Microsoft Security

BlueVoyant®

Azure Sentinel Deployment Best Practices

Authors:

Adrian Grigorof, CISSP, CRISC, CCSK

Marius Mocanu, CISSP, SABSA, CISM, CEH

Jordan Shaw-Young, CISM

Table of Contents

Introduction

The purpose of this whitepaper is to provide security organizations with a practical field guide to assist in developing a deployment strategy for Microsoft Azure Sentinel that will employ best practices to support a stable, cost-effective, and operationally effective implementation of Microsoft's cloud-native security information and event management (SIEM) platform. This document is written from a security practitioner perspective, based on experience deploying and managing Azure Sentinel in a wide range of organizations.

We intend for this guide to serve as a reference and planning document primarily for chief information security officers, security architects, and enterprise architecture and project management leaders in defining adoption and migration strategies and budgets and in planning project and resourcing requirements for a successful implementation of Azure Sentinel. It can be read as a companion document to other Azure Sentinel technical whitepapers such as the *Azure Sentinel Technical Playbook for MSSPs.*[1]

Azure Sentinel cloud-native SIEM

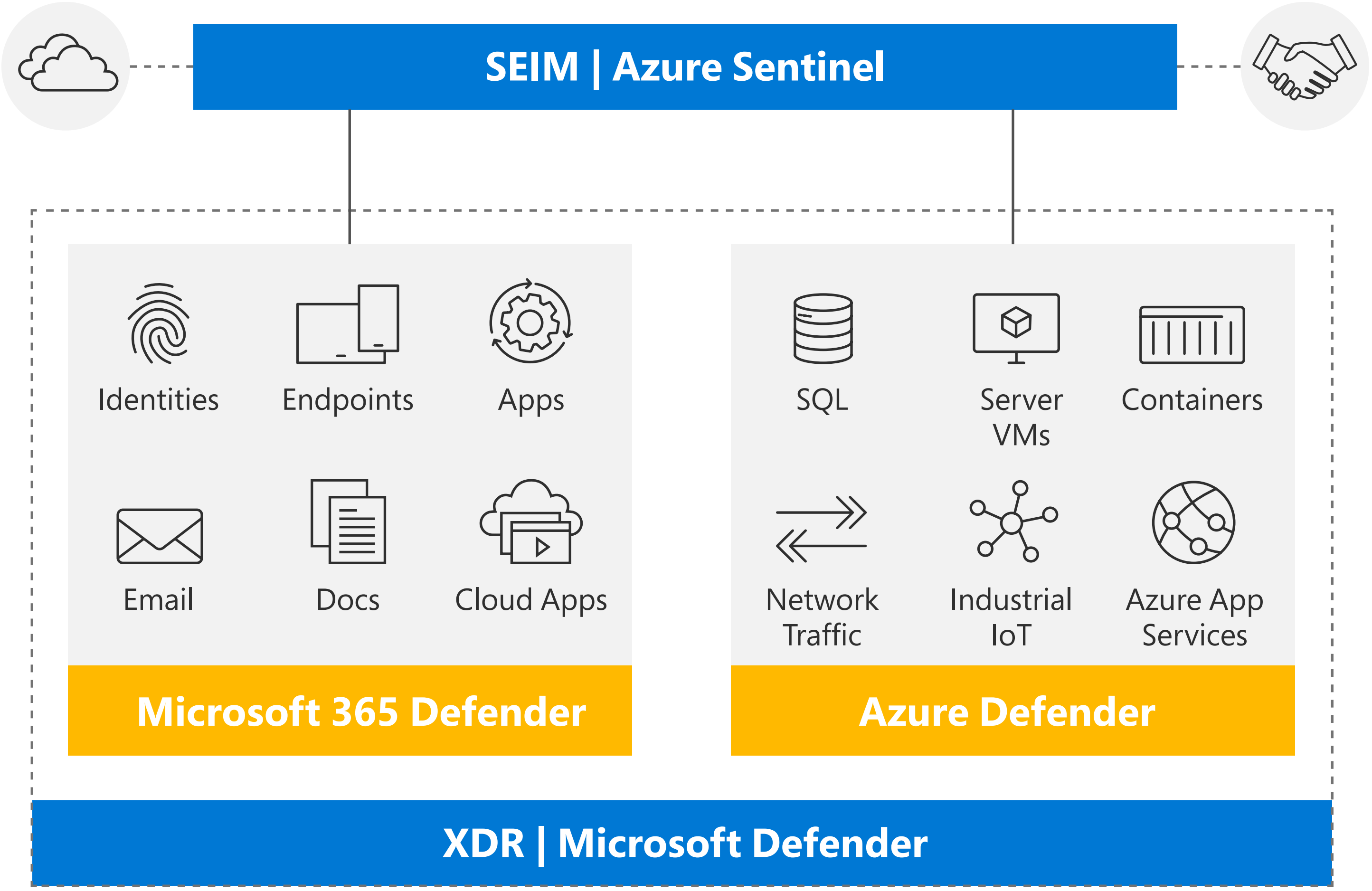
Azure Sentinel is Microsoft's cloud-native SIEM solution and the first cloud-native SIEM from a major public cloud provider. Azure Sentinel is deployed in an organization's Azure tenant and accessed via the Microsoft Azure portal, ensuring alignment with preexisting organizational policies.

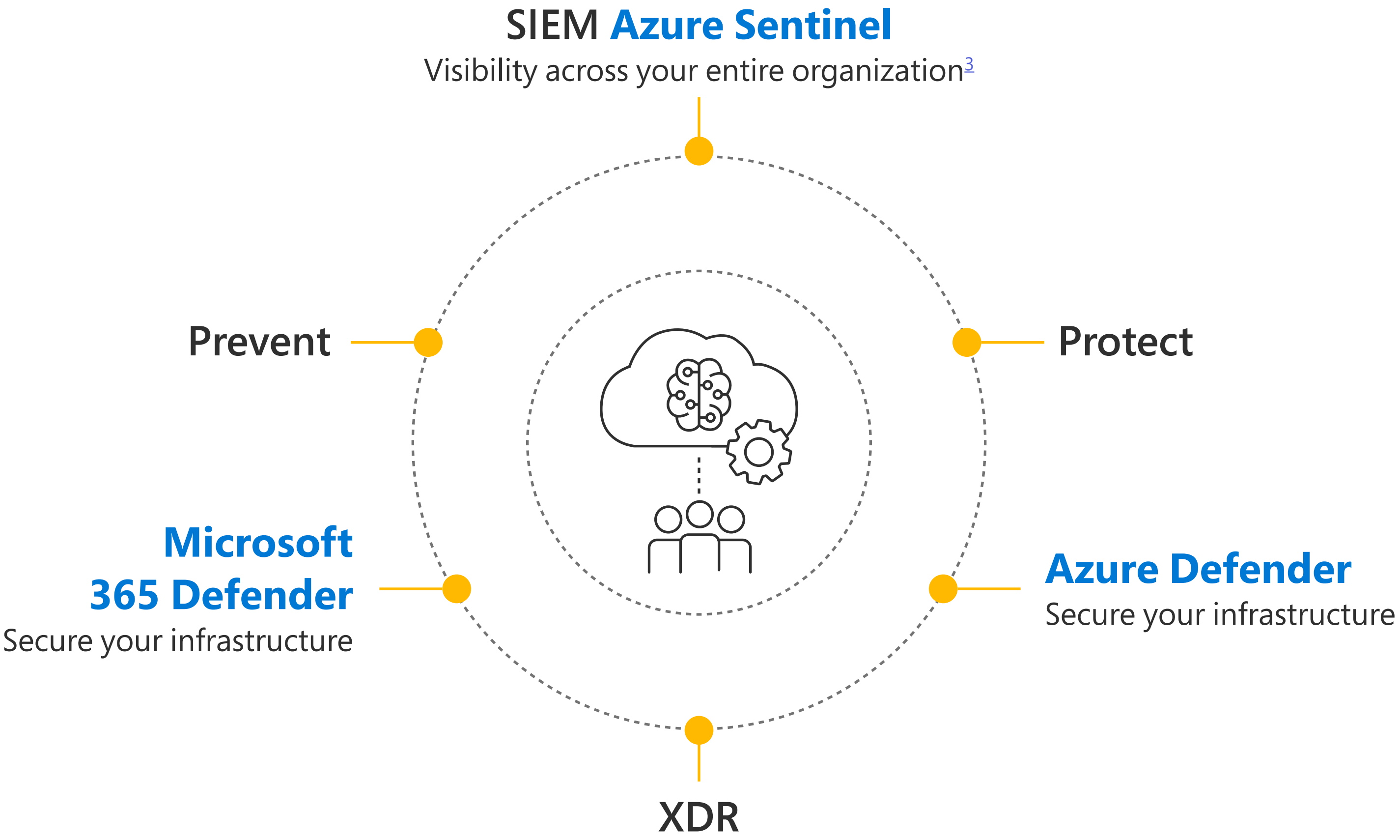
Leveraging native integrations with Microsoft Defender tools and Azure services such as Log Analytics and Logic Apps for analysis and automation capabilities, Azure Sentinel allows organizations to ingest, correlate, and analyze security signals from across the enterprise.

The ability to leverage elastic compute and storage capabilities inherent in Azure for data-intensive applications such as SIEM is a significant advantage over premise-based log analysis solutions. Additionally, Azure Sentinel can make use of infrastructure as a service (IaaS) and platform as a service (PaaS) available in Azure to deliver capabilities like workflow automation and long-term log retention that are typically provided as add-on services from other SIEM providers.

Azure Sentinel unified integration

Azure Sentinel integrates with Microsoft 365 Defender and Azure Defender to provide a unified way to manage risk in your digital landscape under a single umbrella. Incidents, schema, and alerts can be shared between Azure Sentinel and Microsoft 365 Defender, providing a holistic view with seamless drill down for context.[2]


SEIM | Azure Sentinel
Identities
Endpoints
Apps
Email
Docs
Cloud Apps
Microsoft 365 Defender
SQL
Server VMs
Containers
Network Traffic
Industrial IoT
Azure App Services
Azure Defender
XDR | Microsoft Defender


SIEM Azure Sentinel
Visibility across your entire organization[3]
Prevent
Protect
Microsoft 365 Defender
Secure your infrastructure
Azure Defender
Secure your infrastructure
XDR

Cloud SIEM Architecture

We take a two-sided view to the Azure Sentinel architecture. The first is the SIEM solution where security information and events are processed and analyzed. The second includes the multitude of data sources themselves. In our experience, addressing both the setup and operation of the SIEM solution, as well as a thoughtful approach to the data sources themselves, is critical to the success of any SIEM project.

Here we will look at both key aspects of SIEM architecture and at the considerations that organizations can take when approaching a project.

Core Azure Sentinel solution components

In this section, we provide guidance on deployment of the core Azure Sentinel solution components to be deployed in your Azure subscription.

Azure Log Analytics workspace

The first deployment prerequisite of Azure Sentinel is a Log Analytics workspace where all ingested data will be stored. A Log Analytics workspace is created within a specific Azure region and has a configurable retention period, defining how long data will be stored within the Log Analytics workspace (database). The default is 30 days, but this can be configured to as long as 730 days (2 years).

Various forms of data may be ingested into the Log Analytics database. Data sources include a wide variety of structured data such as system information from Azure Monitor Agents (AMAs) or Microsoft Monitoring Agents (MMAs) installed on Windows or Linux network endpoints,[4] application programming interface (API) integrations, and Azure PaaS services.

Log Analytics is a component of overall Azure Sentinel cost and is calculated based on the volume of ingested data and the data retention period. Special consideration should be paid to the extended retention period, as certain event tables might only contain system performance metrics or verbose logging of services, which may not be ideally suited for analysis within an SIEM solution. Data unrelated to security monitoring may not be worth storing over a long period of time when balanced against ingestion costs. Conducting a thorough analysis of the incoming data and aligning to organizational compliance policies will determine if raw data must be kept online in Log Analytics or if alternative storage options are possible. Alternate solutions exist within the Azure ecosystem to store raw data in cheaper storage options, where required.

There are a few initial best practices to follow when configuring Azure Log Analytics for use with Azure Sentinel:

- In multi-region architectures, deploy your Log Analytics workspace in an Azure region that will minimize the egress cost of data transfer between regions. In complex architectures with multiple Azure Sentinel instances, initial consideration should be paid to the region where most data are produced and consumed to avoid data export charges when providing Azure Sentinel with data from disparate Azure regions. In most cases, data export charges between Azure regions are usually lower than the price difference for Log Analytics between regions. Export charges between regions for Azure resources are only applicable to IaaS services (virtual machines [VMs]) and not to Azure PaaS services.
- Limit the number of Log Analytics workspaces, where possible. Understanding the relationship between security and operational data early in the project, and how each will be ingested, can save data ingestion charges at later dates.
- Implement a comprehensive role-based access control (RBAC) strategy for Log Analytics access early in the project.
- Configure Azure Sentinel analytic rules for monitoring various parameters related to data ingestion and costs. Often analytic rule requirements are built based purely on security operations needs; however, analytic rules are powerful and can be configured to perform monitoring on operational aspects of Azure Sentinel itself.
- Data requiring longer retention periods can be stored in alternative solutions, such as Azure Data Explorer (ADX) or Azure Blob Storage.

Azure Sentinel benefits from the inherent elastic storage capabilities of Azure Cloud. As such, it can dynamically scale on demand to meet even the most demanding data ingest requirements. For larger enterprises—organizations that see more than 1 TB/day—Microsoft offers an optional dedicated cluster for Azure Sentinel within Azure's infrastructure. This can improve search performance and, depending on your configuration of Azure Sentinel workspaces, can provide cost savings and efficiencies.[5]

For organizations that need to keep data available for longer than 90 days in a cost-effective storage repository while still being able to perform real-time queries or Kusto Query Language (KQL), there is the option to use ADX, which is a big data analytics platform that is highly optimized for all types of logs and telemetry data analytics. It provides low latency, high throughput ingestions with fast queries over extremely large volumes of data. It is feature rich in time series analytics, log analytics, full text search, advanced analytics visualization, scheduling, orchestration, automation, and many more native capabilities.

Learn more about Microsoft ADX here: https://docs.microsoft.com/en-us/azure/data-explorer/

Case study–Global retailer

ACME Corporation, a large retailer, is currently using Azure Sentinel SIEM as its core cybersecurity analytics monitoring tool. There are several log sources running in Azure Cloud, including Azure PaaS and IaaS resources, on-premises infrastructure, and numerous SaaS applications used by the finance department. The current volume of ingested log is 125 GB/day and ACME Corporation is using capacity reservation for 100 GB/day for both Log Analytics and Azure Sentinel.

ACME Corporation is subject to Payment Card Industry (PCI) data security standard (DSS) regulatory compliance and, therefore, has a log retention requirement of 90 days online and 1 year offline. While investigating other options for extended retention, the company decided to extend the online retention period in Log Analytics to 365 days, paying an additional $3,500/month. Based on East U.S. Azure Region, the company currently pays a combined fee of $15,515/month.

Availability of less costly storage options such as Azure Blob Storage or Cosmos DB are good ones to consider to meet compliance requirements. Our experience from performing cost analysis exercises shows that most organizations below 100 GB/day of data ingestion often choose to retain data in Log Analytics, primarily to maintain advanced security capabilities present in Log Analytics and Azure Sentinel. Capacity reservations are of great benefit once your organization is beyond 100 GB/day, but for larger ingestion and retention requirements, alternative storage options should be considered.

Azure Sentinel

With Log Analytics deployed, the Azure Sentinel resource is available for configuration to perform SIEM functions. We will cover Azure Sentinel itself in greater depth further in this whitepaper.

Azure Logic Apps

Azure Logic Apps provides security orchestration and automated response (SOAR) capabilities in Azure Sentinel. Azure Logic Apps power "playbooks" and are, effectively, a sequence of procedures that can be run in response to a security alert. Playbooks can help automate and orchestrate response actions that would typically be undertaken by security analysts. These can be triggered manually or set to run automatically when specific alerts are triggered.

