A WithSecure™ Consulting guide

WITH secure | Consulting

Microsoft Azure Security Framework

A roadmap for hardening the security of your Azure environment

Authors: Emilian Cebuc and Christian Philipov

Contents

Securing Azure

Although 95% of Fortune 500 companies use Microsoft Azure[1], there hasn't yet been a single, comprehensive guide for hardening the security of your Azure platform. In response, and inspired by Scott Piper's roadmap for building cloud security in AWS[2], this document provides the building blocks so you can start that journey.

No two organizations are the same. Each has a different technology infrastructure and security posture. Thus, when configuring your cloud environment, the security implications may not be immediately apparent in some cases. In others, you may not have the right expertise to begin building effective security controls. As a result, the content in this guide covers as many bases as possible, providing actionable best practices to help you secure your Azure environment.

How to use this guide

This is not a step-by-step manual for building specific security controls. Instead, it outlines the core principles of good security within Azure, providing both instructional and strategic guidance. The chapters are structured to help you resolve fundamental security issues first, before moving towards more complex, long-term remediations. **Our guidelines will help you build a defense-in-depth approach to securing your critical applications and the infrastructure that they rely on.**

The reference numbers throughout cite source material (referenced on page 21), including Microsoft documentation, for further reading and support.

Who it's for

Securing your cloud environment requires collaboration from a range of stakeholders, including but not limited to:

- CISOs and CIOs
- SOC analysts
- IT and security architects
- Cloud platform engineers
- Developers that produce software or projects hosted on Azure

Implementing our security framework within your organization will involve everyone listed above in some capacity.

Inventory management

Principle: understanding and logically grouping all resources avoids the growth of an unrecognized attack surface.

Correctly managing the resources available in each Azure environment is an unavoidable challenge in cloud security. If your organization develops new products or solutions, test environments and sandboxes used in development can remain in your cloud environment even after they are abandoned or become obsolete. No matter how much or how little you test in a development environment, obsolete or unaccounted resources drive up costs and constitute a hidden attack surface. This surface can even lead to the exposure of customers' personally identifiable information (PII)[3]. **It's essential to establish resource hygiene by regularly checking and monitoring the resources in your cloud environment. Maintain accountability over resources by ensuring they have a named owner**. In doing so, you can control your environment and restrict digital sprawl.

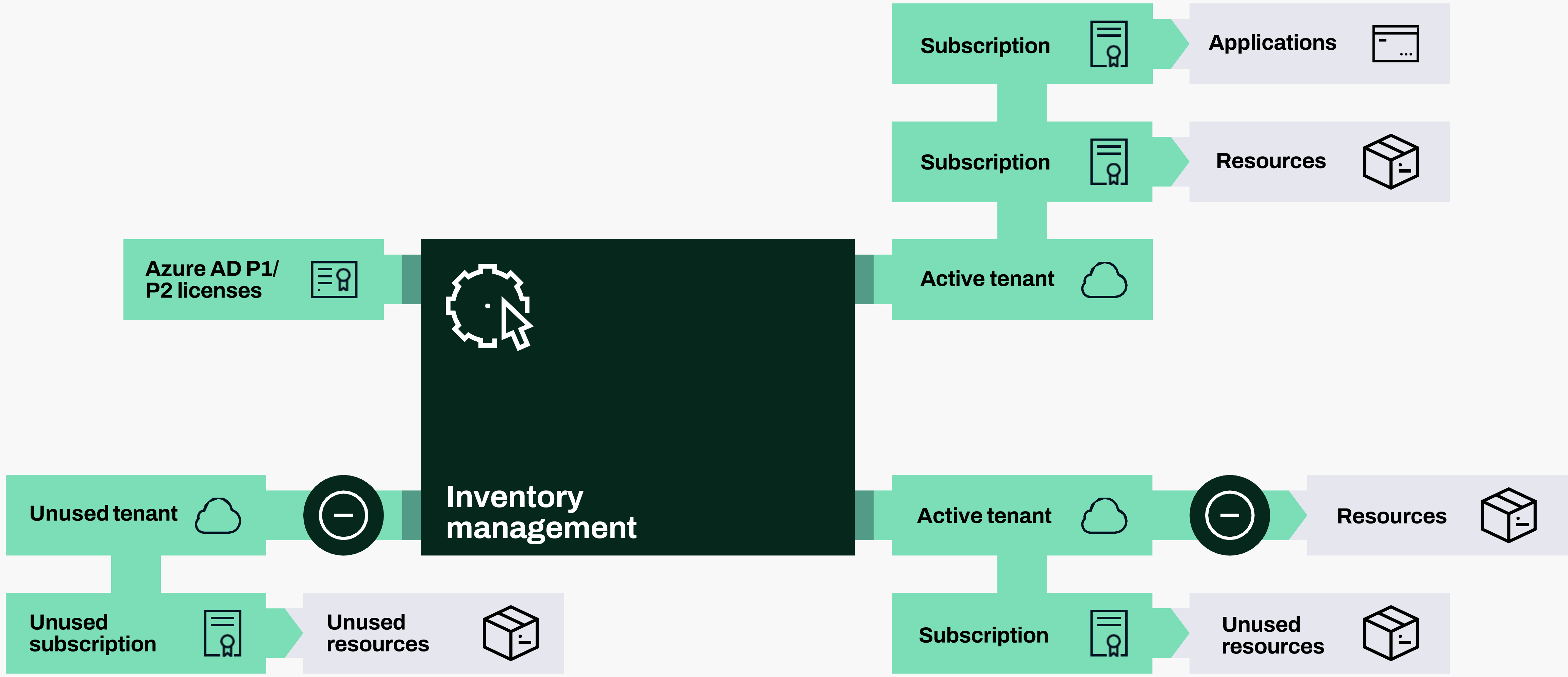


Fig. 1. Example of ineffective tenant management leading to unused resources

Understand Your Resources

Begin by reviewing all existing resources under your organization. Depending on the size of your cloud environment, this can be quite an undertaking. However, it is an essential first step — you can't protect what you don't know you have. At a minimum, try to establish what:

- Entra ID (Entra) tenants you have
- Subscriptions you have within your tenants
- Microsoft product licenses you utilize
- Applications your organization uses[4]

Only keep what you need

Remove any inactive subscriptions or trial subscriptions that have been created by users in error. Although it is important to test resources before development, this should only be done in a dedicated sandbox environment. This will avoid situations where excess subscriptions are created by team members and improperly utilized.

Group subscriptions with management groups to simplify governance

- Ensure there is a valid business case for each subscription
- Use tags[5] within subscriptions to identify the key owners, project names, and allocated cost center, at a minimum.[6] This guarantees a point of contact for any activity occurring in Azure in relation to a given project
- Use Azure Policy definitions to mandate that all new or current resources should have the requisite tags in order to be compliant[7]
- Use management groups[8] to group subscriptions into hierarchies
- Reduce the burden of assigning and managing required access to multiple separate environments by getting development teams to work on dedicated projects groups

These changes will create a well-organized set of management groups with subscriptions that are classified according to the relevant team or project.

Set up cost alerting for subscriptions

These should be based on the cost centers that have been allocated to each project and their budget.

- Restrict user permissions so that only certain users can create resources. (See the "policies" section for more information on configuring guardrails and user permissions).
- Set up an alert which is triggered when your costs reach a certain cap in a specific cost center
- Always investigate large and unexpected variations in average usage costs, as these can be an indicator of anomalous activity[9]

Resource isolation

Principle: isolation of resources helps reduce the “blast radius” of a cyber attack.

