more experience responding to alerts.

It's important to tie detection work to a clear

understanding of what alert triage and response looks like for all of the cool new stuff you've turned on.

As your organization becomes more comfortable with handling Azure alerts, you should focus your time on streamlining triage and response processes before attempting to build out your own set of custom detections.

**Below is our three-phased approach to beginning the process of building a detection and response strategy for Azure.**



*Expel's path to maturity for Azure detection and response strategy*

## Three things to remember

You're now armed with enough information to be dangerous. And while you may have to revisit this guidebook a few times as you develop your detection and response strategy for Azure, here are three key takeaways to keep in mind:

1

### Establish a goal

Focus on understanding what you're going to do with each type of alert once it reaches the SOC, before you enable and pay for every log and alert source available to you. This requires understanding what [investigative questions](#) SOC analysts should be asking for a given alert and equipping them with the tools to quickly find the data they need to answer those questions.

2

### Collaborate and communicate

Being collaborative and communicative with other teams can lead to speedy verification by account or resource owners and reduce the burden on analysts. Anomaly detections provided by ASC offer a good start to identifying when a pattern of activity is abnormal for a given security principal and needs verification.

3

### Consider costs

Costs can add up fast. Be careful with enabling platform logs at scale. When you want to enable a given type of platform log, you should be clear on what those logs will be used for and weigh the associated costs.



(this is the last page)

Our SOC-as-a-service capability offers 24x7 security monitoring and response for cloud, hybrid and on-premises environments. We use the security signals our customers already own so organizations can get more value from their existing security investments. We connect to customer tech remotely through APIs, not agents, so our SOC can start monitoring a customer's environment in a matter of hours, letting their internal teams get back to focusing on the most strategic security priorities that are unique to their business. Learn more at [www.expel.io](http://www.expel.io).