Azure Logic Apps is a great beneficiary of the capabilities of elastic compute and uses the power of the Azure Cloud platform to automatically scale and meet demand—you do not have to worry about the complexity of infrastructure capacity, hosting, maintenance, or availability for your workflows. It is highly likely that if an organization has workloads in Azure Cloud, Logic Apps are already used in automations for other services.

Azure Logic Apps comes with many different out-of-the-box connectors that enable organizations to easily create Azure Sentinel playbooks for automated workflows.[6] Azure Logic Apps pricing structure is based on the number of transactions (or executions) and the type of connector.

As a Microsoft Azure Cloud service, Azure Logic Apps runs under a consumption-based pricing and metering model. This means that the fees are related only to how many workflow actions Azure Logic Apps execute.

The monthly price for deploying and using Logic Apps to orchestrate security event response is generally not an significant factor in the total cost of running Azure Sentinel. The versatility provides organizations with a wide range of options for reporting, alerting, and orchestration involving Azure Sentinel alerts. Other enterprise-grade connectors for non-security applications can come with higher cost price tags, so evaluation on the volume of playbook runs should be undertaken before deployment.

Case study–Community college

EDU Research, a community college with 9,000+ employees, is currently using Azure Sentinel SIEM for security monitoring and automated security incident response. The following Azure Sentinel playbooks were configured using Azure Logic Apps, as part of EDU Research Sentinel alert rules response:

Azure Usage runs a daily query in the Log Analytics usage table and sends an email notification to the EDU SOC team with aggregate daily Azure ingest costs per log source.

Incident Notification runs when a Sentinel alert triggers and automatically opens a ServiceNow ticket with incident details.

Health Monitoring runs when an Azure Sentinel playbook fails and sends an email notification to EDU SOC team.

Running in the East U.S. Azure Region, EDU Research pays a total monthly aggregated fee for Azure Logic Apps of $95.00, which includes standard connector actions, built-in actions, and data retention fees. Relative to the value produced by automating response actions, the cost for building security workflow automations in Logic Apps is often negligible and a powerful way to drive business value from an Azure Sentinel deployment.

Data sources

We regularly encounter a common misconception among security executives and practitioners that Azure Sentinel can only be used for Azure Cloud resources. In fact, Azure Sentinel is successfully used to ingest and correlate data from a wide range of log sources located in a variety of cloud platforms (Azure, Amazon Web Service [AWS] and Google Cloud), on-premises network and compute infrastructure, 3rd party security tools, or software as a service (SaaS) applications. In addition to the growing set of out-of-the-box data connectors, the Azure Sentinel public community is regularly demonstrating new use cases and data connectors that expand the capabilities of the solution.

Case study–Transportation company

Wheels Transportation is a transportation and logistics company that has been in business for over 100 years. At the time of initial engagement, Wheels did not have any workloads in Azure, including common productivity tools such as Microsoft Office 365. Infrastructure was primarily on premise, including servers and network infrastructure.

The security operations team at Wheels had been using an on-premises SIEM solution that required regular capital expenditure on storage and server hardware, which had to be aligned to predicted events per second (EPS) ingestion requirements several years into the future. This EPS forecast was often missed, resulting in unbudgeted purchases of additional hardware and log ingestion licenses.

Wheels chose to deploy Azure Sentinel as its first Azure resource and primary SIEM solution because of the ability to operationalize log ingestion costs using public cloud. Although Wheels did not have any existing Azure Cloud infrastructure, the flexibility of the Azure Sentinel SIEM solution was the right fit.

The wide variety of potential data types as log sources means that the consideration paid to each different data type is important at the outset of an Azure Sentinel project. Azure Sentinel includes more than 100 connectors, out of the box, with the ability to create custom sources to meet individual requirements. We have collected a summary table of some of the more common data source types, with experiential commentary relevant for deployment teams configuring new data ingest sources.

To learn about the more than 100 connectors included with Azure Sentinel, go here: https://docs.microsoft.com/en-us/azure/sentinel/connect-data-sources.

Log ingestion method	Typical log sources	Experience
Microsoft native data connectors	Built-in Azure Sentinel data connectors for Microsoft native log sources, such as Azure Services (e.g., Active Directory, distributed denial of service [DDoS] protection, Kubernetes Service, Web Application Firewall [WAF]), Dynamics 365, Microsoft Office 365, Microsoft Defender security services.	● Several of these data sources are free, such as alerts from Microsoft 365 Defender or Azure Defender, while additional advanced hunting logs are paid. ● Azure Sentinel can natively ingest security alerts from Defender for Identity, Defender for Office 365, Defender for Endpoints, and Microsoft cloud app security.
Log Analytics agent	Windows and Linux machines deployed on-premises or in any other cloud environments. Microsoft Internet Information Server (IIS) Web Servers logs can be collected via this agent. Any logs from other applications running on the same machine where MMA agent is running. This is collected via MMA Custom Log settings.	● Windows events are typically very noisy, and it will require an advanced MMA/AMA agent configuration. Also filtering specific Windows event IDs level (done via Group Policy configuration) may be required on the source servers or via the AMA agent configuration. ● Windows Performance Counters can be collected via this agent, too. Depending on the sample rate interval, this can increase the volume of collected events, increasing the overall Azure consumption

Syslog log forwarder	Firewalls, intrusion prevention systems (IPSs), L2/L3 network devices, and others.	● Syslog data is collected in Syslog format and will require creation of log parsers in Azure Sentinel. ● All Syslog messages are stored in a single Log Analytics table (Syslog table). ● This method does not allow any control on the volume or type of log ingested.
Common event format (CEF) log forwarder	Some type firewalls, Software-Defined Wide Area Network (SDWAN), and Secure Access Service Edge (SASE) platforms CEF is an industry standard format. Microsoft provides the installation scripts and documentation for a Linux agent that can be deployed on the same machine where a Syslog agent runs.	● CEF has a standard schema used by many security vendors, allowing interoperability among different platforms. ● It does not require additional log parsing ● Many platforms, like firewalls, allow customization of CEF templates, which is a great tool to optimize the volume of ingested logs at the source point.
Logic App playbooks	PaaS and SaaS applications. An Azure Logic Apps playbook using a REST API call can be used to pull events from an application or tool.	● Using remote application REST API calls, the data connectors can be set up to extract specific events only, which is an effective tool for log optimization.

	This method is typically used for SaaS applications. Data is ingested in Azure Sentinel Log Analytics workspace and is placed in a custom table.	● Log ingestion is based on "pull," within a predefined time interval (no real-time capabilities). ● This relies on availability of Azure Sentinel playbooks; therefore, additional monitoring is required for playbook health and status. ● The customer has control over Log Analytics table schema definition.
REST API (Azure function)	SaaS applications. This method requires custom development using remote application REST APIs and Azure functions.	● Customer does not need to run a separate machine/VM. ● Logs are ingested in a Log Analytics custom table. ● Log optimization is dependent on remote application REST API. ● This is the Microsoft recommended method for log collection from custom log sources.
Logstash collector	Firewalls, IPS, network devices, and others. A Logstash collector needs to be deployed on-premises or in a cloud environment on a VM.	● Data enrichment (such as geo-location) can be done on collection point. ● This allows log optimization, by collecting only required log fields. ● Once ingested in Azure Sentinel Log Analytics workspace, data will be stored in a custom table.

		● Many parsers and data enrichment tools are available in open-source Elasticsearch-Logstash-Kibana (ELK) community. ● Microsoft recently announced that the Logstash collector method will be supported as well.
Azure diagnostics	Azure PaaS resources. Not always considered a separate log ingestion method, collecting events via Azure Diagnostics is applicable to Azure PaaS resources only. Turning Azure Diagnostics for Azure PaaS resources, the audit logs and metrics events can be collected and stored in an Azure Diagnostics Log Analytics table.	● The customer must create log parsers for each resource type. ● Data ingested via Azure Diagnostics is very noisy and will increase the overall Azure consumption. ● No data optimization can be done via this method. ● Microsoft provides in Azure Sentinel a set of standard data connectors (e.g., Azure Key Vault, Azure Kubernetes) for a few PaaS resources. ● If the customer does not have any compliance requirements for log retention, using Azure Defender for monitoring and protecting Azure PaaS resources, in general, is a more cost-effective solution for this situation.

File-based ingestion	Used for ingestion of data from files located on the same machines where Log Analytics Agent is running. This option uses the Custom Logs configuration of the Log Analytics Agent.	● Logs are collected in a Log Analytics custom table (_CL). ● Using the Log Analytics Agent configuration tool, a specific record delimiter and a path to the file can be used.
Amazon Web Services	Microsoft provides an AWS CloudTrail Azure Sentinel data connector out of the box. Collecting events from other AWS resources, a REST API function can be developed.	● Limited to AWS CloudTrail only (default). ● Data is placed in AWS CloudTrail table, and it is already parsed.
Threat intelligence platforms: Structured Threat Information Expression (STIX) and Trusted Automated eXchange of Indicator Information (TAXII)	Available from external Threat Intelligence Platforms and from STIX/TAXII feeds, both open source and premium, these can enrich Azure Sentinel with additional indicators of compromise (IoC) such as known malicious internet protocols (IPs) or domain name service (DNS) names. STIX/TAXII is an ingest protocol unique to importing security intelligence data.	● Employ either the Azure Sentinel Threat Intelligence Platforms data connector or Azure Sentinel TAXII data connector. ● This requires one or more external Threat Intelligence Platforms and/or STIX/TAXII feeds.

Microsoft provides a set of guidelines for vendors and Azure Sentinel community for development of new data connectors.

Azure Sentinel uses Azure Log Analytics as the backend for the log storage and querying capabilities through KQL. A wealth of information is available from various log sources stored in Log Analytics as “tables.” There are a variety of default tables, though not always populated with data while others created as specific Azure Sentinel data connectors are enabled and configured. Another range of tables, not covered in the list following, are represented by custom logs that can be used to ingest logs from custom applications that fall outside the scope of standard SIEM log sources.

Azure Sentinel table	Description	Log sources	Relevant data	Billable
AuditLogs	Azure Active Directory activities audit such as creation/modification of users, groups, applications	Azure Active Directory	Account, location, activity	Yes
AWSCloudTrail	AWS CloudTrail log entries	AWS CloudTrail	Account, location, activity	Yes
AzureActivity	Azure activity such as creation/ modification/deletion of Azure resources, policy updates	Azure	Account, activity	No
AzureDiagnostics	Storage of diagnostic logs for Azure resources (resources must be configured to send the diagnostics logs to the specific Log Analytics workspace)	Azure Resources	PaaS diagnostic data	Yes
AzureMetrics	Provides storage of metrics recorded by various Azure resources	Azure Resources	Metrics	Yes
CommonSecurity-Log	Logs from security devices logging via syslog using CEF	Security Devices	Source, destination, protocol, action	Yes
ComputerGroup	Information on computer group membership (as configured in the Log Analytics workspace data collection)	Azure Active Directory	Account, location, activity	No

DnsEvents	Microsoft DNS events (registrations, configuration changes). Note: DNS queries outside the authoritative zone are not recorded	Microsoft DNS	DNS registrations, failures	Yes
DnsInventory	Log DNS records created on the DNS zone	Microsoft DNS	DNS records	Yes
Event	Windows event log entries (excluding Security event log)	Windows event logs	Errors, warnings	Yes
Heartbeat	MMA heartbeat	MMA agents	MMA health	No
McasShadow-ItReporting	Microsoft Cloud App Security (MCAS) Shadow IT information: records of access to applications typically used in "shadow IT" (filesharing, meetings)	MCAS	Application used, compliance	Yes
NetworkMonitoring	Network information on the monitored resources	Azure Active Directory	Account, location, activity	No
OfficeActivity	Office 365 activity: Exchange, SharePoint, data loss prevention (DLP), OneDrive	Office 365	Office 365 user and admin activities	No
Operation	Records related the functionality of monitoring agent logs (data collection, availability, issues)	MMAs	Status of agents	Yes

Perf	Windows and Linux performance counters collected by MMA	Windows and Linux performance counters	Performance counter	Yes
ProtectionStatus	Azure Security Center records related to the status of endpoint protection solution on monitored endpoints	Azure Security Center (ASC)	Status of endpoint protection	Yes
SecurityAlert	Alert details (Azure Sentinel, Azure Security Center (ASC), MCAS, Microsoft Defender for Endpoint (MDE), Active Directory Identity Protection	Azure Sentinel, ASC, MCAS, ATP, ADIP	Alert details	No
SecurityBaseline	ASC records related status of monitored endpoints versus configured policies for security baseline (e.g., levels of patching)	ASC	Status of updates versus security baseline	Yes
SecurityBaseline-Summary	ASC records with statistics for the monitored endpoints related to compliance with configured policies	ASC	Policy compliance stats	Yes
SecurityDetection	Microsoft Defender ATP logs for potential security issues detected on the monitored endpoints	Microsoft Defender for Endpoints	Potential security issues	Yes
SecurityEvent	Window Security event logs entries	Windows Security Event log	Account, source, activity	Yes

Security Incident	High-level details of security incidents created in Azure Sentinel or other M365 security tools	Azure Sentinel M365 Security tools (if "create incidents" is selected)	Incident description, severity, source app	No
SigninLogs	Azure Active Directory sign-in logs	Azure Active Directory	Account, source, location, activity	Yes
Syslog	Logs from syslog devices	Syslog-capable devices	Event, account, source, destination, action	Yes
ThreatIntelligenceIndicator	Used for ingestion of threat intelligence data from supported providers (e.g., MISP, MineMeld)	Various threat intelligence (TI) sources	Malicious IP, host, URL, Hash	Yes
Update	ASC missing/required updates (Windows, Linux)	ASC	Computer, update	Yes
UpdateSummary	ASC records with the status of current updates for the monitored endpoints	ASC	Computer, update	Yes
W3CIISLog	Microsoft IIS logs	Microsoft IIS logs	Source, destination, universal resource locator (URL), status code	Yes
WindowsFirewall	Microsoft Windows Firewall log entries (firewall running on endpoints)	Microsoft Firewall logs	Traffic allowed and traffic dropped on endpoints	Yes

Implementing A New Azure Sentinel Solution

With an initial view to the key components of Azure Sentinel, including solution components and data sources, we will provide an overview of recommended approaches to deploying your new Azure Sentinel environment.

Project resourcing

Project planning

Duration and complexity of an Azure Sentinel deployment project will vary depending on a variety of factors. Some key variables that should be captured at the project planning stage, which will affect project durations, include:

- Access to log sources and owners
- Types of log sources (i.e., standard data connectors versus development required)
- Complexity of Azure architecture (i.e., multiple tenants, cross-tenant searching)
- Requirement for custom SOAR automation playbooks
- Azure cost assessment and optimization

Key roles required for a successful Azure Sentinel deployment are as follows:

- Project manager
- Security architect
- Cloud engineer
- Engineering–systems owner
- Engineering–SIEM
- Business analyst
- Security operations
- Developer
- Compliance manager

Project manager

Experienced project management staff with Project Management Professional (PMP) and Information Technology Infrastructure Library (ITIL) backgrounds are recommended, as stakeholder management requirements can be quite broad. Azure Sentinel projects will involve input and work effort from teams supporting both cloud and on-premises infrastructure, end-user-facing services such as SaaS applications and workstations, as well as mission-critical server infrastructure.

Cost impacts from each of the various log sources must be quantified prior to the project, with variances tracked as new log ingestion begins. Change management from existing security tools is a critical factor to ensure business continuity and cyber defenses are not impacted.

Security architect

Questions like *"what log sources?"* and *"why?"* are important to analyze at the early stages of an Azure Sentinel project, taking a clear risk-based approach. There are numerous methods to gain security visibility to assets in the organization's information technology (IT) environment; however, log ingestion from these sources must always be accompanied by analysis of the cost impact of ingestion and analysis of the data.