If one resource is compromised by an attack, other resources may be affected as a result. This is known as the ”blast radius”. It is a risk organizations must control if they wish to limit the impact of a compromise and it will vary from one instance to another. Measures can be as simple as establishing which data will be exposed if a storage account is made public, or as complex as deducing whether initial foothold on a virtual machine (VM) will allow an attacker to reach an instance hosting Jenkins in an entirely different subscription in a peer-joined network. The blast radius varies from case to case, and will therefore require dedication, resource, and time to address the risk. However, the impact of doing so can prove the difference between the loss of several isolated projects and a full environment takeover.

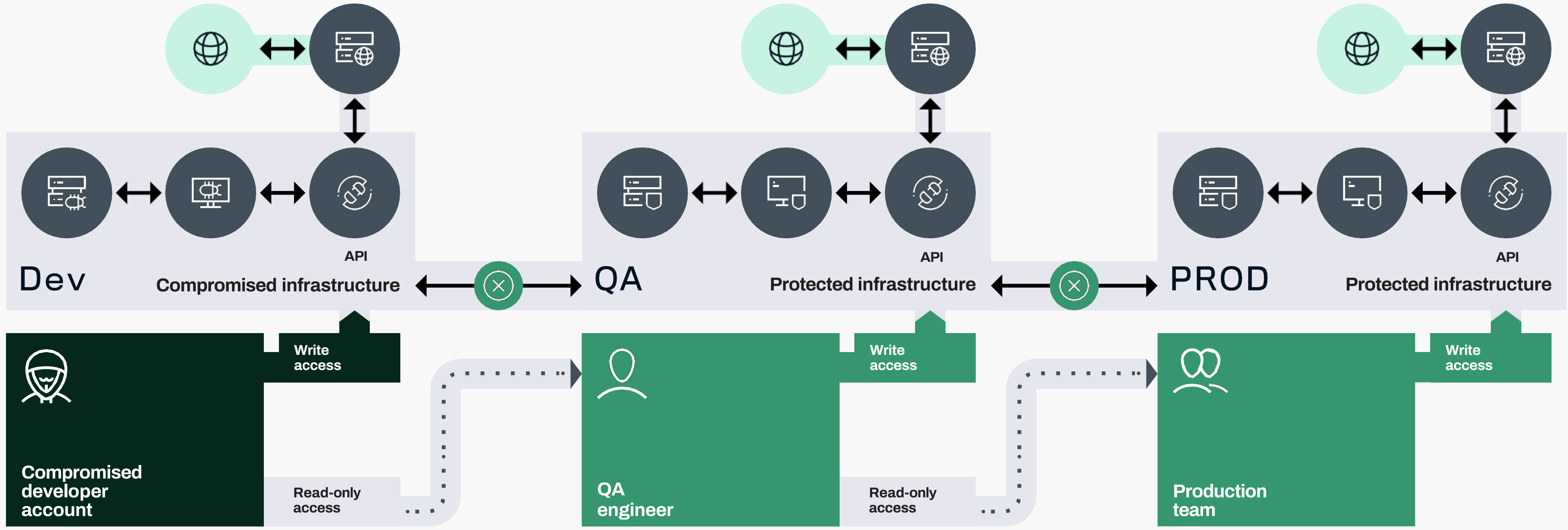


Fig. 2. An ideal resource isolation scenario, where an attacker’s lateral movement from a compromised developer account is prevented. Note: this is only an example and may not represent a realistic goal for your organization.

Enforce logical separation between resources when developing within your azure platform

This should be carried out when any major new addition is made to your cloud environment. For example, it could include creating a new application or adding a new component of a greater platform.

- Start by separating resources into production and non-production environments.
- Use separate subscriptions and management groups for each workload in accordance with whether it will host production or testing data.[10] This applies to core services, such as networking, and to more specialized services.
- Although it is convenient to use peer networks to allow all resources to communicate freely between subscriptions, ensure resources in production are adequately segregated from any other environment. Follow a secure network topology layout[11].
- Implement logical separation for the resources to ensure that any compromise of a non-critical asset in a non-production environment will not propagate to a compromised equivalent resource in the production environment.

Reduce external-facing resources

Minimizing your organization's external footprint reduces the number of points of ingress into your network. In Azure, this can be done by:

- Configuring Azure's platform as a service (PaaS) services so that they are only accessible over a private endpoint that doesn't expose them externally.
- Limiting access through a local virtual network.

Ensure testing environments correspond

Environments should not become so distinct and disparate as a result of logical separation that they are no longer representative of one another.

This will lead to a loss of functionality in testing between pre-production and non-production environments, causing issues for your team and your end users. As such, the pre-production environment should always be as similar to production as possible, ensuring it effectively represents the live environment when performing security testing. This includes teams being allowed to use more granular development environments to test new features in different ways as part of the product release process (e.g., "dev", "nft", "nonprod" or "prod").

Review resource isolation periodically in light of changes in your environment

Resource isolation is the most effective solution to help minimize lateral movement from an initial foothold in Azure. As resources need to communicate with each other, it is also the hardest to track, and activity must be revisited and monitored following the addition of any new services or major changes to infrastructure.

Example: logical separation

Separate key vault stores should be created for production and development resources. If both production and non-production secrets are stored in a single secret store, then the compromise of a single Azure Key Vault or equivalent key store could lead to an attacker gaining access to all secrets used in the platform.

Backups and disaster recovery

Principle: backups can keep business-critical resources online 24/7 in the event of a disaster.

Disasters do happen and losing all your business-critical resources right when they are most needed is something every organization should be prepared for. The main cloud providers, Azure included, guarantee 99.99% availability for your services and data. However, the Code Spaces[12] and OHV[13] incidents have shown us the importance of always being prepared for the 0.01% chance not covered by Azure's Service Level Agreements (SLAs)[14]. Thankfully, there are viable measures you can take internally to protect your organization in this scenario.

Implement frequent backups for your most critical resources and services

This could include snapshots of any particularly critical VMs, replication of Storage Accounts (with both cold and hot storage options), containers hosting sensitive or business-critical data, DevOps project repositories, Key Vaults containing keys, and secrets for your most used applications.

Backups can be performed natively through the Azure Backup[15] service. Organization administrators can define resources and frequency, as well as other selection criteria to set the backup policy within the Azure environment.

Implement comprehensive business continuity and disaster recovery strategies

Further measures can be taken to minimize the impact to business operations in the event of a fault. This includes protecting your data, apps, and workloads, and keeping them online. For this purpose, Microsoft offers:

- Azure Site Recovery[16]: keeping business apps and workloads running during outages.
- Azure Migrate[17]: a centralized hub for discovery, assessment, and migration of on-premise machines to Azure.

How to: store backups

Small organization without critical data

Local Redundant Storage (LVRS) and Zone Redundant Storage (ZRS) options tend to be cheaper. In the unlikely event of an outage in multiple data centers, smaller organizations may not need backups in other regions.