The Azure Sentinel environment will contain highly sensitive data, and appropriate role-based access control must be applied to the Azure resources in scope of the project. The security architect will have responsibility for the security design of the Azure Sentinel solution.

Cloud engineer

Azure Sentinel will likely be one of many services running in your organization's Azure tenant, and determining resiliency requirements, Azure regions, data residency, and required tagging or templates applicable to Azure Sentinel will be the domain of the organization's Azure Cloud engineer/administrator.

Engineering – systems owner

Configuring log sources to send data to Azure Sentinel is often one of the more time-consuming activities in an Azure Sentinel deployment, particularly in complex organizational structures. Log structuring and format will vary from source to source, and organizational owners of assets such as SaaS applications, workstations, servers, cloud endpoints, and security infrastructure are often dispersed.

Subject matter experts (SMEs) and asset owners with the administrative ability to provide samples of logs and configure log-forwarding parameters on each asset will be required to dedicate effort to working with the project team to ensure data is sent to Azure Sentinel.

Engineering–SIEM

The SIEM engineer(s) are responsible for configuring Azure Sentinel, including Log Analytics, Logic Apps, workbooks, and playbooks. Working with the security architect, systems owner, and project manager, the SIEM engineer will be responsible for the following high-level tasks:

- Initial configuration of the Azure tenant, including provisioning required resources, assigning access roles, and configuring workspace parameters such as log retention, resource tagging, and blueprints.

- Deployment and configuration of syslog/CEF log collection agents in appropriate locations to collect logs from on-premises devices. This step will also include provisioning appropriate servers to run syslog or other log collection solutions.

- Working with system owners to enable log forwarding and configuring any required parsing of log data in Log Analytics.

- Working with security operations to create and deploy KQL analytic rules to provide detections for SOC/computer security incident response team (CSIRT) use.

- Tuning of alert rule parameters, including thresholds, detection logic, and assigned criticality levels to minimize false positives and appropriately identify potential attacker behavior.

- Working with project management for stakeholder engagement, creating workbooks for data visualization, and dashboarding of Azure Sentinel content. Security operations is likely to be the primary consumer of workbooks; however, data visualizations in Azure Sentinel may be created and customized for a wide audience of stakeholders; therefore, appropriate requirements gathering through project governance is advised.

- Creating automated workflows using Azure Logic Apps is recommended at the project phase. Working with security operations to document response workflows for various incident types and provisioning playbooks to automate response actions per incident type are effective ways to provide immediate value from the Azure Sentinel implementation.

- Working with systems owners for other IT systems such as IT service management (ITSM) and helpdesk tools to build integrated ticket workflows are recommended at the project phase. Azure Sentinel has capabilities to integrate with platforms such as ServiceNow or other ITSM API-enabled tooling to provide workflow automation for incident handling.

Network engineer

Network engineering resources will be required on demand to apply changes to firewalls or network infrastructure to facilitate log forwarding from data sources to Azure.

Business analyst

Capturing and evaluating the budget and resource impact of Azure data ingestion and various data source ingestions are important aspects of an Azure Sentinel project. As a cloud-native SIEM, organizations are shifting costs from capital expenditure to operational expenditure, and cost forecasting for the Azure Sentinel solution is recommended at the project stage.

As part of the Azure Sentinel project, the business analyst (BA) should be able to provide an Azure cost analysis of each technical requirement. In conjunction with the SIEM engineer, the BA should model this expected Azure cost impact over time as changes to the IT environment are seen. An effective risk-based security program will be able to quantify the risk mitigation effects of security controls as related to the mitigation cost for a specific control.

Security operations

Security operations stakeholders in the Azure Sentinel project are primarily assigned to document the detection, alerting, and threat hunting requirements of the solution. While the security architect and SIEM engineer are able to provide access to security-relevant data and present these back to security operations in the form of alerts, incidents, dashboards, or reports, it is ultimately the responsibility of security operations as the end consumer of the service to articulate the build requirements

Developer

Developer resources are often the most overlooked requirements for an Azure Sentinel project. Programming languages such as C# and Python and developer effort are often required to obtain data from log sources such as some SaaS applications and can be leveraged to great effect by Azure functions.

Compliance manager

If your organization has legal, regulatory, or industry-specific compliance requirements that will need to be satisfied by Azure Sentinel, the interaction between the core Azure Sentinel team and compliance manager is mandatory. Decisions such as log retention period, custom workbooks, and compliance reporting mandates are overseen by this resource.

Benchmark project effort and duration

Included here are high-level benchmarks for full-time equivalent (FTE) requirements and project duration for a sample 5,000-employee organization. Actual effort is highly variable depending on organization-specific factors.

Sample organization, 5,000–10,000 employees			
Resource type/ function	**Benchmark effort (FTE)**	**Benchmark duration (days)**	**Key tasks**
Project manager	1	60	• Project planning • Stakeholder engagement • Resource planning • Change management • Project governance
Security architect	0.5	60	• Data ingestion strategy and methods • RBAC controls • Compliance requirements • Access control

			• Identify existing security controls • Identify log sources to be ingested into Azure Sentinel • Provide expertise in use-case creation • Identify gaps in data ingestion • Provide consulting on governance/ risk/compliance
Azure Cloud engineer	0.1	15	• Managing administrative permissions/RBAC in Azure tenant • Service resiliency • Provision Azure resources • Configure Azure Active Directory service accounts • Configure Azure Active Directory groups and assign membership
Engineering–systems owner	0.2 (per log source)	30	• Configuring log forwarding on assets required to send data to Azure Sentinel • IT sysadmin: deploying monitoring agents on endpoints • Deploy on-premises Azure Sentinel log collector • Provide expertise around application logging capabilities and potential use cases • Configure application logging • Assist in validation of alert rules and SOAR playbooks
Engineering–SIEM	1	90	• Developing KQL detection rules • Developing workbooks for data visualization • Creating Azure functions for data retrieval, moves

			- Creating playbooks for workflow automation
Network engineer	0.5	30	- Assist with required connectivity between Azure Sentinel and logging sources
Business analyst	1	30	- Azure cost analysis - Documentation of business workflows to be automated with playbooks
Developer	0.5 per custom log source	30	- Development of new custom data connectors to ingest data to Azure Sentinel
Security operations	1	60	- Documentation of detection use cases for detection rules - Documentation of detection parameters (e.g., alert thresholds, exclusions, threat intelligence sources)
Compliance manager	0.1	15	- Provide recommendations on the compliance requirements applicable to Azure Sentinel SIEM - Review and provide feedback on Azure Sentinel components designed and built for compliance purposes

Architecture planning and considerations

The following factors affect the initial architecture for deployments of new Azure Sentinel instances or the migration from existing SIEM platforms.

Data residency requirements

There are over 63 Microsoft Azure regions spanning 140 countries with over 30 of them supporting Log Analytics workspaces. The geographical regions include North America, South America, Europe, Middle East, South-East Asia, and Africa. While additional support is added on a regular basis, Azure Sentinel is not available in all regions.

See [Quickstart: Onboard in Azure Sentinel | Microsoft Docs] for the current list of Azure regions supported by Azure Sentinel.

Depending on the type of business and customer residency, organizations may have compliance restrictions related to the logged data. The compliance regulations are not always very clear, and organizations may choose to use a local region to avoid further complications due to changes in legislation or auditing processes.

The selection of the region also carries implications for Azure Sentinel and Log Analytics costs as well as the availability of resources for the specific region. Regions such as East U.S. can offer a significant cost advantage versus other regions. For example, East U.S. offers a 17% discount compared with Canada Central. Depending on the volume of log ingestion, the discount can be significant, therefore, it is recommended that a project team obtain organizational requirements relating to data residency.

Case study–Technology company

TechSupport is an IT service provider with services on various IT platforms for their local market in South Africa. TechSupport was looking to deploy an Azure Sentinel instance to improve security monitoring of their internal infrastructure. Aside from Microsoft Office 365, the infrastructure was primarily on-premises, including firewalls, endpoint protection, and servers both on Windows/Linux and network infrastructure. After an Azure Sentinel deployment, the initial daily Azure consumption associated with Sentinel was around 15 GB/day after log optimization.

It was determined that TechSupport had no data residency requirements for its corporate workloads; therefore, TechSupport decided to deploy Azure Sentinel in East U.S. Azure region. The Azure Sentinel monthly bill for this daily consumption was approximative $1,900 based on 3 months of online retention. For the same volume of data, if TechSupport had deployed to the South Africa North Azure region, the monthly bill was estimated to be $2,800, which would be 32% more expensive than East U.S. Careful consideration of solution requirements, including compliance and data residency, can provide substantial operational cost savings in the medium to long term.

Number of Azure Active Directory tenants

An Azure AD tenant provides identity and access management (IAM) capabilities for applications and resources used within an organization. An identity is a directory object that can be authenticated and authorized for access to a resource. Identity objects exist for human identities (e.g., employees) and non-human identities (e.g., computing devices, applications, and service principals).

While most organizations have a single Azure Active Directory tenant, some may have one or more through mergers and acquisitions or the need to segregate environments such as corporate versus production infrastructure. Each Azure tenant requires a dedicated, independently managed Azure Sentinel instance. Azure Lighthouse can provide cross-tenant management experience for unified platform tooling, management at scale, and increased visibility.

Additional resources on Azure Sentinel integration with Azure Lighthouse:

- Azure Sentinel and Azure Lighthouse (microsoft.com)
- Build a scalable security practice with Azure Lighthouse and Azure Sentinel
- Multi-tenant access for Managed Security service providers - Microsoft Tech Community

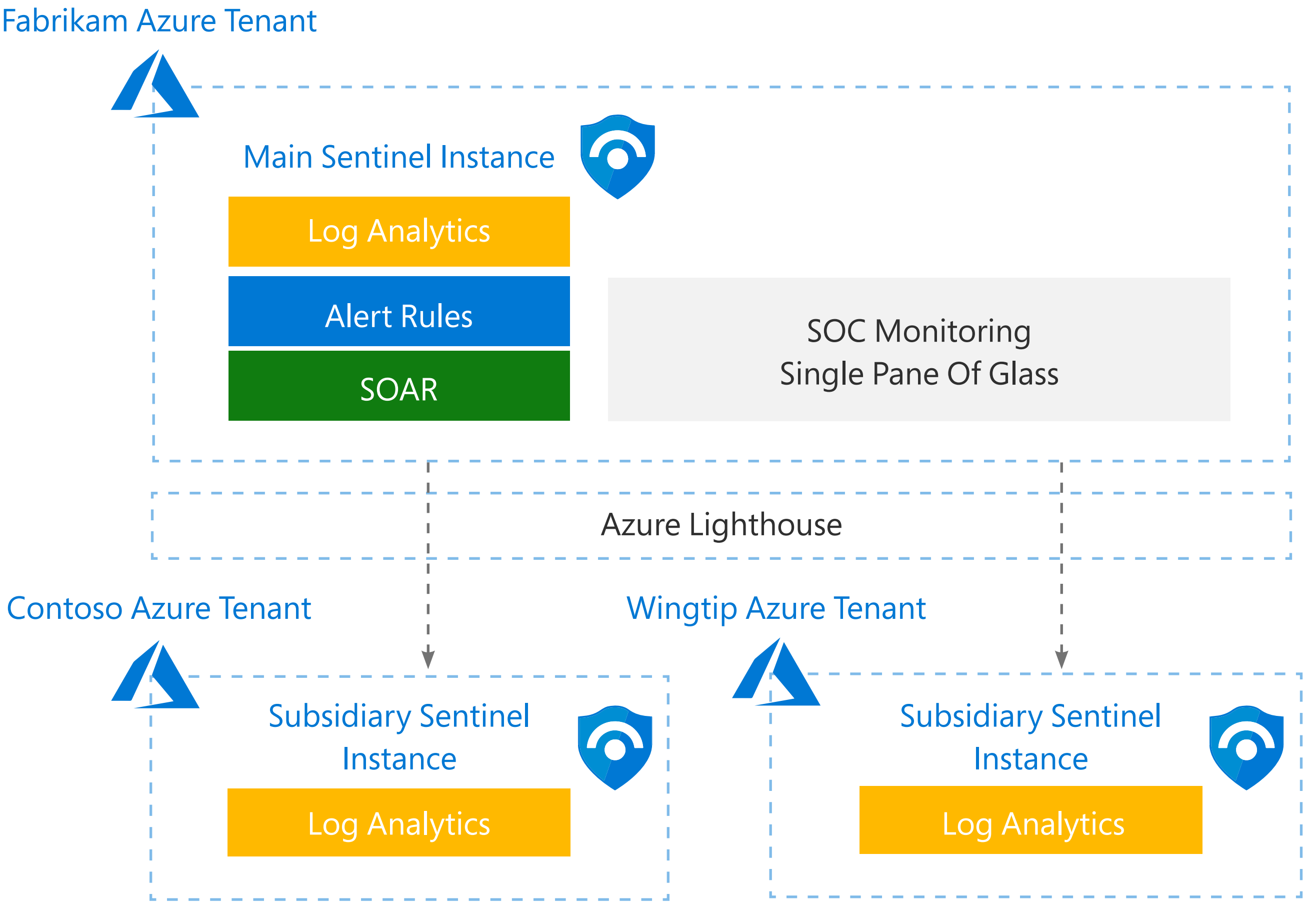


Fig. 1. A single Azure Sentinel instance can be used to monitor subsidiary Azure Sentinel instances across multiple Azure Active Directory tenants

Number of Azure subscriptions

A single Azure Sentinel instance can integrate data from multiple Azure subscriptions; however, some security or operational solutions may have restrictions on their logging capabilities that limit sending logging/diagnostics data to different subscriptions. For example, Microsoft Intune can only send audit logs to a log analytics workspace within its subscription. This is a limitation of that solution, not of Azure Sentinel. Such solutions require a custom connector that can be configured using a variety of methods.

Azure Sentinel can be deployed in existing subscriptions or in its own subscription without any implications for its functionality.

A dedicated subscription is recommended in the following situations:

- There is a need to clearly identify or segregate any costs associated with Azure Sentinel.
- Permissions need to be assigned at the subscription level to allow creation and management of various resources required for a full Azure Sentinel configuration. In a complex environment, this could be VMs, function apps, automation accounts, storage accounts, data explorer clusters, machine learning, key vaults, and databases.

If Azure Sentinel is deployed in multiple subscriptions, they can be managed centrally through regular assignment of Azure Active Directory roles. Azure Lighthouse is not required in this case, as it is designed to provide cross-tenant access.

Case study–K-12 school board

ABC School is a large school board in North America with more than 100,000 students and 7,000 staff members. ABC School has organizational separation between network operations, security operations, and server management teams, where each team has specific access rights to view and access resources (segregation of duties) and different cost centers.

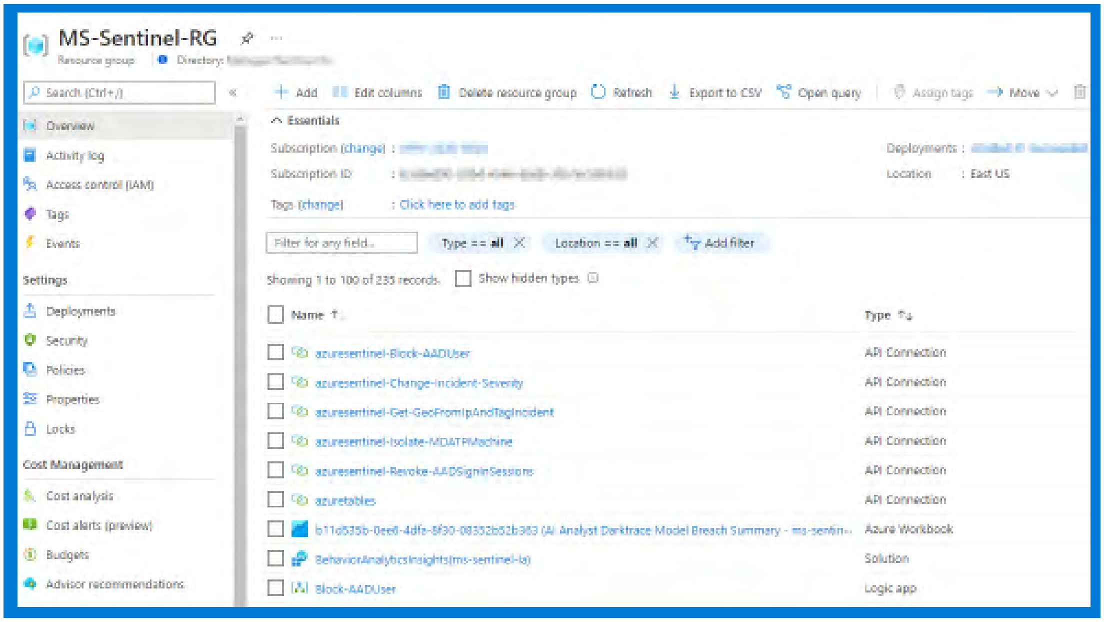
ABC School has decided to deploy Azure Sentinel in a new Azure subscription under the abcschool.com tenant, as Azure Sentinel was to be used only for security monitoring purposes, with limited access to Azure Sentinel granted to people outside of SOC. This approach allowed ABC School security team to have a separate Azure invoice assigned to the appropriate cost center.

Number of Azure resource groups

As is the case with most Azure resources, an Azure Sentinel Log Analytics workspace resides in a resource group. A resource group is a container that holds related resources for an Azure solution; in this case, it would be Azure Sentinel. Resource groups allow for granularity in assigning permissions and logical grouping of resources based on their purpose.

As a solution, Azure Sentinel will use multiple types of resources—some mandatory, some optional—such as Log Analytics workspaces, workbooks, Logic Apps, API connections, function apps, automation accounts, storage accounts, key vaults, application insights, VMs, and many others. In most cases, a single resource group is sufficient but in certain instances, such as those where different types of Azure Function Apps need to be used, the full solution may span multiple resource groups.

If a dedicated subscription is not practical, it is highly recommended to maintain all Azure Sentinel–related resources in a dedicated resource group.



MS-Sentinel-RG

Resource group | Directory: [redacted]

Subscription (change) : [redacted]

Subscription ID : [redacted]

Tags (change) : Click here to add tags

Deployments : [redacted]

Location : East US

Showing 1 to 100 of 235 records.

Name	Type
azuresentinel-Block-AADUser	API Connection
azuresentinel-Change-Incident-Severity	API Connection
azuresentinel-Get-GeoFromIpAndTagIncident	API Connection
azuresentinel-Isolate-MDATPMachine	API Connection
azuresentinel-Revoke-AADSigninSessions	API Connection
azuretables	API Connection
b11d535b-0ee6-4dfa-8f30-08352b52b383 (AI Analyst Darktrace Model Breach Summary - ms-sentin...	Azure Workbook
BehaviorAnalyticsInsights(ms-sentinel-la)	Solution
Block-AADUser	Logic app

Fig. 2. Example of resources used in an Azure Sentinel solution

Distribution of Azure PaaS resources

There is no cost for traffic that spans between Azure PaaS regions, but traffic egressed to non-Azure environments such as internet and on-premises virtual private network (VPN) incurs a bandwidth cost. When considering an Azure region for the Log Analytics workspace used for Azure Sentinel, the cost of transferring data out of other Azure regions or other cloud providers should be understood and taken into consideration as an additional cost.

For this reason, the preferred location should be in the region with the majority of log-generating Azure resources. Multiple Azure Sentinel instances can be deployed, but the reduced bandwidth costs should be weighed against the additional management complexity. In practice, we have found that the reduced bandwidth costs rarely justify the increased management costs for multiple Azure Sentinel instances.

Additional resources for Microsoft Azure bandwidth pricing:

- Pricing - Bandwidth | Microsoft Azure

Data segregation requirements

Some organizations have strict requirements on the accessibility of logging data between different business units due to legislative or regulatory compliance requirements or internal dictates. Permissions can be applied to specific types of logging data within a single Azure Sentinel instance, but for full, clear isolation, a dedicated Azure Sentinel instance should be considered.

A common scenario is a central SOC that requires visibility across the entire environment but also needs organizational units to be able to access the logging data from only their own resources. For example, a manufacturing business may have a dedicated Azure Sentinel instance that collects logging data from operational technologies (OT) devices and must provide access to analysts specialized in OT. These can include application developers, dedicated security analysts, and other specialized roles. By creating separate Azure Sentinel instances, it is possible to provide the OT teams full visibility to the logs from devices under their purview while maintaining full visibility for the central SOC.

Because the overall volume of logs will remain the same, there are no additional Azure ingestion costs for multiple Azure Sentinel instances.

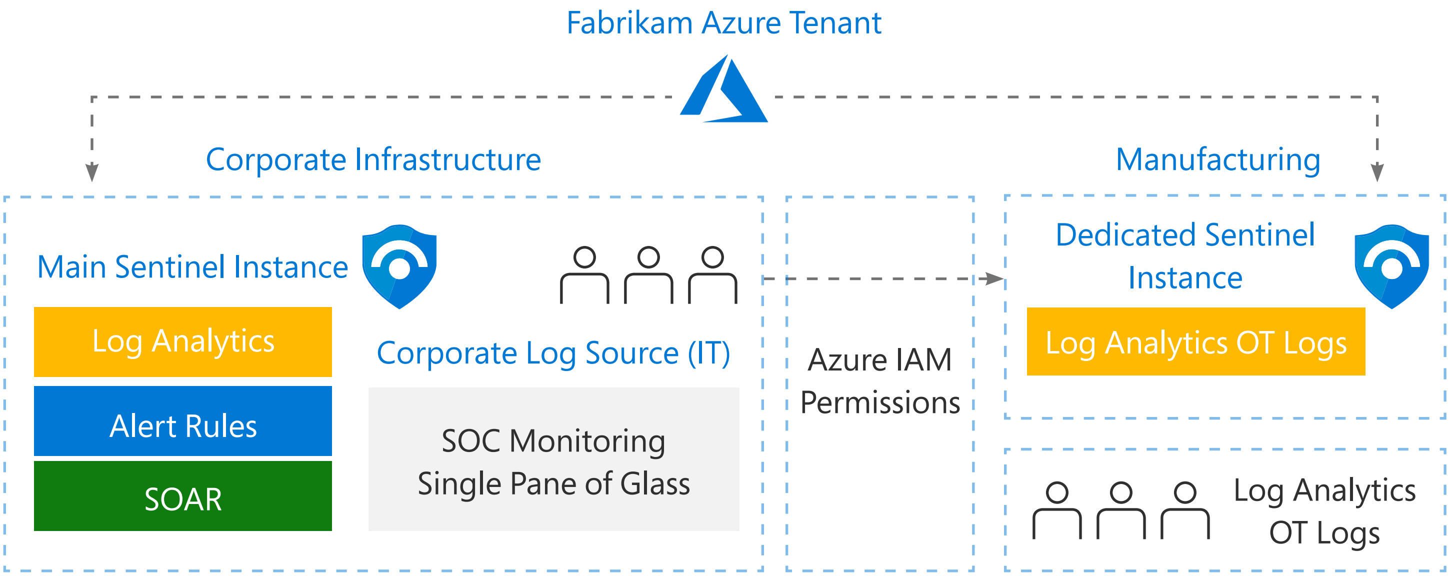


Fig. 3. Separation of Azure Sentinel instances for corporate versus manufacturing units

Complex organizational structures

SIEM deployments are typically driven by the IT security department to address specific security needs. An SIEM can be an expensive security control; therefore, it is common for multiple business units to contribute to the overall expense.

Based on the IT security tenet of "separation of duties," access to the analytical data and the security events generated will need to stay within the Chief Information Security Officer's control.

Various organizational units, such as human resources (HR), may require a level of access to specific dashboards or sets of data. Azure Sentinel provides the ability to assign table-level permissions and limit the level of access to the minimum required to perform requisite job functions.

Our experience shows:

- Organizations experience challenges in determining the full range of teams that need access to Azure Sentinel at the outset of a project. Change management for this fact should be considered and accounted for.
- While various business units may require log collection and storage, the responsibilities around the logging infrastructure and coverage of the operational costs are not always identified.

Role-based access control (RBAC) requirements

Azure Sentinel provides an extensive list of Azure built-in roles that can be used to provide granular access based on job requirements and permitted level of access. Part of these roles are three dedicated Azure Sentinel roles:

1. **Azure Sentinel Contributor.** Can perform all engineering-related configuration, such as creating alert rules, configuring data connectors, and additional similar tasks.
2. **Azure Sentinel Reader.** Can query the log data stored in Azure Sentinel but cannot modify any settings.
3. **Azure Sentinel Responder.** Can query the log data, can view and update incidents raised by security rules, but cannot modify alert rules.

In addition to Azure Sentinel-specific roles, there are additional roles required to fully configure and use Azure Sentinel:

- **Logic App Contributor.** For creation and management of SOAR playbooks.
- **Log Analytics Contributor.** For creation and management of workbooks (Azure Sentinel dashboards).
- **Log Analytics Reader.** For accessing/reading workbooks.

Depending on the requirements, custom RBAC roles can be created and associated with the Azure Sentinel instance.

The roles can be applied at the subscription or resource group level—the recommendation being to provide the minimum permissions required to perform the job.

Assigning the Azure Sentinel Reader role provides access to all the logs by default, but custom roles can be assigned on a "per table" basis to limit visibility to just specific Azure Sentinel tables.

For Azure resources, resource-specific permissions can be applied using a resource-context RBAC that can provide multiple options to identify specific resources and grant permissions to the appropriate users or groups.

If access to data is needed for only a subset of the data in Azure Sentinel tables, there are options to provide read-only dashboards or present the data based on custom queries via Microsoft Power BI.

Additional resources about roles and permissions:

- Azure built-in roles - Azure RBAC | Microsoft Docs
- Azure custom roles - Azure RBAC | Microsoft Docs
- Table-level RBAC in Azure Sentinel - Microsoft Tech Community
- Permissions in Azure Sentinel | Microsoft Docs
- Controlling access to Azure Sentinel data: Resource RBAC - Microsoft Tech Community
- Manage access to Azure Sentinel data by resource | Microsoft Docs

Ingestion of operational logs versus security logs

Organizations may also collect performance and operational logs for monitoring purposes and day-to-day operations analytics, such as various performance counters and diagnostics logs. These types of logs fall under the "availability" umbrella of the confidentiality–integrity–availability (CIA) triad.

This type of data sometimes overlaps with the security-related logging data, and from a logistics perspective, all the logging data may be aggregated into a single log analytics workspace. Enabling Azure Sentinel will apply the additional costs to the full set of data, so if the operational data represents a significant proportion of the overall logs, then a separation of operational and security logs is recommended. Some organizations consider that the additional visibility or correlation across the entire data set justifies the additional expense.

Our experience shows:

- Existing Log Analytics workspaces, storing both security and operational logs, can increase Azure Sentinel costs. Often the costs associated with Azure Sentinel are intended to be borne by a security organization, but operational logs in Log Analytics may result in IT operations costs being hidden within security spend.
- After cost versus benefit analysis, organizations often decide to maintain the aggregated operational and security logs together to take advantage of the potential correlation capabilities, with the operations department contributing to the operational costs through internal cost allocation mechanisms.

Case study–Manufacturing company

Fabrikam Inc. is a large international manufacturing company with offices and datacenters on three continents. Fabrikam is in the process of migration all internal workloads to Azure Cloud, with substantial Windows and Linux server estate located on premises. As part of its Azure migration strategy, Fabrikam Inc. has decided to use Azure Monitor for server monitoring.

The legacy environment included a total of 780 servers, both Unix and Windows, located in Azure and within the three datacenters on premises and logging to a QRadar SIEM solution. The Fabrikam server operations team had deployed MMA and Operations Management Suite (OMS) agents to all servers. The team has determined that the total daily Azure Monitoring logging was approximately 36 GB/day.

Fabrikam corporate security decided to migrate the on-premises QRadar SIEM to Azure Sentinel. During the initial setup of Azure Sentinel, the team enabled security events collection for all remote MMA agents, using common-level event streaming. This configuration increased the log collection by adding another 29 GB/day in SecurityEvent Log Analytics table.

Operational logs can satisfy key business and technical requirements; however, in our experience, projects that account for operational logging requirements as a separate cost item from security logging requirements are more easily able to demonstrate return on investment for the SIEM project.

Estimation of log ingestion volume and pricing model

The estimation of future log ingestion is a difficult exercise because, during an Azure Sentinel deployment, organizations are typically trying to include many log sources that are new to them and have not been included in a prior SIEM solution, or they are trying to include the full range of logs from existing log sources that were previously only logging partially (or not at all).

The additional difficulty arises from the fact that each organization has a different distribution of log source types that are based on their industry, number of users, internet presence, distribution of their assets (on-premises versus cloud versus SaaS), and compliance requirements.

Selecting a sample of log sources and configuring them to send full logs for a typical day (or a typical week) is often the most precise way to estimate the log ingestion volume.

With all the caveats mentioned previously and if a testing sample is not possible, an estimate based on more than 100 successful Azure Sentinel deployments indicates that one can expect around 25 MB/user/day. For example, an organization with 500 employees, would generate roughly 12 GB of logs per day. Again, this is a very high-level estimate as there are many factors that can affect the volume of logs that are not directly attributed to employee head count.

Microsoft has recently released an Azure Sentinel and Log Analytics calculator that takes into consideration different log source types, a size estimate of each event type, and Azure region and SOAR automation playbooks. The calculator is a useful tool to estimate both GB/day and EPS, which will provide the customers a good insight into Azure costs prior to deployment.

Based on the expected logging volume, a pricing model can be selected to take advantage of the commitment tier discounts offered by Microsoft, especially when logging volume is estimated to be consistently over 100 GB/day.

Architecture design output

At the conclusion of the high-level design phase, the following items should be decided and documented:

- Azure region used for the Azure Sentinel Log Analytics workspace (or workspaces, if more than one region is to be used).
- Azure subscription, resource group, and log analytics workspace, including naming convention and tags.
- Azure Active Directory groups and the RBAC to be applied to each.
- Log sources in scope (on-premises, cloud, SaaS).
- Azure Sentinel data connectors required to ingest in-scope log sources.
- Custom data connectors to be developed (if applicable).
- On-premises syslog collectors (quantity, location, operating system [OS] type, any additional configuration).
- Initial list of use cases to be implemented.
- Internet/VPN/LAN/WAN connectivity between log sources and Azure Sentinel.
- Estimated log ingestion volume (GB/day).
- Data retention policy (in days or months).
- Pricing model (pay-as-you-go) versus reserved capacity, depends on the estimated log ingestion volume.

Deploy

Once the high-level design is completed, the provisioning of Azure Sentinel and the related resources can be initiated.

Azure resources

As an analytical solution built around Log Analytics workspace and Logic Apps, Azure Sentinel requires the following resources to be created:

- Subscription (if a dedicated subscription will be used)
- Resource group(s)
- Log Analytics workspace(s)
- Automation rules/playbooks
- Alert rules
- Workbooks

An Azure global admin or an Azure security administrator is required to create these resources and enable Azure Sentinel for the selected Log Analytics workspace.

During the deployment, in addition to the Azure region, some basic configuration will be required such as log retention (the default is 90 days) and selection of a pricing model.

Automation rules, playbooks, alert rules, and workbooks are typically created gradually following the onboarding of various log sources. These resources are part of the ongoing SIEM tuning and maintenance and should follow the typical change control procedures.