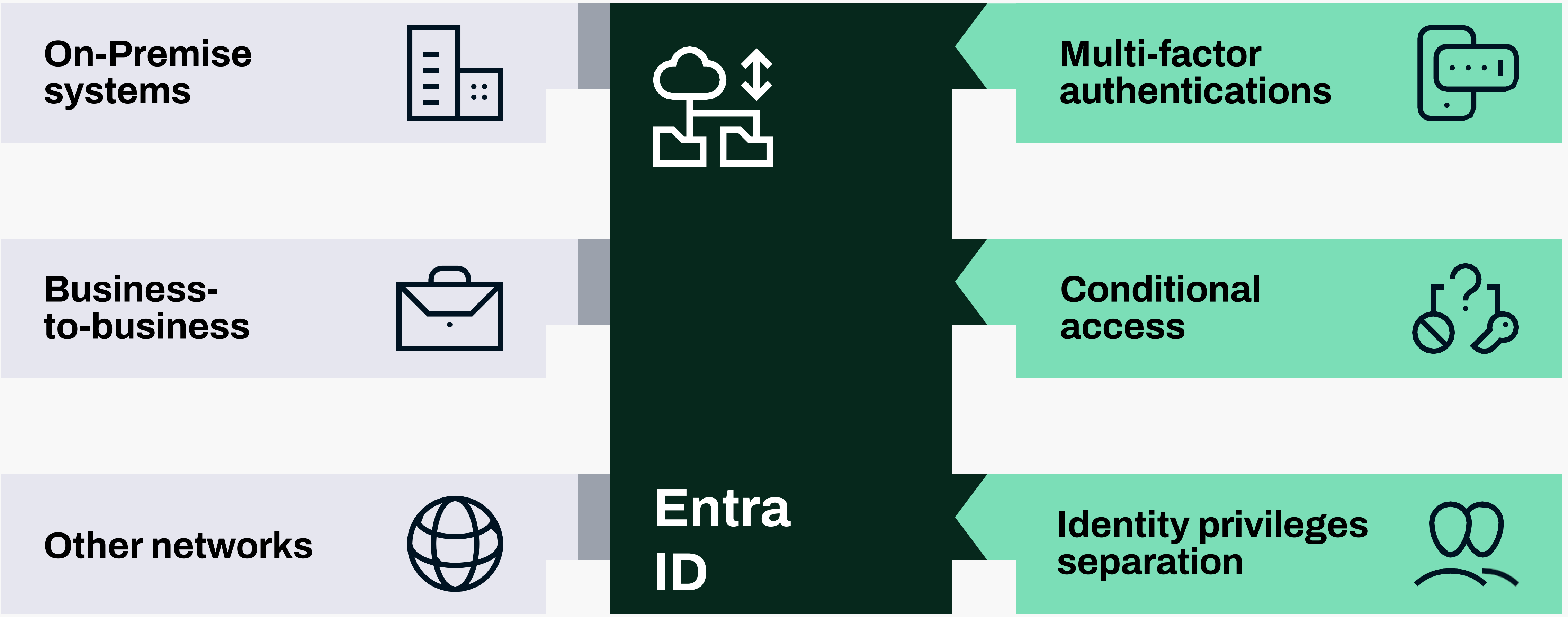
Large enterprise with significant critical data

Backups in separate regions (Geo-Redundant Storage options —GRS and RA-GRS). This helps minimize the risks related to any potential region-wide outages.

Identity and access management (IAM)

Principle: defining appropriate user permissions and auditing them can limit an attacker's access and persistence.

For all cloud providers, IAM is crucial, as it serves as a first line of defense against malicious external and internal threats. Identity access controls can help track and manage what type of access users have, and which actions are available to them for the various resources of a tenant. This can help you better understand how users interact with services, prevent data breaches, minimize account hijacking, and prevent lateral movement and privilege escalation. IAM is one of the most important areas of cloud security, but the hardest to get right. In Azure, it is split across Resource-Based Access Controls (RBAC) and identity-related access controls under Entra ID. Though similar in some regards, they are two different areas, and it is important to get both right.



RBAC

Azure's resource authorization system involves the assignment of roles to users, groups, service principals, or managed identities, at a particular scope.

Apply the principle of least privilege from the outset

In the place of coarse-grained roles such as "Owner" and "Contributor", more granular roles can be applied that consider the specifics of the tasks delivered and the permissions required. An attack surface increases unnecessarily with excessive role assignments. One compromised senior engineer could lead to the compromise of multiple projects that they didn't require access to.

Implement a bottom-up approach

1. **Identify your starting point**. Begin with the most important individual resources or resource groups, i.e., those critical to daily operations and output, and set role assignments to only the relevant personnel. This will help avoid broadly defined access to management groups or subscriptions, which would lead to unintended, inherited access to the components within them.
2. **Start with pre-made roles**. Make use of Azure's pre-made specialized roles first. If those do not satisfy the needs of your team, create custom-defined ones wherever relevant[20,21]. (The next section on Entra ID explains how custom role misconfiguration exposes privilege escalation paths.)
3. **Assign roles in a hierarchy.** Set "Owner" roles for relevant resources or resource groups, and for any personnel who are the resource or project owner(s) for those components. Any other members, developers, or application service principals requiring access in that specific project should be assigned specialized roles based on their required activity:

   - Make use of "Reader" access. Members of staff from other teams and external parties should only be provided read-only "Reader" access to the environment. This can be supplemented by any other RBAC assignment required to perform any audit checks needed.
   - Gradually move "up" to less critical resources. This will include broader selections, such as resource groups or even subscriptions.

Following these steps will lead to a well-defined and restrictive, but functional, set of role assignments. Resource access will be granted to only the essential required personnel group or service, without exposing them to undue risk.

Entra ID

When using Entra ID as your identity provider, there are options for managing IdAM in the cloud.

If you're implementing Business-to-Business (B2B) collaboration, use Microsoft Entra External ID to provision Guest accounts in your Entra tenant, but perform regular audits[22] and remove old, unused, or unnecessary ones[23].

If you're implementing Business-to-Consumer (B2C), security becomes heavily dependent on the type of customers you serve and the applications you host. Set up Conditional Access policies as a minimum. For example, governing approved locations or user access based on their risk level (the probability that a user account is compromised). This will tighten authentication policies and minimize the effectiveness of credential stuffing attacks.

Implement single sign-on (SSO)

SSO will enable users to authenticate and access the resources they need with the same set of credentials. This avoids the need for multiple passwords for various services, reducing

the likelihood of weak passwords or reuse. The benefits of SSO apply to cloud-native environments, hybrid cloud, and on-premise environments.

Manage all of your environments

The way you build your cloud environment— whether it is cloud-native, on-premise, or a hybrid solution—will determine your approach to authentication for IAM. In most cases, organizations tend to have a hybrid solution, even if the end goal is to be fully cloud-native. For example, you might use an application proxy that opens on-premise services to the cloud or leverage Active Directory Domain Services (ADDS)[24] in the cloud.

With Entra ID, there are 3 main IAM options that you can implement. Typically, you will be leveraging Entra ID Connect to synchronize with your onpremise estate:

- **Entra ID Password Hash Sync (PHS)**.
  This is the least effort option, where validation happens completely in the cloud. Entra ID Connect synchronizes cloud identities with password hashes on-premise and does not require any additional infrastructure.

- **Entra ID Pass-Through Authentication (PTA).**
  This approach uses a "middleman" authentication agent (1-3 maximum recommended), validating password signin attempts with the Domain Controller (DC) on-premise. On-premise account policies are enforced at the time of sign-in. PHS should be deployed as a backup method.

- **Federated Authentication, either with Active Directory Federation Services (ADFS) or other third-party federation providers.** Here, sign-in attempts are redirected to federation proxies between cloud and on-premises. The federation servers perform the validation with on-premises AD. Password hash synchronization (PHS) can be included also.