Azure Sentinel provides hundreds of alert rule templates along with many workbook (dashboard) templates and hunting scripts (threat hunting scripts that are typically used ad-hoc and not as alerting rules). The templates can be used to activate/deploy schedule alerts, create customized dashboards, and perform threat hunting activities.

Methods of deployment:

- Manual. Using the Azure portal, the administrator manually configures the Azure Sentinel resources. Any manual process has the inherent risks of human operator error, lack of compliance with potential change control procedures, and undocumented changes.
- Automation tools. Azure Sentinel resources support several infrastructure-as-code tools, such as Hashicorp Terraform, that can provide consistency to processes. Microsoft provides a PowerShell automation library called Az.SecurityInsights that can be used to script a wide range of Azure Sentinel–related deployment tasks. The Azure Sentinel community provides additional resources, such as the AzSentinel PowerShell library and a wide range of Azure Resource Manager (ARM) templates for a variety of Azure Sentinel playbooks, alert rules, and workbooks.

Optional Azure resources:

- Service principal names (SPNs). SPNs are typically used in automation playbooks for authentication when accessing various resources for retrieval of log data or execution of automation tasks. SPNs should be provided with the minimal permissions required to perform their tasks.
- Storage accounts. Storage accounts can be used for temporary storage of analysis data, long-term log retention, and other tasks that require basic storage outside Log Analytics.
- Function apps. As serverless compute resources, function apps can perform a wide variety of tasks related to log collection and automation tasks. Only one type of compute platform can be used for one resource group. If, for example, both Python and .Net function apps are required, they need to be deployed in different resource groups. Some of the built-in Azure Sentinel data connectors require the deployment of function apps.
- Key vaults. Typically used for secure storage of secrets used by various log collectors.
- Event hubs. In certain designs that require integration with legacy SIEMs or third-party analytical platforms, Azure Event Hubs can be used to share analysis data.

None of the optional resources are required for the initial Azure Sentinel configuration, but they may end up being used as the complexity of the deployment increases and new automation tasks are required.

In our experience:

Azure Sentinel deployment projects can run into some foreseeable challenges that can be avoided with appropriate project management and advance planning. We have included a few key items:

- **Understand your data as it pertains to potential data residency requirements; there may be a need to choose one region over another.**
- **Clearly identify and coordinate with Azure administrators at the inception of your project.**
- **Keep the Azure administrators briefed on the scope and status of the project.**
- **Consider naming conventions, tagging, etc. Where none exist, plan to future proof the deployment.**
- **Run a risk assessment with your stakeholders about whether to create new log analytics workspaces (starting fresh) versus using the pre-existing legacy log analytics workspaces.**
- **Understand your planned automation tasks and their touchpoints to ensure that required SPN permissions are pre-staged (for example, what permissions are needed to enable or disable an Azure Active Directory user account?).**

Additional resources:

- New Year - New official Azure Sentinel PowerShell module! - Microsoft Tech Community
- Automating Azure Sentinel deployment using Terraform and PowerShell | Marius Sandbu (msandbu.org)
- Deploying and managing Azure Sentinel as code - Microsoft Tech Community

Log source onboarding

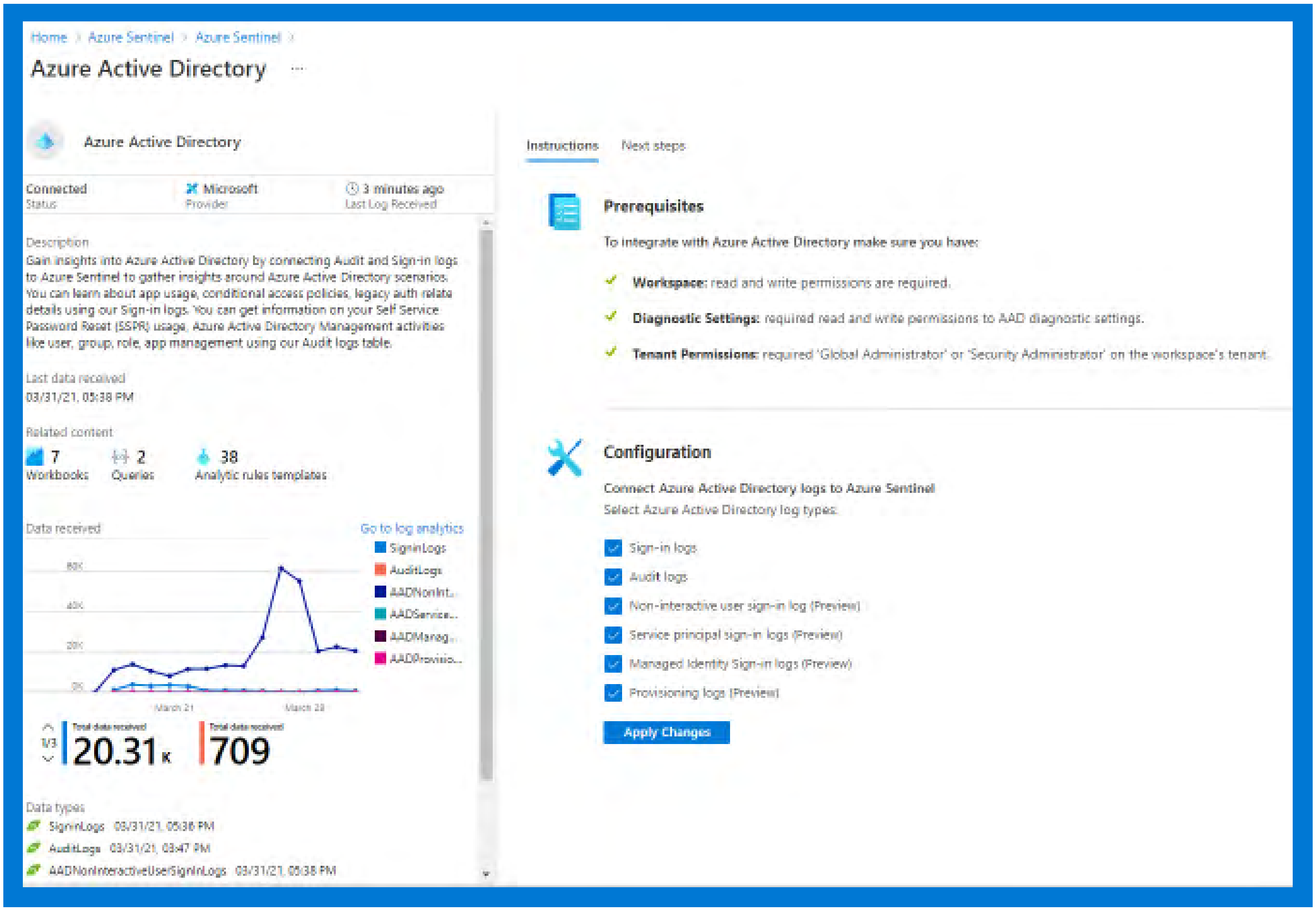
Azure Sentinel includes many data connectors for a wide range of log sources such as Azure (Azure Active Directory, PaaS), Microsoft 365 (Defender) solutions, non-Azure cloud (AWS), on-premises sources (e.g., firewalls, Network Access Control, VPN, LAN/WAN, Windows Active Directory, DNS), SaaS (multiple solutions), and threat intelligence feeds.

Depending on the type of log source, the onboarding process varies from just a few clicks of the mouse, deployment of MMA, or to more complex configurations involving the deployment of additional log collection resources such as Azure Sentinel playbooks based on Azure Logic Apps, Azure Function Apps, and vendor-provided log retrieval tools.

For each supported data connector, Azure Sentinel provides full instructions on how to onboard a particular log sources and, where applicable, with options to automate the deployment of the required data collectors and relevant log parsers.

Built-in data connectors

Azure Sentinel includes many connectors that can be deployed in a few clicks via the Azure Sentinel portal and the requisite RBAC permissions. This includes Azure Active Directory, Azure subscription activity, Office 365, and the whole family of Microsoft Defender products. New data connectors for other products are added on a regular basis. Consider the built-in data connectors over custom ones, where feasible, as they are fully supported by Microsoft and the Azure Sentinel community.


Home > Azure Sentinel > Azure Sentinel >

Azure Active Directory ...

Azure Active Directory

Connected
Status

Microsoft
Provider

3 minutes ago
Last Log Received

Description
Gain insights into Azure Active Directory by connecting Audit and Sign-in logs to Azure Sentinel to gather insights around Azure Active Directory scenarios. You can learn about app usage, conditional access policies, legacy auth relate details using our Sign-in logs. You can get information on your Self Service Password Reset (SSPR) usage, Azure Active Directory Management activities like user, group, role, app management using our Audit logs table.

Last data received
03/31/21, 05:38 PM

Related content
7 Workbooks
2 Queries
38 Analytic rules templates

Data received

Total data received
20.31K

Total data received
709

Data types
SigninLogs 03/31/21, 05:38 PM
AuditLogs 03/31/21, 03:47 PM
AADNonInteractiveUserSignInLogs 03/31/21, 05:38 PM

Prerequisites

To integrate with Azure Active Directory make sure you have:

- ✓ **Workspace:** read and write permissions are required.
- ✓ **Diagnostic Settings:** required read and write permissions to AAD diagnostic settings.
- ✓ **Tenant Permissions:** required 'Global Administrator' or 'Security Administrator' on the workspace's tenant.

Configuration

Connect Azure Active Directory logs to Azure Sentinel
Select Azure Active Directory log types:

- ☑ Sign-in logs
- ☑ Audit logs
- ☑ Non-interactive user sign-in log (Preview)
- ☑ Service principal sign-in logs (Preview)
- ☑ Managed Identity Sign-in logs (Preview)
- ☑ Provisioning logs (Preview)

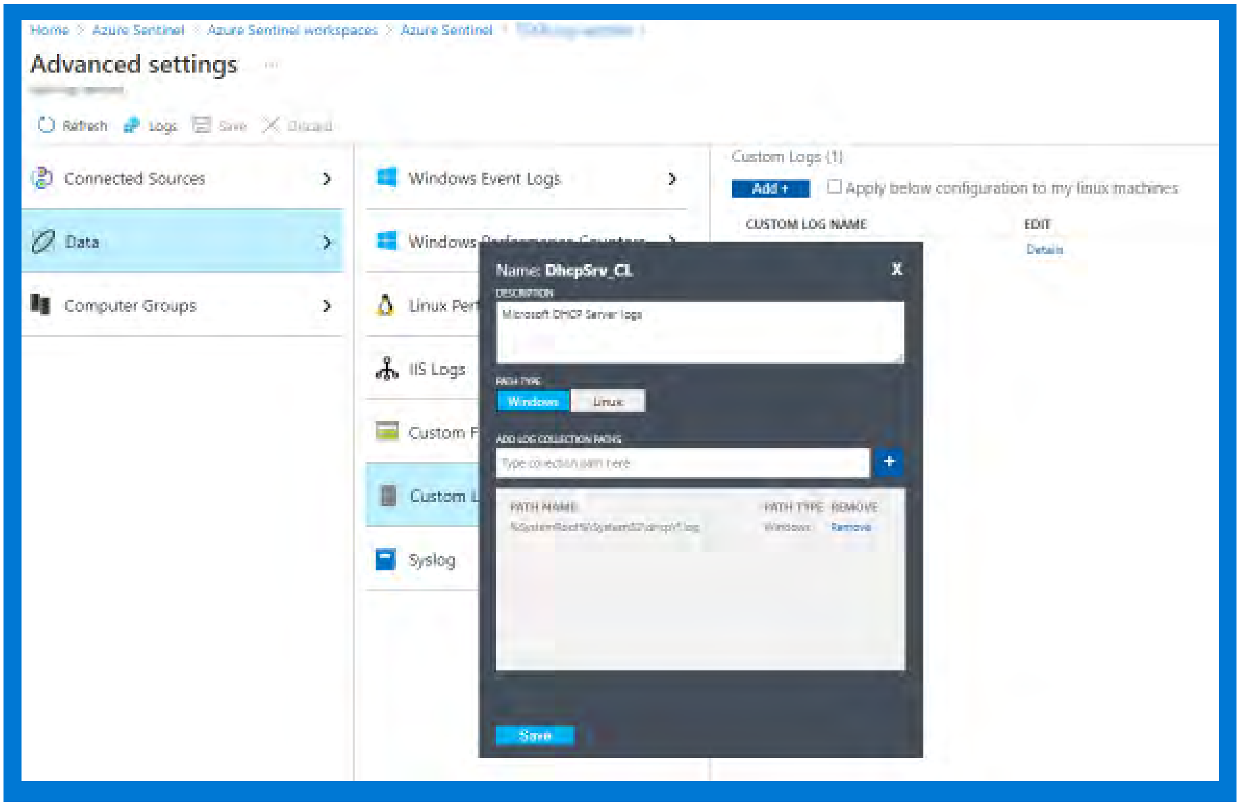
Fig. 4. Azure Active Directory built-in Azure Sentinel data connector

Deploying *MMAs*

The MMA is used across multiple Microsoft solutions and has undergone a few name changes as its features and functionality have evolved. You may also see it denoted as the OMS, Log Analytics Agent, Azure Sentinel Agent, or AMA. See the article "[Overview of Azure Monitor agents]" for a side-by-side comparison of different agent versions and some additional details about the differing nomenclatures. Deploying the MMA on Windows allows for the collection of Windows event logs, performance counters, IIS logs, and any other local text-based log.

MMA commonly collects Windows Security, application, and system event logs as well as web server logs from Microsoft IIS. The logging level for security event logs, up or down, are controlled from the Windows Security data connector.

The collection of custom logs is typically used when an application is storing logs on the local hard disk with a consistent log naming convention. One such example is Microsoft DHCP Server that stores the logs in C:\Windows\System32\dhcp\*.log:



Name: DhcpSrv_CL

DESCRIPTION: Microsoft DHCP Server logs

PATH NAME	PATH TYPE	REMOVE
%SystemRoot%\System32\dhcp\*.log	Windows	Remove

Fig. 5. Collecting Microsoft DHCP server logs as a custom log in Azure Monitor Agent

Deploying the MMA on Linux allows for the collection of any syslog message or local logs that follow a consistent naming convention. The log collection can be filtered by both syslog facility and syslog message severity. Any Linux agent with MMA installed can act as a syslog collector for remote syslog log sources.

Real-life considerations for the MMA during an Azure Sentinel deployment project:

- Understand the role and placement of the MMA, including which systems to deploy to.
- Consider pre-existing agents, such as OMS, and develop a migration/upgrade strategy.
- Consider the central deployment and management methodology for MMA. Your company may already have a distribution mechanism such as Microsoft Endpoint Manager.
- Determine which endpoints are in scope for deployment (both servers and workstations) as these affect your potential log sources and ingest volume.

Deploying a syslog collector

For remote syslog log collection, Azure Sentinel requires a syslog server with Linux rsyslog or syslog-ng syslog servers as very common choices.

The server can be deployed on-premises as a VM or physical server or as a VM in Azure or other cloud environments. The main requirement for a VM is to provide routing and connectivity from the log sources that need to send syslog data.

For isolated locations with limited WAN or VPN connectivity to the syslog collector location, depending on the log sources capabilities, a TLS-based syslog server can be deployed in an internet accessible location (such as an Azure VM) to receive encrypted logs via the internet. Most TLS-aware log sources support self-signed certificates, but if a Public Key Infrastructure (PKI) solution is available, its encryption/authentication certificates can be employed. Access to the TLS syslog can be restricted to selected sources as applicable.

Some SaaS solutions use TLS-enabled syslog as the sole logging option offered to their customers.

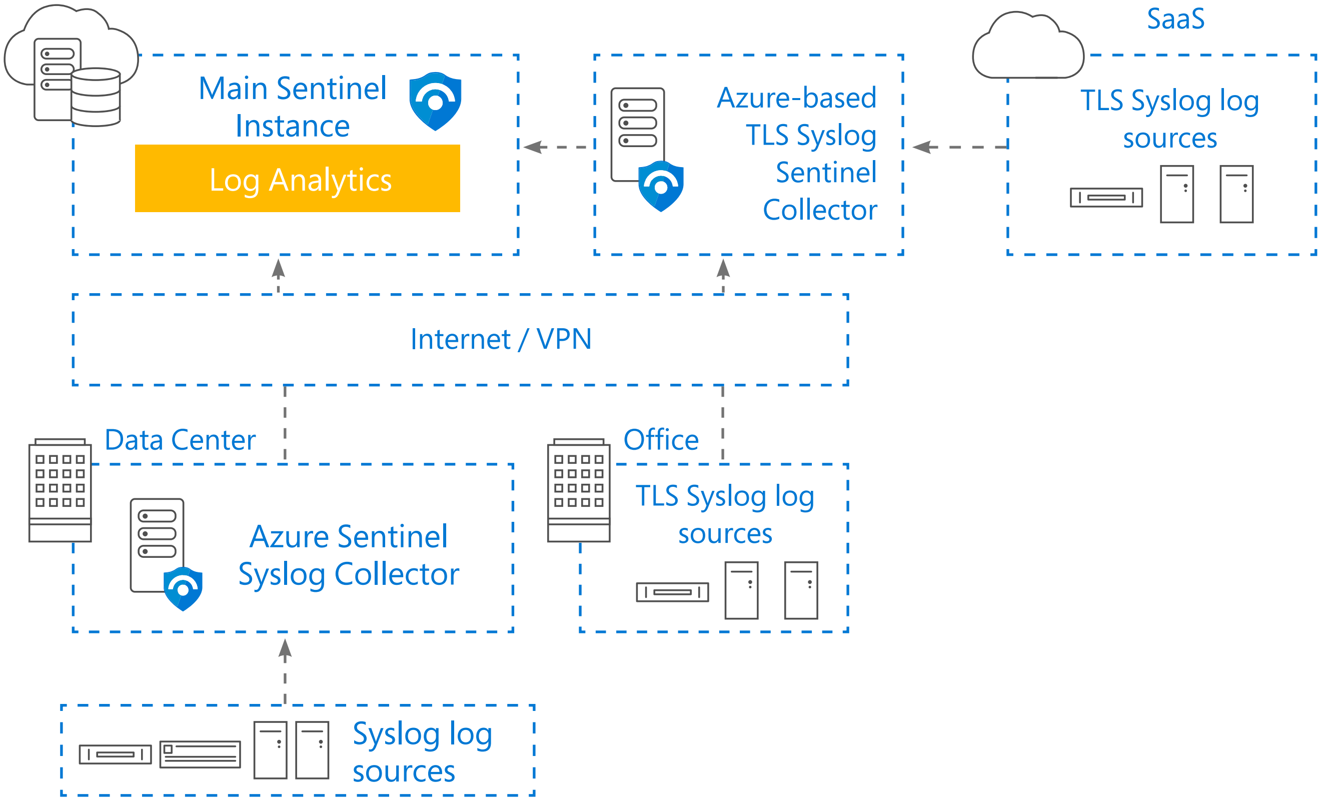


Fig. 6. Typical Azure Sentinel syslog collector deployment configurations

The hardware requirements for the syslog collector depend on the expected logging volume. The ingestion rate limits depend on several factors, such as the type of log source, size of a typical log entry, internet connectivity, protocol used (regular UDP syslog, TCP syslog, TLS syslog) and others. Based on our tests with a range of typical logs, such a mix of firewalls, and LAN devices, a single collector using 2 CPUs, 8 GB of RAM, and 50 GB of local storage can handle up to 6,000 EPS (peak value).

In most situations, the syslog collector will simply receive the logs and forward them to Azure Sentinel. If the intention is to also keep a copy of the raw logs on the syslog server as an offline queue, log backup, or for long-term retention, then required hard disk space must be provisioned.

The syslog collector is used to receive both plain syslog logs and CEF. If this is the case, pay special attention to make sure that the same facility is not used by both CEF and non-CEF log sources.

Some considerations, based on our experience deploying syslog collectors:

- Coordinate with the team to ensure that a technical resource is available to help configure the syslog collector – this may require personnel that have not previously been engaged in the Sentinel project.
- Be sure to provision access to the syslog collector, especially if Azure Sentinel is deployed by an external consultant.
- Try to avoid comingling existing syslog collectors to minimize potential ingest or parsing issues.
- Architect to your risk tolerance in terms of syslog collector availability. Determine potential impacts to log collection if a network outage occurs.
- Deploy your syslog collector on a locally common Linux distribution. If there is none, this may be a good instance to help introduce a standard.

Some syslog collectors may provide limited configuration options , leading to challenges around the log ingestion configuration.

In our experience, log source types that do not adhere to syslog standards or that lack logging configuration options may require additional steps to properly ingest and parse data into Azure Sentinel. For example, Cisco Meraki and Firepower devices logging formats can be challenging, depending on version. Consider one of the more configurable commercial syslog engines such as Syslog-NG pro in these scenarios.

Azure Sentinel playbooks

Based on Azure Logic Apps connectors, an Azure Sentinel playbook can be used to retrieve logs from various sources using Logic App connectors. Providing that the log source offers a log query/collection capability via API, an HTTP request using a REST API connection to the log source interface is a good example of this type of scenario. Azure Sentinel playbooks should be used just for low volume/log complexity log sources as they are not designed to perform large data transfers. The playbooks can retrieve authentication secrets from Azure Key Vault and authenticate using a managed security identity.

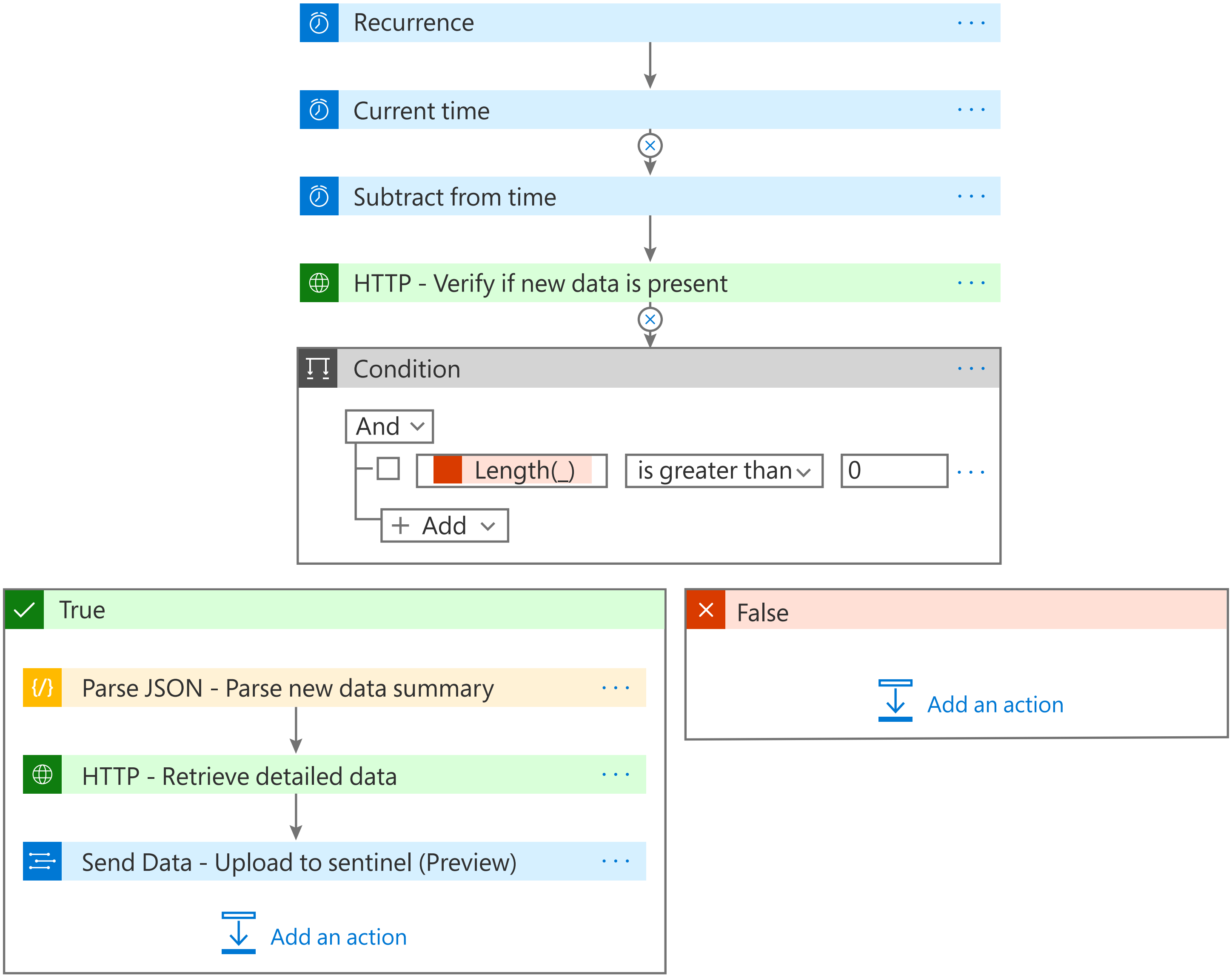

Recurrence
Current time
Subtract from time
HTTP - Verify if new data is present
Condition
And
Length(_)
is greater than
0
+ Add
True
Parse JSON - Parse new data summary
HTTP - Retrieve detailed data
Send Data - Upload to sentinel (Preview)
Add an action
False
Add an action

Fig. 7. Example of log ingestion via Azure Sentinel playbook–retrieval of Office 365 workloads not ingested via Office 365 Data Connector (e.g., DLP, Power BI)

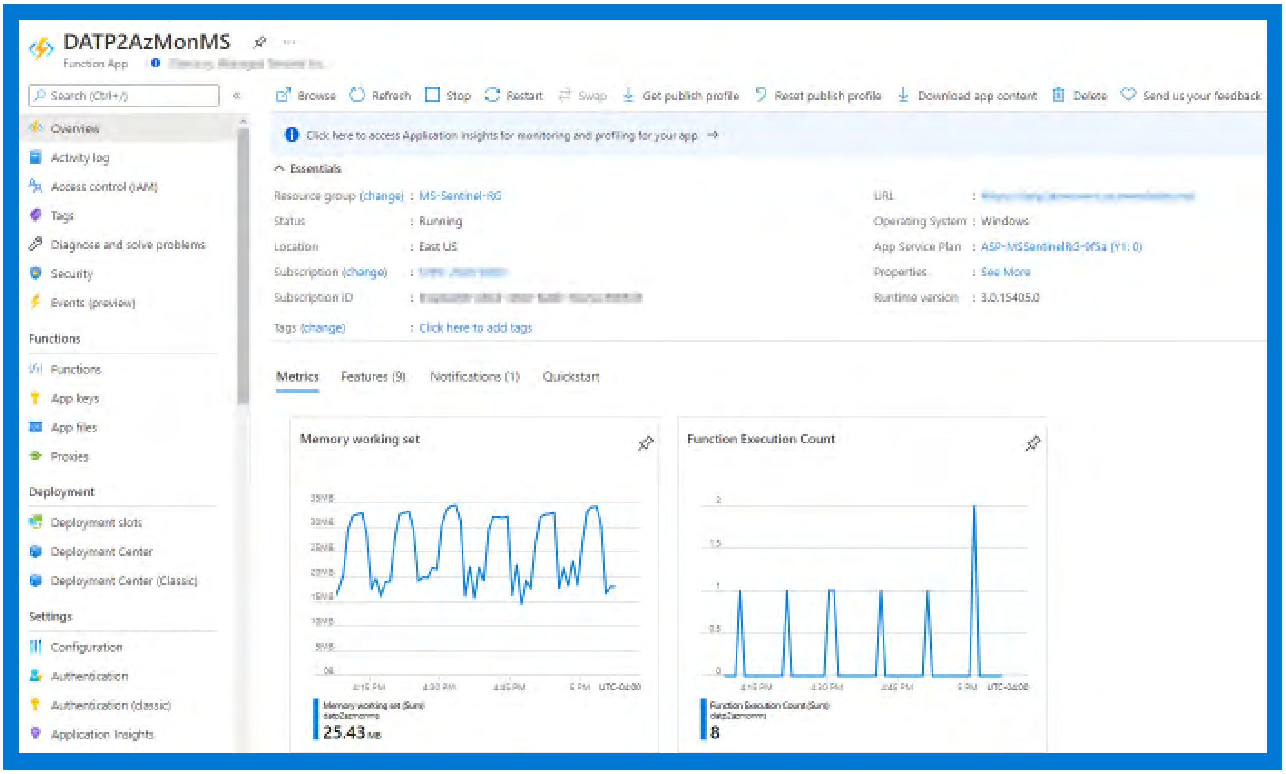
In our experience, the following are some considerations encountered by teams deploying playbook automations in Logic Apps:

- Plan for the acceptance requirements of Azure Sentinel playbooks automation. Automated actions can be high-impact, and understanding risk tolerances is important.
- Monitor playbook runs and plan to alert if issues are encountered so the problem can be quickly remediated versus discovered too late.
- Consider any third-party tools involved and how they may work with Azure Sentinel playbooks. You may need to consult the third party for details about its API or other connectivity options.
- Create and maintain API connections across various Log App connectors. For example, consider the use of SPNs versus individual service accounts.

Azure Function Apps

As a low footprint, relatively inexpensive resource, Azure Function Apps are one of the most stable and performant log ingestion methods. Functions apps provide the full capabilities of .Net, Python, PowerShell, and recently, Node.js and can be used to perform a wide range of log ingestion tasks, including but not limited to log retrieval via REST APIs, pagination, filtering or parsing, and enrichment of data. Azure Function Apps require more advanced programming capabilities.

Many built-in Azure Sentinel connectors rely on the deployment of function apps, and typically they are developed by the log source vendor working in collaboration with Microsoft.



Resource group (change) : MS-Sentinel-RG
Status : Running
Location : East US
Subscription (change) : [redacted]
Subscription ID : [redacted]
Tags (change) : Click here to add tags
URL : [redacted]
Operating System : Windows
App Service Plan : ASP-MSSentinelRG-9f5a (Y1: 0)
Properties : See More
Runtime version : 3.0.15405.0

Memory working set (Sum)
datp2azmonms
25.43 MB

Function Execution Count (Sum)
datp2azmonms
8

Fig. 8. Azure Function App retrieving custom data from Microsoft Defender for endpoints

In developing Azure Function Apps, we suggest the following considerations be addressed when deploying:

- Leverage members of the team who have programming expertise, as deploying Azure Function Apps requires programming.
- Gain a good understanding of log source interface capabilities (i.e., REST API) before deploying.
- Deploy only one compute platform per resource group (i.e., Python versus .Net).

Case study–Engineering company

CMEng Inc. is an engineering company located in Canada with customers around the world. Currently CMEng is running a hybrid infrastructure with workloads in Azure, AWS, and Oracle Cloud Infrastructure (OCI) as well on-premises. Also, CMEng is using several SaaS applications for various business lines.

CMEng has decided to migrate to Azure Sentinel as its main security monitoring tool. Onboarding all log sources in Azure Sentinel would require the development of several custom data connectors. Some of the log sources selected for custom data connector development were OCI, Duo multi-factor authentication (MFA), and Cloudflare WAF. None of these data connectors existed in the out-of-the-box Azure Sentinel data connectors or in easily accessible community forums.

The data connectors were developed based on vendor REST APIs using Azure Function App, and application data was ingested in Log Analytics custom tables (_CL). Effort required for this project was around 40 hours, completed by a senior cloud security developer. Access to team members with programming expertise can broaden the options for data ingest significantly.