Assign, monitor, and manage user permissions in Entra ID

1. **Apply the principle of least privilege on the users in your Entra ID tenant.** Always aim to have a minimum of 2 and a maximum of 4 Global Administrator (GA) accounts to avoid ever being completely locked out of your environment. Too many GA accounts increases the risk of targeted phishing attacks, potentially resulting in compromised accounts with unrestricted permissions.
2. **Assign specialized, narrow roles for administrative requirements, and read-only roles for your users [25], in a similar fashion to the Azure RBAC roles.**
3. **Monitor the activities of service principals by setting alerts for suspicious activities.** You will most probably end up having service principals within your tenant as a result of third-party solutions or entities which need to run with high privileges. Compromise of these high-privileged applications would have a significant impact on your environment, giving a foothold to an attacker. Either delete service principals them when not needed or consider less-privileged roles as substitutes.
4. **Use Entra ID's built-in roles, and when these do not fit a necessary team member role, create custom ones defining the needed permissions[26].** With custom roles, due to the complexity and granularity of the permission model, make sure to avoid the broad permission definitions using the star ("*") actions. These could lead to assigning an unintended level of access, so ensure your policy definitions only assign necessary permissions, for the necessary scope.

Use Entra Privileged Identity Management (PIM) to avoid unnecessary Administrator accounts

Constantly updating user access permissions can become an intensive overhead for your IT personnel. Just-in-Time (JIT) access and automation solutions for this problem do exist, including Entra Privileged Identity Management (PIM)[27]. This is a must-have security service for organizations that can afford an Entra ID P2 license. With PIM, users need to request the role access they require, when they require it, enforcing a time constraint on the role. This significantly reduces the number of excess administrator accounts which can be compromised and used by attackers to perform privileged actions within Azure. It also helps solve the issues of timeboxing and auditing of access management. Enforce strict policies for highly privileged roles such as GAs, such as a maximum of two hours restriction and approval requirement from another senior administrator. Similarly, for resources, Just-in-Time (JIT) access can be implemented to reduce exposure.

Use Entra Conditional Access (CA) policies to govern sign-in attempts and set conditions to be met after a correct sign-in

An Entra ID P1 license is required at a minimum. CA policies allow you to enforce things like:

- Multi-factor authentication (MFA)
- Trusted locations or trusted/compliant device logins, via Microsoft Intune MDM
- Restricting specific types of clients from authenticating into the estate
- Advanced risk-based sign-in management via Entra ID Protection

Implement MFA for all administrative users

This will create an additional security layer for sign-in attempts and transactions. Ideally, you should plan a gradual rollout to all users in your tenant. Implementing these measures will ensure that all your important administrative users are adequately protected with MFA and can request the high-level access needed to perform a more sensitive action only when needed. In doing so, your internal employees, apps and services, and any third-party collaborators in your tenant will have access to your environment governed by the policies you've established. This means they will only be able to gain access from locations or devices that you have defined as safe and accepted, and activity carrying risk will be monitored.

Use Azure Key Vaults to manage access to authentication credentials

When credentials or secret objects are required for authentication, use a centralized storage system such as Azure Key Vault. Users and apps can still use credentials to authenticate whenever needed, but only permitted users will be able to access them. Ensure granular RBAC roles are set at both the vault management level and the secrets level.

Monitor and update permissions to maintain principle of least privilege

With an established plan for IAM configuration, organizations should continuously monitor people's access and roles. Ensure these are edited or removed accordingly whenever needed, maintaining the principle of least privilege to avoid unnecessary risks.

Logging and monitoring

Principle: comprehensive and continuous log collection creates a reliable and detailed audit trail for both resource performance and control plane activities.

Logging is an area which requires a continuous effort: your resources might scale, get moved, and acquire different purposes, and new services might be introduced. You should start configuration and Log onboarding within the Security Operations Center (SOC) as early as possible and add other Telemetry as your Azure platform evolves. Getting your logging and monitoring right from the beginning helps ensure that nothing is overlooked. Left until later, the implementation of a logging plan becomes a time-consuming task and risks suspicious activity going undetected if the right Logs are not configured.

Logging basics

1. **Establish visibility over what happens in your tenant, both at the control plane level and the resource level.** This reveals what actions users are performing in your tenant and with your resources. Performance information helps you make the most of your resources. It also gives clues about how a suspicious login, a sudden spike in a VM performance stats, and a high number of read operations from a Storage Container could together be indicators of data compromise and exfiltration.
2. **Ingest telemetry for resources and tenant activity into your SOC**. These logs should be used in conjunction with alert policies to inform relevant SOC staff of activities of interest or events happening within your tenant. Effective alerts are fundamental to your incident readiness and response effort. Creating an effective set of alerts requires contextual information about the Components within your tenant.

Log types

You can’t enable logging for everything, as this will incur greater costs and can overwhelm the SOC with low-risk alerts. Start small, and focus on the most critical resources first:

- Use metrics logs to create visibility over what happens internally to your resources, such as performance, health, diagnostics, technical issues, etc.
- Use activity logs to create visibility over what events happen with resources in your tenant, such as control plane actions like creation of resource groups, a change to an Azure Policy, etc.

Azure monitor

The Azure Monitor service[28] works as a comprehensive solution for collecting, analyzing, and acting on telemetry, both from your cloud and on-premises environments. It provides general Information on how resources in your environment are performing and more detailed information on activity

and telemetry for infrastructure, applications and networking aspects. Monitor can collect all available information for each supported resource. To enable this collection, set up the Azure Monitor agents (e.g., for VMs), and enable diagnostic settings for each service supporting them (SQL DBs, Key Vaults, Entra ID, etc.).

Metrics

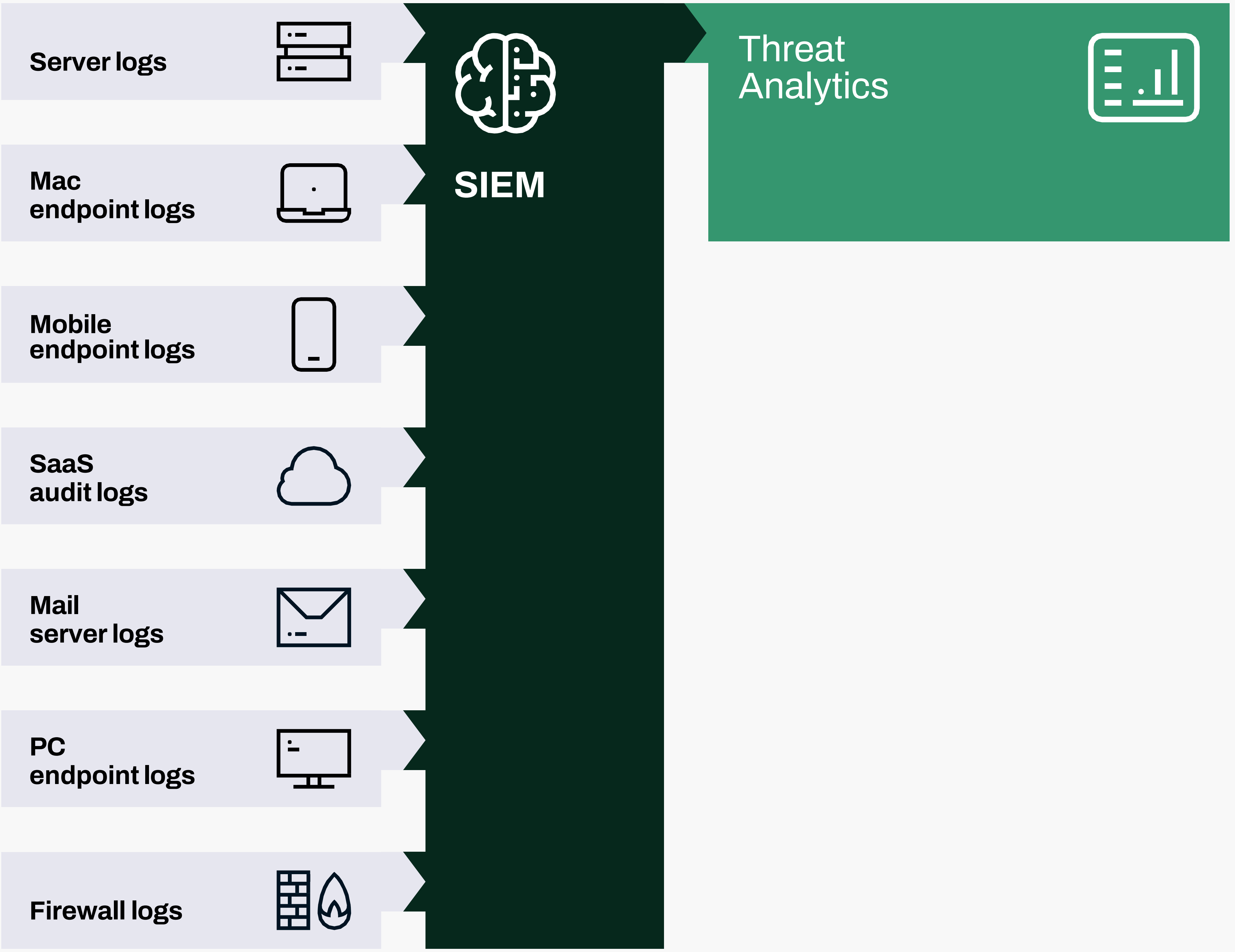
Whether it's suspicious and unexpected activity within your network flow traffic, abnormal use of your resources, spikes in performance, or unusual devices being suddenly active, these kinds of metrics can alert you to a potential compromise within your environment.