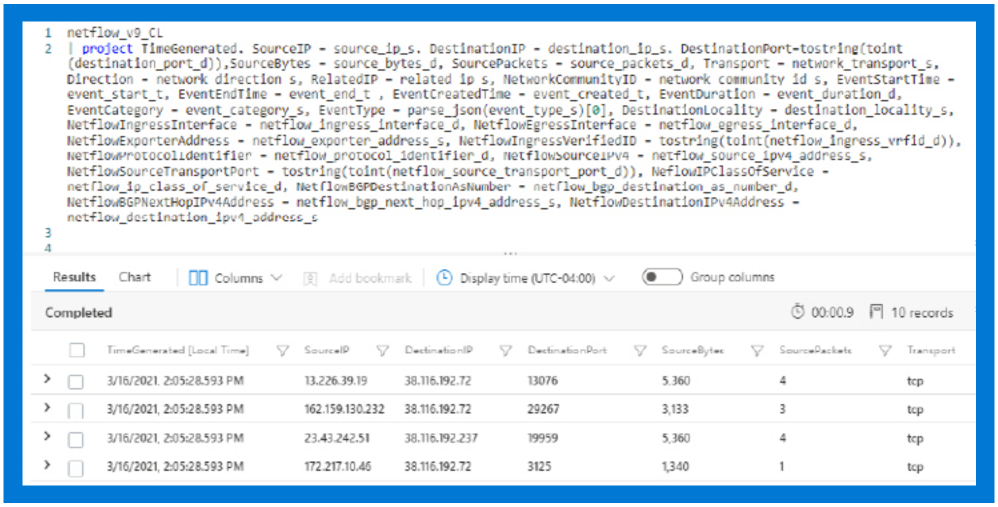
Vendor-provided log retrieval tools

Depending on their specific design and maturity level, some SaaS platforms may offer specialized utilities that can be run at scheduled intervals to retrieve activity logs. These could be scripts (such as PowerShell and Python) or executables. Once these scripts are configured according to the vendor's instructions, the MMA can be used to monitor the logs' location and update new data as it is downloaded by the log retrieval script.

For more advanced processing/filtering capabilities, tools such as Logstash can be used to collect the local logs, process them, and upload them to Azure Sentinel using the Microsoft Log Analytics Logstash plugin.

An important aspect of logging data ingested via Logstash or the custom logs AMA capability is that the logs end up in Azure Sentinel as a custom log (with the extension _CL). Custom logs may require additional parsers and data normalizations that can be implemented through KQL.

Fig. 9. Example of Netflow version 9 custom log and associated parser



```
netflow_v9_CL
| project TimeGenerated. SourceIP = source_ip_s. DestinationIP = destination_ip_s. DestinationPort=tostring(toint
(destination_port_d)),SourceBytes = source_bytes_d, SourcePackets = source_packets_d, Transport = network_transport_s,
Direction = network direction s, RelatedIP = related ip s, NetworkCommunityID = network community id s, EventStartTime =
event_start_t, EventEndTime = event_end_t , EventCreatedTime = event_created_t, EventDuration = event_duration_d,
EventCategory = event_category_s, EventType = parse_json(event_type_s)[0], DestinationLocality = destination_locality_s,
NetflowIngressInterface = netflow_ingress_interface_d, NetflowEgressInterface = netflow_egress_interface_d,
NetflowExporterAddress = netflow_exporter_address_s, NetflowIngressVerifiedID = tostring(toint(netflow_ingress_vrfid_d)),
NetflowProtocolIdentifier = netflow_protocol_identifier_d, NetflowSourceIPv4 = netflow_source_ipv4_address_s,
NetflowSourceTransportPort = tostring(toint(netflow_source_transport_port_d)), NeflowIPClassOfService =
netflow_ip_class_of_service_d, NetflowBGPDestinationAsNumber = netflow_bgp_destination_as_number_d,
NetflowBGPNextHopIPv4Address = netflow_bgp_next_hop_ipv4_address_s, NetflowDestinationIPv4Address =
netflow_destination_ipv4_address_s


```

Completed 00:00.9 10 records

TimeGenerated [Local Time]	SourceIP	DestinationIP	DestinationPort	SourceBytes	SourcePackets	Transport
3/16/2021, 2:05:28.593 PM	13.226.39.19	38.116.192.72	13076	5,360	4	tcp
3/16/2021, 2:05:28.593 PM	162.159.130.232	38.116.192.72	29267	3,133	3	tcp
3/16/2021, 2:05:28.593 PM	23.43.242.51	38.116.192.237	19959	5,360	4	tcp
3/16/2021, 2:05:28.593 PM	172.217.10.46	38.116.192.72	3125	1,340	1	tcp

For certain solutions, log collection tools are provided by third-party developers not associated with the vendor itself. Most of these tools are available in free-to-use GitHub repositories.

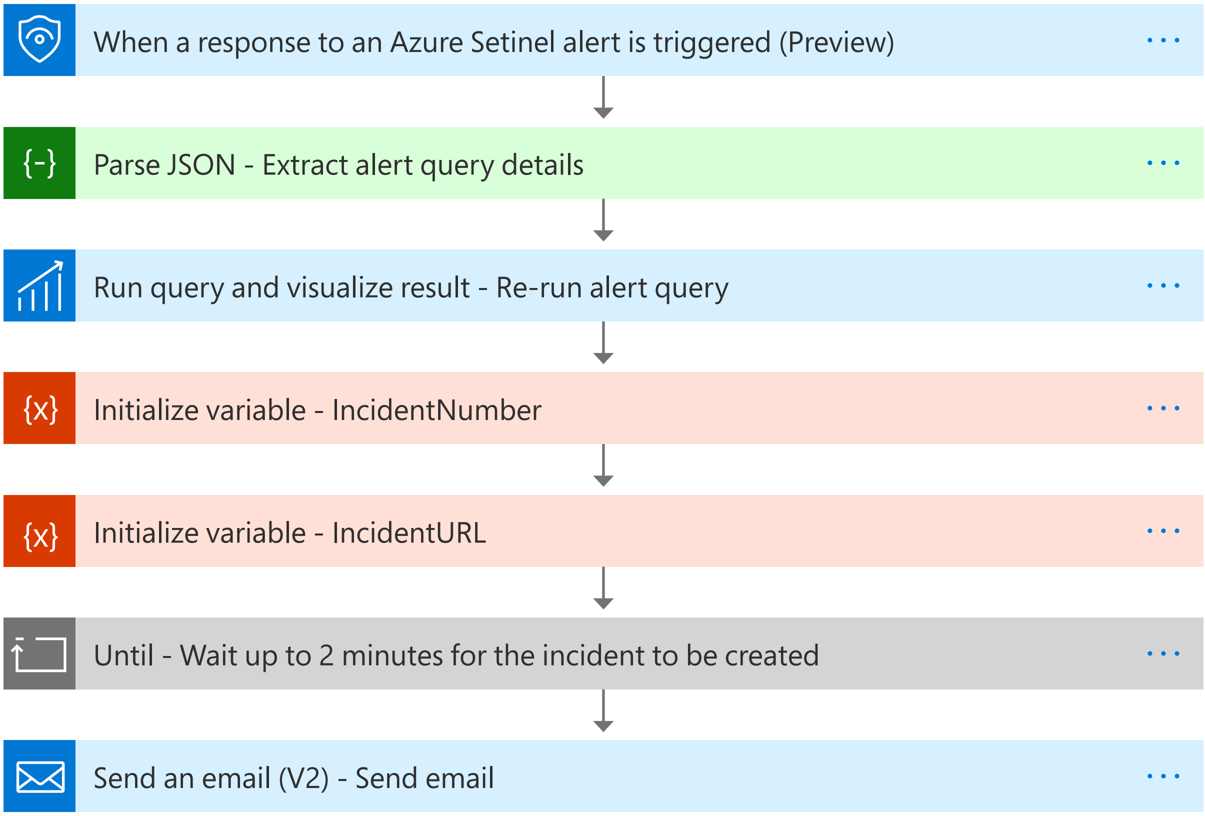
Additional resources:

- Connect data sources to Azure Sentinel | Microsoft Docs
- Resources for creating Azure Sentinel custom connectors | Microsoft Docs
- Connect data sources through Logstash to Azure Sentinel | Microsoft Docs
- Overview of the Azure monitoring agents - Azure Monitor | Microsoft Docs
- Configure data collection for the Azure Monitor Agent (preview) - Azure Monitor | Microsoft Docs

Automation playbooks

The main engine behind the Azure Sentinel automation capability is Azure Logic Apps. Logic Apps were released as General Availability in 2016—while part of Azure Sentinel SOAR capabilities, they are a proven technology used across the Azure ecosystem.

A playbook is initiated by a trigger and performs its functionality through a variety of connectors using Microsoft Flow. Flow allows for transfer of data between connectors and a wide range of processing functions. In Azure Sentinel, the typical trigger is “When a response to an Azure Sentinel alert is triggered,” and a Logic App connector allows the collection of data by an Azure Sentinel alert that can be passed to various connectors to perform tasks based on the desired outcome of the security orchestration/automation scenario.



When a response to an Azure Setinel alert is triggered (Preview) ...

Parse JSON - Extract alert query details ...

Run query and visualize result - Re-run alert query ...

Initialize variable - IncidentNumber ...

Initialize variable - IncidentURL ...

Until - Wait up to 2 minutes for the incident to be created ...

Send an email (V2) - Send email ...

Fig. 10. Sample Azure Sentinel playbook with an Azure Sentinel trigger and connectors for sending alert details by email

Azure Sentinel playbooks can perform very advanced tasks with activities branching based on criteria identified in alerts or data retrieved using indicators collected in alerts. The full processing can be performed within the Azure Sentinel environment and is limited only by the automation capabilities provided by the third-party security controls required to provide information or perform additional tasks.

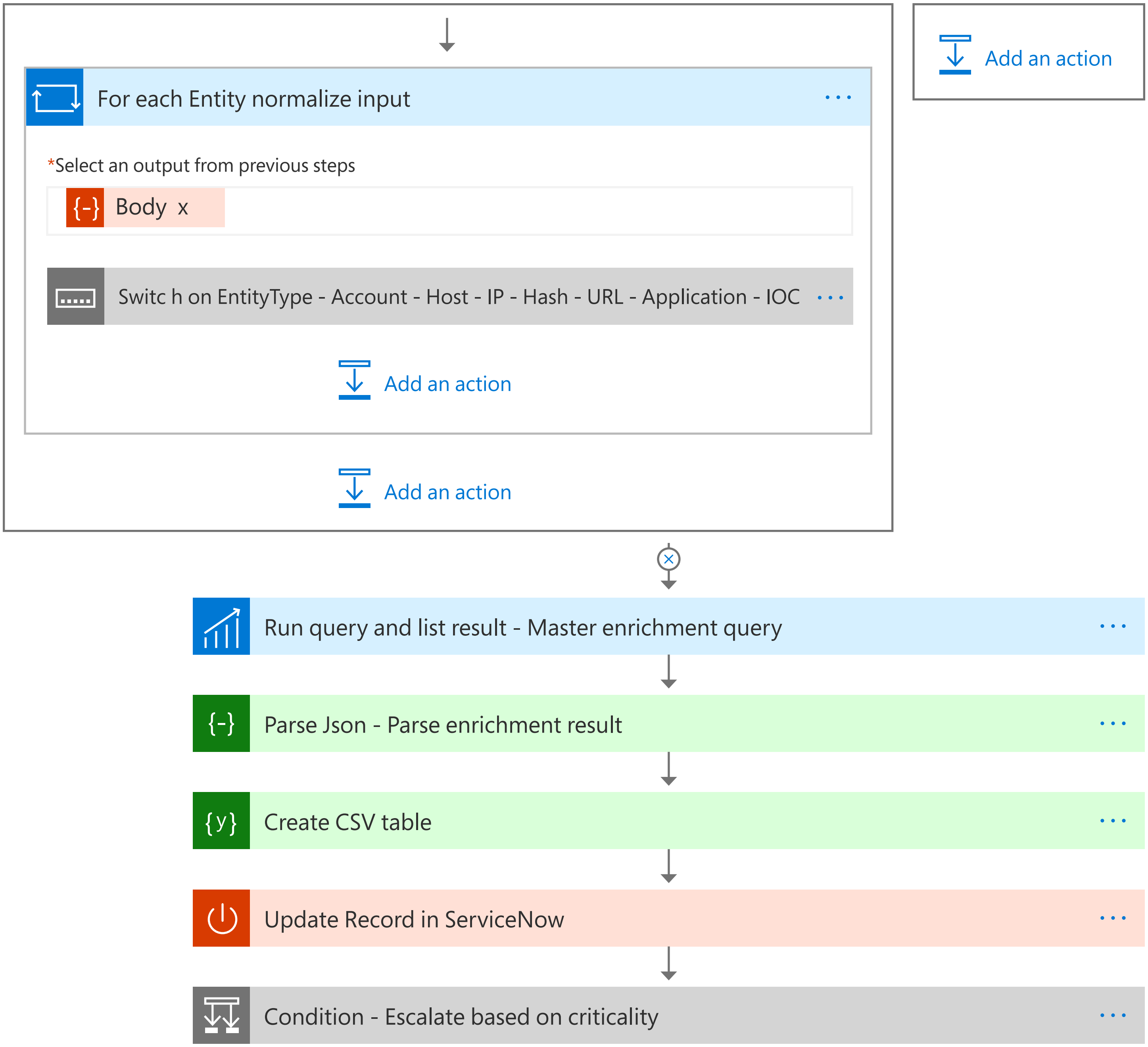

For each Entity normalize input
*Select an output from previous steps
Body x
Switc h on EntityType - Account - Host - IP - Hash - URL - Application - IOC
Add an action
Add an action
Add an action
Run query and list result - Master enrichment query
Parse Json - Parse enrichment result
Create CSV table
Update Record in ServiceNow
Condition - Escalate based on criticality

Fig. 11. Alert data enrichment playbook with integration with ServiceNow

The most used Logic App connectors in Azure Sentinel are:

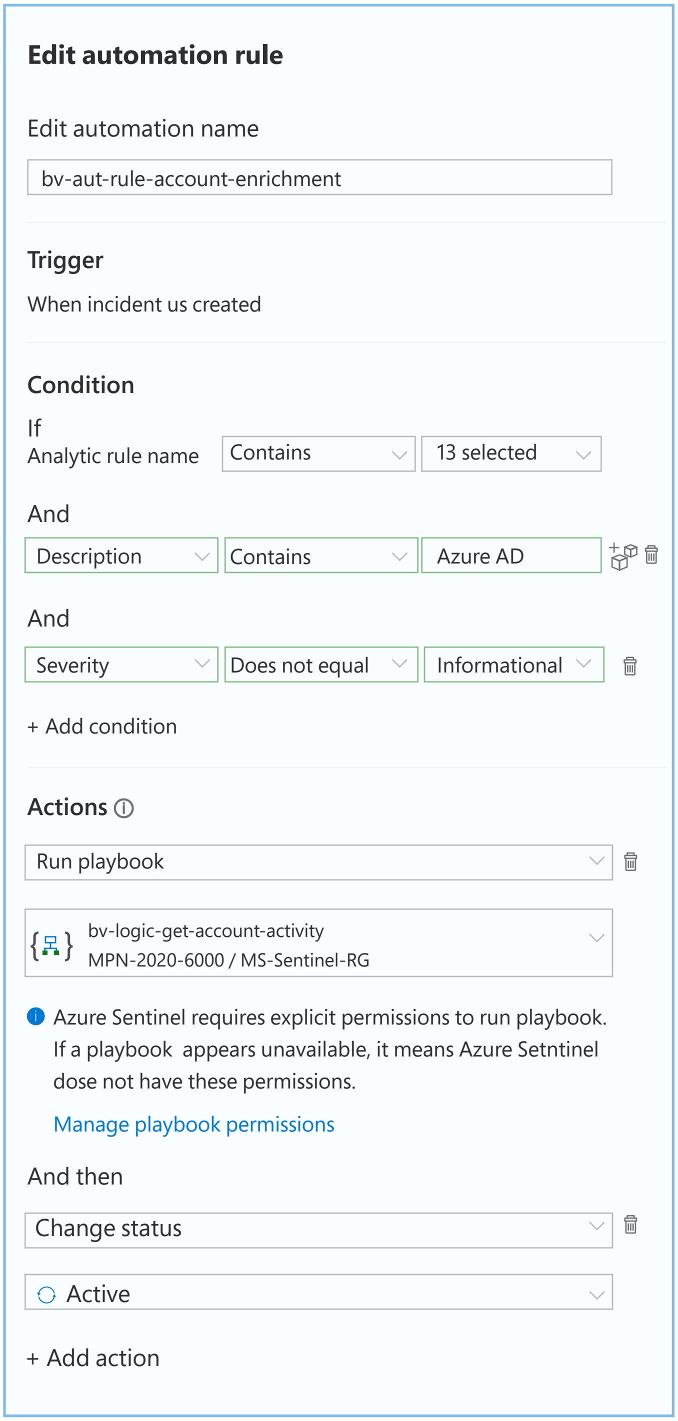
- Azure Sentinel. Occurrence triggers
- HTTP request. To retrieve data from third-party log sources via REST API
- Azure Monitor. Run KQL queries to retrieve additional data
- Data Operations. Parse JSON, Compose, Convert to CSV
- Controls. Conditions (if … else), For each, Switch
- Variables. Placeholders for data used during alert processing
- Send Data. To upload data into an Azure Sentinel table for further use
- Azure Sentinel. Retrieve/Update incidents
- Office 365. Send emails
- Notification controls. PagerDuty, Signl4, Twilio
- ITS tools connectors. ServiceNow, Freshservice, Freshdesk

Automation rules

The automation rules allow for a more intuitive construction of SOAR activities, providing the ability to build combinations of playbook runs and incident updates (severity, ownership, status, tagging) to match the required output.