1. **Enable Application Insights to monitor both cloud and on-premise**. If you have managed Application Services, Azure Functions, Azure Kubernetes Service etc. deployed within your tenant, enable Application Insights for them, to monitor the availability, performance, errors, exceptions, and usage. It can monitor both cloud and on-premise environments. Enable NSG Flow logs in Virtual Networks and write them to a storage account or Log Analytics workspace, to have clear visibility over network traffic.
2. **Enable diagnostics logs for "Compute" type resources.** "Compute" type resources include VMs, Container Instances, Kubernetes Service, etc. Enable diagnostics logs and deploy the relevant Azure Monitor agents (more on Log Analytics further down below). Ensure that both platform-level (what Azure sees) as well as host-level (what the OS sees) monitoring is in place.
3. **Enable Storage Diagnostics for "Storage" type resources.** "Storage" type resources include Storage Accounts, SQL Databases, storage disks, etc. Enable diagnostic settings, so that Monitor can collect metrics on each component of a storage account.

Entra ID logs

The following actions can be taken to enable and make use of logging from Entra ID:

1. **Enable Audit and Sign-in logs in the tenant diagnostic settings to create visibility over events at the control plane level[29].** The core types of logs to enable are the following:
   - **AuditLogs**: the history of every platform management task that is performed in your tenant
   - **SignInLogs**: collection of data on all user and service principal sign-in events
   - **NonInteractiveUserSignInLogs**: sign-in events that are done on behalf of a user.
   - **ServicePrincipalSignInLogs**: sign-in events from service principals and applications
   - **ManagedIdentitySignInLogs**: sign-in events from managed identities in the tenant
   - **MicrosoftGraphActivityLogs**: audit logs of all HTTP requests accessing tenant resources via the Graph API



Log analytics

Log Analytics workspaces allows the storage and processing of logs from all your various sources[30]. This enables correlation of data, complex analysis, insights, and querying capabilities via the Kusto Query Language (KQL). Sending resource and activity logs to Log Analytics is the preferred option, while storage accounts should be used for long-term cheaper data retention.

1. **Use workspaces to group logs and manage who can access them.** Opt-in for multiple workspaces to more effectively group logs by purpose or by logically-related resources. For backup purposes or for manual review all critical logs should also be sent to a storage account configured for cold storage of data.

   Risk can be lowered by restricting access to certain types of log data, such as sensitive access logs or Key Vault logs. Plan carefully how you will create workspaces and ensure that each has appropriate access permissions configured, based on the purpose of the workspace and the sensitivity of the logs stored there.

2. **Retain your logs for backups, disaster recovery, and investigation in the event of a cyber attack**. Cost is an essential consideration for log collection. However, defaulting to the standard retention period of 90 days for activity logs, having them appear in each resource's page, and not considering Workspaces fails to take advantage of using Log Analytics for establishing a timeline of events and understanding of the actions. Storing and analyzing more logs becomes increasingly advantageous when you consider that initial compromise tends to happen several months before the first detection of an intrusion. As Log Analytics allows you to also increase the retention policy up to 730 days, opt for a longer retention policy that suits your business needs.

Policies

Principle: effective guardrails help establish a minimum security and compliance configuration, without disrupting development activities.

As your Azure environment develops—especially in fast-paced, growing organizations—a lack of effective policies can result in unchecked areas of vulnerability, leaving an open goal for attackers. Guardrails are the most efficient way to ensure that any resource has a secure baseline configuration.

Control resource deployments with Azure Policy

This is one of the key services available in Azure to ensure resource governance at scale and is the primary way of establishing guardrails in a given platform. Azure policies are made up of a policy definition resource written in a JSON format.
Within that policy definition, users can define the specific parameters it evaluates, the logical condition it uses to evaluate those parameters, and the action that will be performed if the condition is evaluated to be true. When a policy is evaluated, Azure Policy provides 11 options for actions, including: “Audit”, “Deny”, and “DeployIfNotExists”.[31] These self-explanatory activities represent the common outcomes and can be used either to just provide oversight or directly interfere with the usage of a noncompliant resource.

Organizations have different requirements for their environments, resources, and the services they are using. It is important to define some custom policies, fitting security guardrails appropriate to the organization’s development processes.

Review Microsoft’s pre-defined policies[32]

These pre-made policies serve as a good foundation for your organization’s Azure environment. They are built around common security recommendations from governing bodies such as the Center for Internet Security (CIS)[33] or National Institute of Standards and Technology (NIST)[34].

Once you review the pre-made policies, scope any relevant policy definitions accordingly to a subscription or management group level. Where possible, use the pre-made policies to not only audit resources, but also attempt to remediate common issues by explicitly denying the creation of insecure services.

Define custom policies to prevent development misconfigurations

When defining custom policies, try to strike a balance that works for your organization. Your policies should be sufficiently restrictive to prevent development misconfigurations, but also maneuverable enough so that developers and engineers do not attempt to "work around" them. Due diligence must be carried out to ensure that any required exceptions are categorized and recorded in advance. Additionally, the implementation of policies should be done only after sufficient testing using an audit condition. As a first step, implement custom policies such as:

- A policy to ensure appropriate resource tagging on creation.
- A policy to automatically enable appropriate logging for all created resources.
- A policy to define approved regions and services that could be deployed to a production environment.

Test your policies to ensure they do not hamper the development workflow

Careful deliberation will be required to tailor any policy guardrails to your environment. Once implemented and considered as part of the development workflow, your guardrails ensure that any created resource has a secure baseline configuration. Each product can now be built upon continuously and any new security features can be added to the templates so that security doesn't have to lag behind development.

Resource governance

Principle: continuous evaluation of existing practices can highlight deficiencies in your security and inform design improvements.

Resource governance is essential in any digital environment. In the context of Azure, it will not only help you identify areas for improvement. It also highlights potential faults that cause the recurrence of security issues due to new developments. If current processes are failing to build a scalable and secure environment, now is the time to consider making some core design changes.

Define, build, and review deployment templates

Modern development can make infrastructure too complex to manage manually, which has prompted a change in thinking about controlled automation. It is important to support new processes, such as deployment templates, ensuring they establish a secure baseline that is both maintainable and reproducible.[35]

There are multiple native and third-party tools that can be used to define and build deployment templates. A popular third-party tool that integrates with Azure is HashiCorp Terraform[36], which supports the creation of complex infrastructure via Terraform files.

Alternatively, an entirely native Azure solution would be to create Azure Resource Manager (ARM) template definitions[37] and have the templates built within the cloud environment using the Azure Blueprints service[38]. As new resources will likely be deployed using these same templates, they are a key component in establishing a security baseline across all new resources. These templates must establish secure defaults and the necessary tags to ensure Azure Policy definitions correctly audit each resource on creation.

Use caution when deploying any new templates within a cloud environment. Implement a "four-eyes" review process for any new deployment template to ensure it does not unnecessarily increase the available attack surface of the platform. These security baseline checks can be done via automated tooling (as part of the pipeline deployment) to ensure the template has not changed fundamentally from the reviewed version.

Audit resources within your platform regularly

In an ideal scenario, any small change would be reviewed at the time and deemed expected (i.e., safe) or potentially malicious. However, reality is rarely so kind. The size and the number of deployments required to complete a product can make manually reviewing each one impractical. As such, the cloud management team should conduct monthly audits of Azure Policy data to investigate the status of, and reasons behind, any noncompliant resources

.

Continuous detection and monitoring

Principle: continuously collecting relevant telemetry and implementing suitably prioritized alerts support an effective detection and response capability.

Log management is a continuous process, needing constant attention and updates based on changes to your environment and resources. But only collecting logs no follow-on activity wastes both effort and money. When logs are processed, grouped accordingly by risk, false-positive fidelity, actions, etc., SOC analysts can better prioritize their time and focus on the most pressing alerts. This should be the goal.

Establish a list of the most critical resources present within your environment

Here, security analysts and engineers should come together to create an established list of critical resources that can be used to prioritize alerts. Consider important users and service accounts that would be the likely targets of an attack.

Set up risk-prioritized altering policies based on the criticality of resources

In doing so, once there is a prioritized list of resources and personnel know what each resource is expected to do, they will be notified by alerts when certain conditions are met. The context of these organized logs will then allow analysts to deduce malicious behavior or if further manual review is needed. As a minimum, define a number of core alerts. For example:

- a privileged role assignment has been performed
- a critical Azure Policy has been modified
- an unexpectedly high number of resources (such as VMs) has been deployed
- an unexpected access to an Azure Key Vault entry
- impossible travel distances for logins
- logins from unidentified or unexpected locations

Enable Foundational CSPM in Microsoft Defender for Cloud[39]

This is an invaluable one-stop-shop for security alerts and remediation actions, and it has great integration with most resources and services. It provides a good single pane of misconfigurations present within the tenant. It provides any organisation with a free baseline that could be adhered to without additional costs.

Implement a security self-assessment practice

Use your self-assessment to monitor and identify security-related issues within the tenant and feed the information back into the SIEM solutions. Tools such as Maester[40] can help significantly as part of continuous monitoring. It performs checks for all sorts of misconfigurations across a tenant and is continuously updated with new types of tests.

Create automated runbooks for actions to be performed upon detection of a specific event

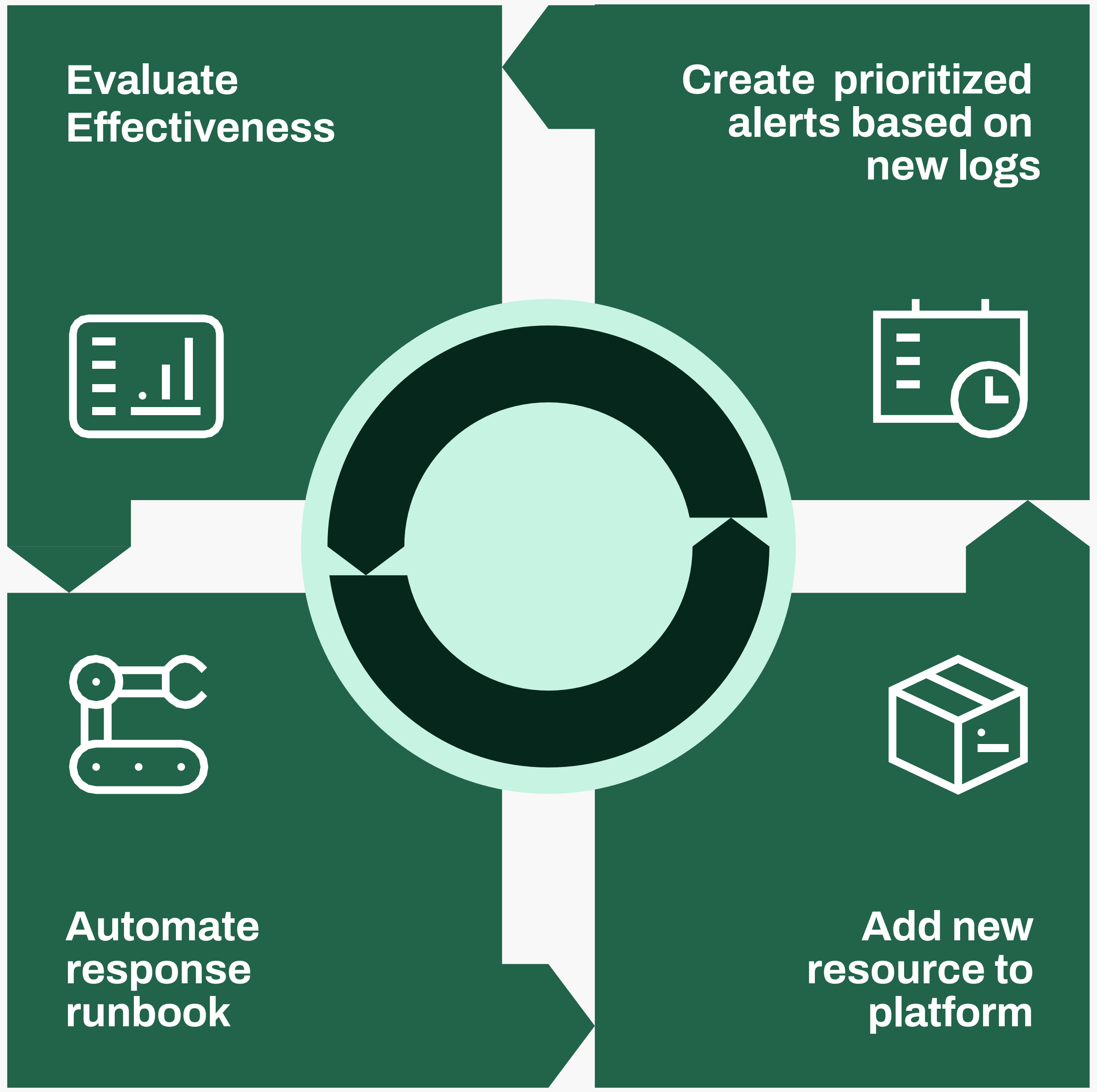
These can help reduce certain types of alerts, thus reducing the time spent on more trivial alerts and tasks. As your Azure platform evolves, sending specific types of alerts to a ticketing solution (e.g. Jira) may also be useful for security and risk management.

Use Entra ID Protection (P2 Licenses) to identify user and sign-in Risks

Act upon these events based on set risk policies and/or alert personnel for manual investigation. Machine learning can aid this process by minimizing false positives. It can be set up in conjunction with Entra Conditional Access policies for highly-advanced authentication protection.

As an organization increases in size and its security posture develops, the need for a SIEM and a SOAR (Security Orchestration Automation Response) increases. Microsoft Sentinel[41] is Microsoft's native offering, with a plethora of features to correlate logs and alerts and group together possible threats for further investigation.

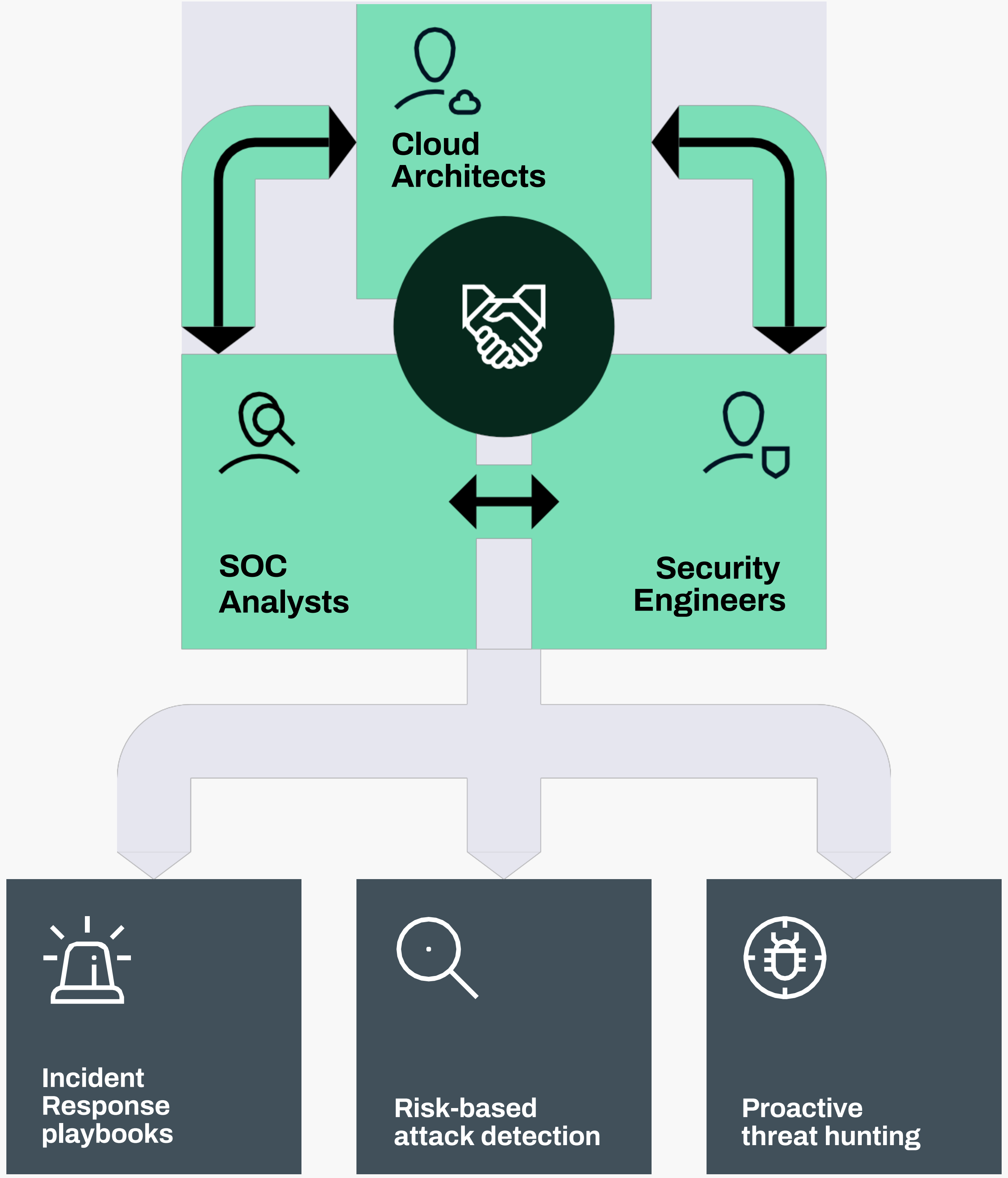
Machine-learning-based analytics can help identify advanced security incidents and provide powerful hunting capabilities. Integration with Microsoft Defender for Cloud and Microsoft Sentinel playbooks will allow you to build automated actions for remediation.


Evaluate Effectiveness
Create prioritized alerts based on new logs
Automate response runbook
Add new resource to platform

Incident response (IR)

Principle: collaboration between cloud engineers and security analysts, and the use of appropriate tools, help form a capable and proactive investigation team.

The “assume breach” model posits that an attacker will eventually attain some level of privileged access within your environment. As your organization increases the number of services it uses, the number of potential points of compromise increases too. We’ve already discussed how to monitor for such intrusions and minimize lateral movement. This section will provide guidance on what to do when you observe telemetry pointing to a live security incident.



Create incident response playbooks for your cloud environment

In the context of thorough, accurate, and up-to-date incident response plans and playbooks, the cloud is still a new frontier. The MITRE cloud Matrix is being continuously updated as new data is gathered on known breaches[42].

Tactics, Techniques, and procedures (TTPs) used in the cloud are changing all the time, so, as with on-premise, it is important to establish a dynamic and continuous approach to your cloud-based IR activities. Establish playbooks that include an action plan for assumed breach. Given the increase in cloud attacks and the novelty of the TTPs used, it is crucial that cloud engineers, the SOC, and key platform stakeholders work together to ensure that they are as contextually-relevant as possible.

To encourage continuous improvement and providing a current baseline against attacks, evaluate and update your cloud IR playbooks regularly, whenever:

- new deployments have been made within your platform
- new threat intelligence is published
- new offensive techniques are identified publicly

Further IR measures

- Perform regular tabletop exercises to prepare for various intrusion points based on their level of assigned risk.
- Configure all critical components to generate as much telemetry as possible. This data should be funneled to the security team or SOC for their continuous review.
- Utilize Microsoft resources to supplement your existing playbooks with guidance around common attack methods used currently by attackers[43].
- Where possible, organizations should conduct their own internal research and analysis to ensure their response effort is relevant and appropriate.
- Accomplishing this requires significant time and resources, and may not be possible large-scale. However, minimal investment can potentially change the outcome of a compromise. Your central aim should be to reduce the time it takes for the security team to identify a live threat and respond with the correct measures.

Significant time and resource are needed and may not be possible large-scale. However, minimal investment can potentially change the outcome of a compromise by reducing the time it takes your security team to identify live threats and respond.

Use the tools at hand to generate high-impact alerts based on anomaly detection

Native tools such as Microsoft Sentinel enable IR teams to build responsive activities that can be triggered automatically, or by a human operator. Automation means these actions can be performed at scale across a large organization. Although the specifics can differ between security orchestration automated response (SOAR) tools, the fundamental idea remains

the same: certain actions can be automatically applied the moment a given security alert is triggered. This helps reduce the impact to the organization while maximizing immediate, automated response to certain critical alerts.

What next?

The principles covered in this document are the building blocks with which you can form a solid foundation for your Azure security. Building anything takes time—it won't be achieved overnight. Take stock of your current security posture and the resources you have available, such as budget, team members, and timeline. This will help you to prioritize where to start and what you can achieve in the short term. No matter where you start, one maxim remains true: any cloud environment that is secure and well-organized will help minimize the risk and impact of cyber attacks targeting it.

In reality, there is no such thing as a completely secure environment. Your IT infrastructure and the Azure services you use will inevitably change over time and your cloud environment is likely to grow more complex. For this reason, maintaining the security of your cloud platform relies on continuous improvement. Establishing processes for repeated, periodic review will make remediation activities easier to achieve.

With an increase in cloud migration expanding this attack surface, we can expect an increase in attacks on organizations' cloud environments. Owing to their novelty, we have some understanding of the attack techniques used, but the available threat intelligence is more limited. Updating and expanding our knowledge of the techniques attackers are likely to use against cloud environments will take time and (unfortunately) concrete evidence from real cyber attacks. Always remember: a stitch in time saves nine. Begin securing your cloud environment sooner rather than later and build defense-in-depth from the start. As attacks in the cloud increase in frequency and sophistication, you will stand a much better chance than those organizations who fall behind.

Principles

Principle 1: Understanding and logically grouping all resources avoids the growth of an unrecognized attack surface.

Principle 2: Isolation of resources helps reduce the "blast radius" of a cyber-attack.

Principle 3: Backups can keep business-critical resources online 24/7 in the event of a disaster.

Principle 4: Defining appropriate user permissions and auditing them can limit an attacker's access and persistence.

Principle 5: Comprehensive and continuous log collection creates a reliable and detailed audit trail for both resource performance and control plane activities.

Principle 6: Effective guardrails help establish a minimum security and compliance configuration, without disrupting development activities.

Principle 7: Continuous evaluation of existing practices can highlight deficiencies in your security and inform design improvements.

Principle 8: Continuously collecting relevant telemetry and implementing suitably prioritized alerts support an effective detection and response capability.

Principle 9: Collaboration between cloud engineers and security analysts, and the use of appropriate tools, help form a capable and proactive investigation team.

References

1. What is Azure? https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure/
2. AWS Security Maturity Roadmap https://summitroute.com/downloads/aws_security_maturity_roadmap-Summit_Route.pdf
3. Hunting Azure Blobs Exposes Millions of Sensitive Files https://www.cyberark.com/resources/threat-research-blog/hunting-azure-blobs-exposes-millions-of-sensitive-files
4. Tutorial: Discover and manage shadow IT in your network https://learn.microsoft.com/en-us/defender-cloud-apps/tutorial-shadow-it
5. Use tags to organize your Azure resources and management hierarchy https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources
6. Resource naming and tagging decision guide https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming-and-tagging-decision-guide
7. Assign policy definitions for tag compliance https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/tag-policies
8. What are Azure management groups? https://learn.microsoft.com/en-gb/azure/governance/management-groups/overview
9. Use cost alerts to monitor usage and spending https://learn.microsoft.com/en-us/azure/cost-management-billing/costs/cost-mgt-alerts-monitor-usage-spending
10. Organize your Azure resources effectively https://learn.microsoft.com/en-gb/azure/cloud-adoption-framework/ready/azure-setup-guide/organize-resources
11. Traditional Azure networking topology https://learn.microsoft.com/en-gb/azure/cloud-adoption-framework/ready/azure-best-practices/traditional-azure-networking-topology
12. Code Spaces goes titsup FOREVER after attacker NUKES its Amazon-hosted data https://www.theregister.com/2014/06/18/code_spaces_destroyed/
13. OVH data center burns down knocking major sites offline https://www.bleepingcomputer.com/news/technology/ovh-data-center-burns-down-knocking-major-sites-offline/
14. Service-level agreements https://azure.microsoft.com/en-gb/support/legal/sla/
15. What is the Azure Backup service? https://learn.microsoft.com/en-us/azure/backup/backup-overview
16. About Site Recovery https://learn.microsoft.com/en-us/azure/site-recovery/site-recovery-overview
17. Azure Migrate https://azure.microsoft.com/en-gb/products/azure-migrate/
18. Assign Azure roles using the Azure portal https://learn.microsoft.com/en-us/azure/role-based-access-control/role-assignments-portal
19. Assign Azure roles to a managed identity (Preview) https://learn.microsoft.com/en-us/azure/role-based-access-control/role-assignments-portal-managed-identity
20. Azure custom roles https://docs.microsoft.com/en-us/azure/role-based-access-control/custom-roles
21. Create and assign a custom role in Entra ID https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/custom-create
22. What are Entra ID access reviews? https://learn.microsoft.com/en-us/entra/id-governance/access-reviews-overview
23. How To: Manage inactive user accounts in Entra ID https://learn.microsoft.com/en-us/entra/identity/monitoring-health/howto-manage-inactive-user-accounts
24. What is Azure Active Directory Domain Services? https://learn.microsoft.com/en-us/entra/identity/domain-services/overview
25. Entra ID built-in roles https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/permissions-reference
26. Create and assign a custom role in Entra ID https://learn.microsoft.com/en-us/entra/identity/role-based-access-control/custom-create
27. What is Entra ID Privileged Identity Management? https://learn.microsoft.com/en-us/entra/id-governance/privileged-identity-management/pim-configure
28. Azure Monitor overview https://learn.microsoft.com/en-us/azure/azure-monitor/overview
29. Audit logs in Entra ID https://learn.microsoft.com/en-us/entra/identity/monitoring-health/concept-audit-logs
30. Log Analytics tutorial https://learn.microsoft.com/en-us/azure/azure-monitor/logs/log-analytics-tutorial

31. Understand Azure Policy effects https://learn.microsoft.com/en-us/azure/governance/policy/concepts/effect-basics
32. Azure Policy built-in policy definitions https://learn.microsoft.com/en-us/azure/governance/policy/samples/built-in-policies
33. Details of the CIS Microsoft Azure Foundations Benchmark 2.0.0 Regulatory Compliance built-in initiative https://learn.microsoft.com/en-us/azure/governance/policy/samples/cis-azure-2-0-0
34. Details of the NIST SP 800-53 Rev. 5 Regulatory Compliance built-in initiative https://learn.microsoft.com/en-us/azure/governance/policy/samples/nist-sp-800-53-r5
35. Repeatable Infrastructure https://learn.microsoft.com/en-us/azure/well-architected/operational-excellence/infrastructure-as-code-design
36. Terraform https://www.terraform.io/
37. What are ARM templates? https://learn.microsoft.com/en-gb/azure/azure-resource-manager/templates/overview
38. Managing Blueprints as Code https://github.com/Azure/azure-blueprints/blob/master/README.md
39. What is Microsoft Defender for Cloud? https://learn.microsoft.com/en-us/azure/defender-for-cloud/defender-for-cloud-introduction
40. Maester https://maester.dev/
41. Microsoft Sentinel https://learn.microsoft.com/en-us/azure/sentinel/overview
42. Cloud Matrix https://attack.mitre.org/matrices/enterprise/cloud/
43. Incident response playbooks https://learn.microsoft.com/en-us/security/operations/incident-response-playbooks

We're global. Get in touch wherever you are.
https://consulting.withsecure.com/contact-us/

Who We Are

As the leading experts in cyber security, WithSecure™ Consulting stands at the forefront of safeguarding organizations globally. At WithSecure™ Consulting, we distinguish ourselves through our research-led approach and our teams comprising some of the finest minds in the cyber security domain. With a global footprint, we have successfully helped organizations across industries in fortifying their defenses and mitigating cyber risks, enhancing operational reliability and preserving critical business continuity.

WithSecure™ Corporation (formerly F-Secure for Business) was founded in 1988, and is listed on NASDAQ OMX Helsinki Ltd.

W/TH secure | Consulting