The automation rule can be applied using a combination of filtering criteria, such as alert names, description, severities, or type of entities identified in the incident created by the alert rule. Multiple playbooks can be applied in succession, allowing for practically unlimited possibilities on how to automate the response for different types of incidents.

In the next example, for 13 alert rules selected from the dropdown list, those that contain "Azure AD" in their description and have a severity other than Informational, an enrichment playbook will be executed, and the incident status changed to Active.



Edit automation rule

Edit automation name

bv-aut-rule-account-enrichment

Trigger

When incident us created

Condition

If
Analytic rule name | Contains | 13 selected

And
Description | Contains | Azure AD

And
Severity | Does not equal | Informational

+ Add condition

Actions ⓘ

Run playbook

bv-logic-get-account-activity
MPN-2020-6000 / MS-Sentinel-RG

Azure Sentinel requires explicit permissions to run playbook. If a playbook appears unavailable, it means Azure Setntinel dose not have these permissions.

Manage playbook permissions

And then

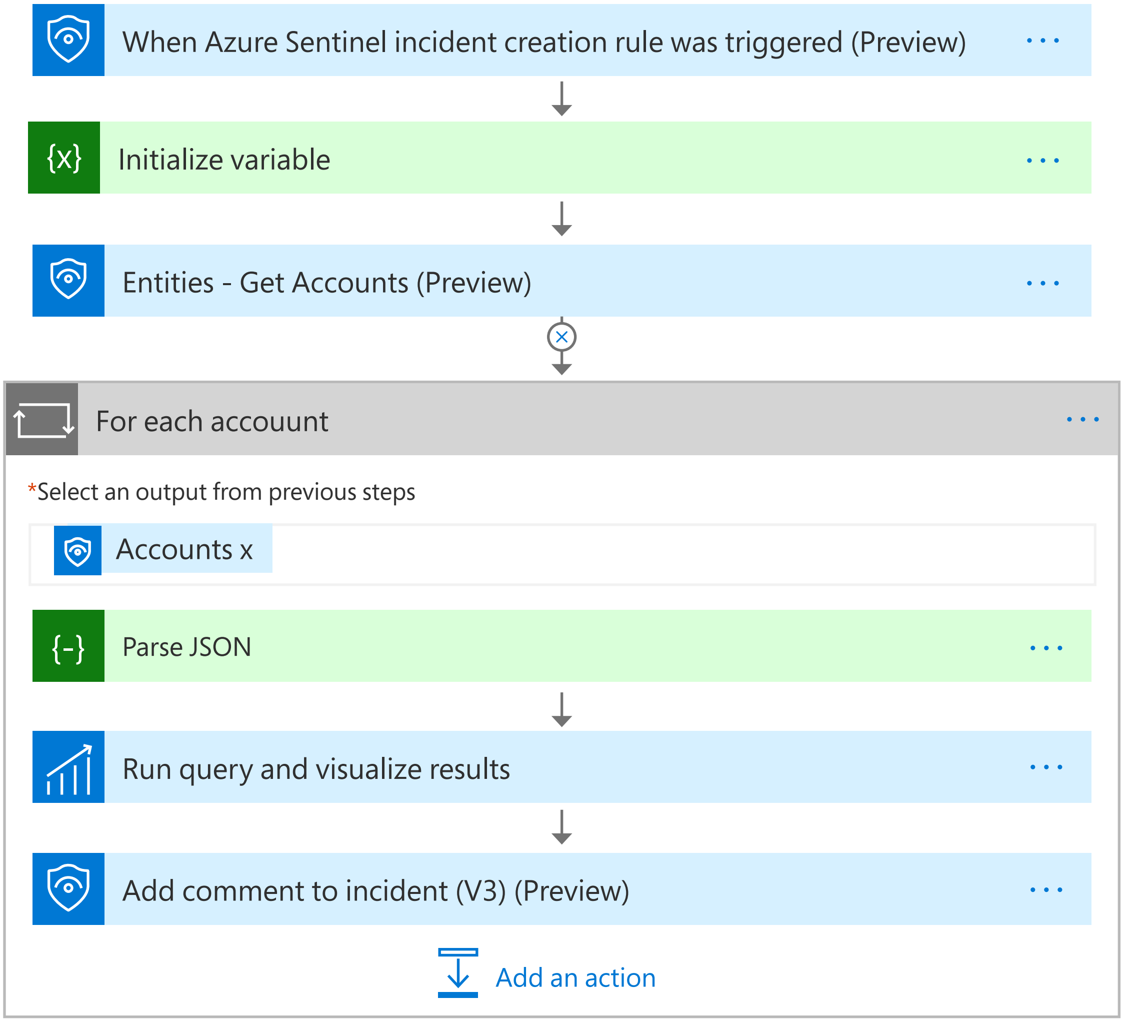
Change status

Active

+ Add action

Fig. 12. Automation rule

The playbook will retrieve the list of account entities from the alert, run a summary of their activities in Azure AD SigninLogs, and post the results as a comment in the incident, allowing the security analyst to have a snapshot of the suspicious account activities without leaving the incident page.



When Azure Sentinel incident creation rule was triggered (Preview)

Initialize variable

Entities - Get Accounts (Preview)

For each acoount

*Select an output from previous steps

Accounts x

Parse JSON

Run query and visualize results

Add comment to incident (V3) (Preview)

Add an action

Fig. 13. Enrichment playbook

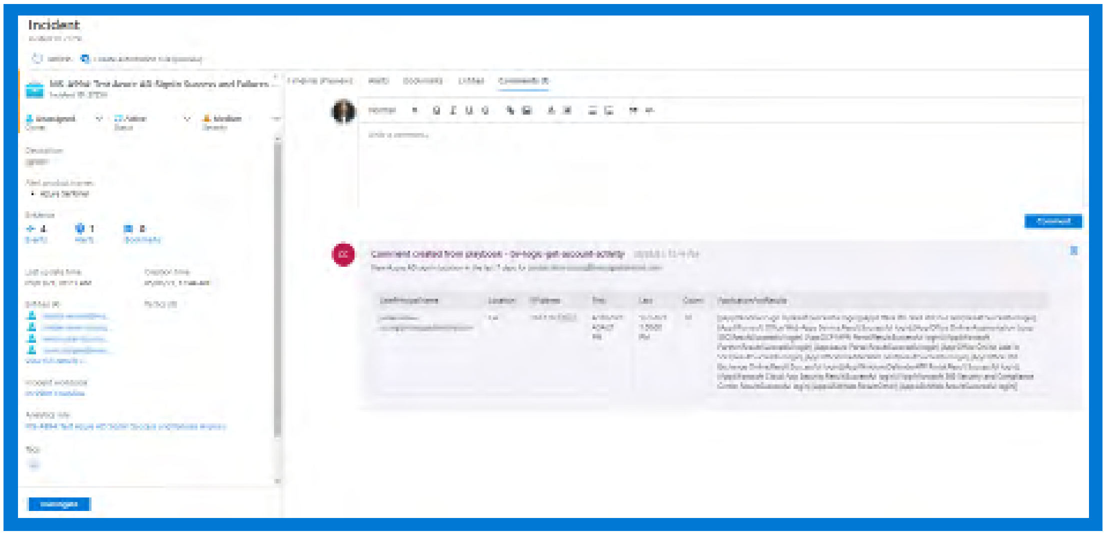


Fig. 14. Enriched alert with summary of Azure AD Signin logs

Additional resources:

- Playbooks & Watchlists Part 1: Inform the subscription owner - Microsoft Tech Community
- Playbooks & Watchlists Part 2: Automate incident response for Deny-list/Allow-list - Microsoft Tech Community

Deploying workbooks

The Azure Sentinel workbooks provide a wide range of data visualization based on KQL queries and integration with additional Microsoft resources (via REST APIs). Over 100 workbook templates are provided for the typical log sources such as Azure Active Directory, Office 365, Windows Active Directory, and third-party log sources (e.g., firewalls, SaaS).

The workbooks provide several visualization controls (e.g., bar, pie, area, time charts), conditional formatting, and several other features commonly found in analytical platforms.

Workbooks can retrieve data from multiple sources, allowing for complex integration with various Microsoft services: Azure Log Analytics Workspace, Microsoft Graph, Azure Data Explorer, Azure Resource Manager, and many other sources. The existing templates that ship with Azure Sentinel can be reused for new workbooks tailored for customer-specific requirements.

In our experience:

- The top required custom workbook is for key performance indicators (KPIs) that would allow executives to make decisions around cybersecurity governance. Defining such KPIs and extracting them from the logs can be challenging due to the fact that most KPIs rely on measuring the efficiency of a processes-tools-people combination rather than log data provided by security controls. Through regular review and feedback from the consumers of reports, workbooks can become very effective tools.
- Low fidelity alerts can be captured in workbooks for regular review and avoid the cluttering of the incident management interface.
- One of the most used workbooks is Security Operations Efficiency.

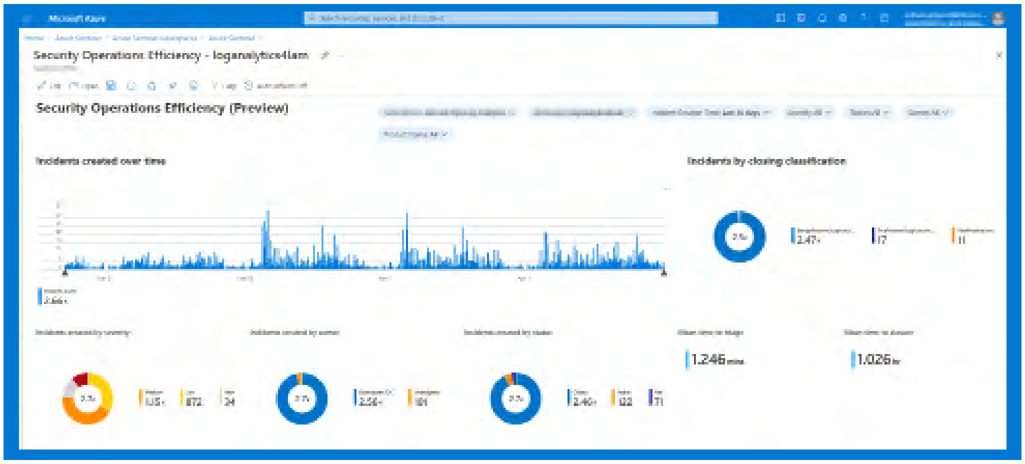

Microsoft Azure
Security Operations Efficiency - loganalytics4lam
Security Operations Efficiency (Preview)
Incidents created over time
Incidents by closing classification
2.47
17
11
2.66
1.246
1.026
1.15
872
24
2.56
101
2.46
122
71

Fig. 15. Security Operations Efficiency Workbook

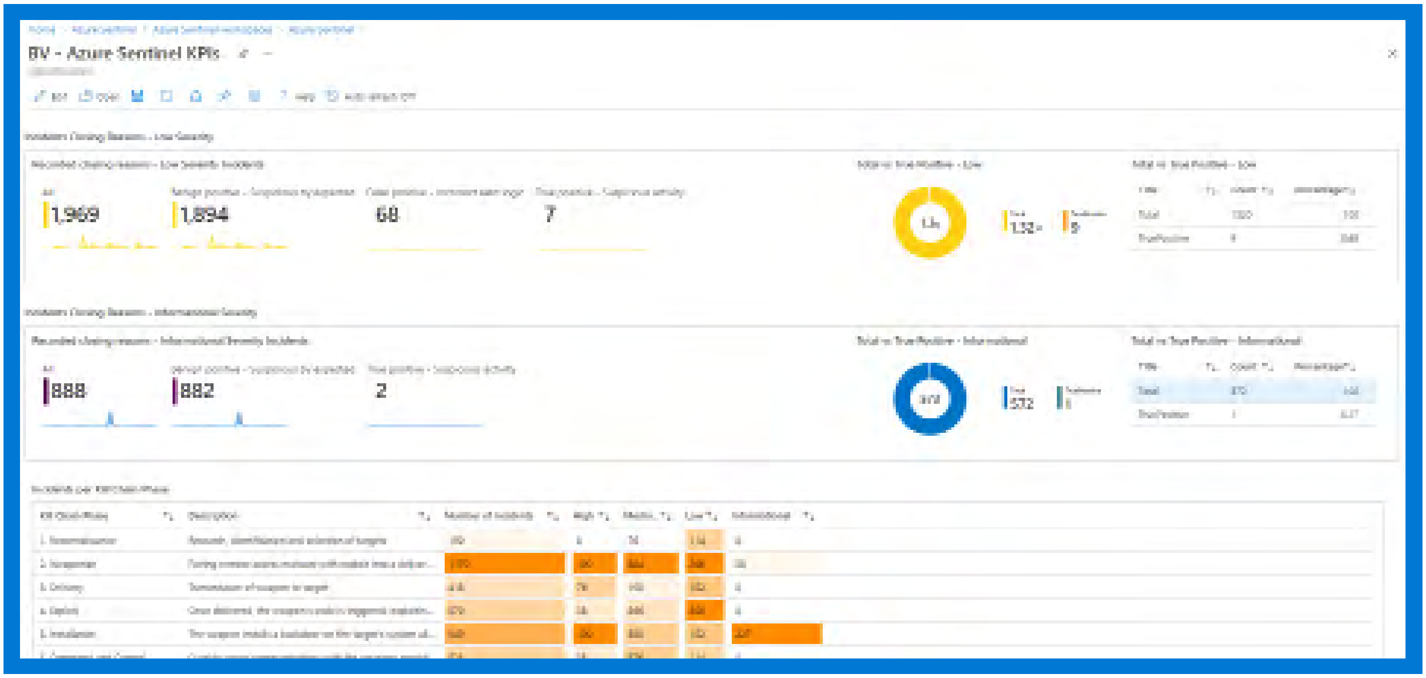


Fig. 16. Example of a custom KPIs workbook

Additional resources:

- [Azure Sentinel Workbooks 101 (with sample Workbook) - Microsoft Tech Community]()
- [Commonly used Azure Sentinel workbooks | Microsoft Docs]()
- [How to use Azure Monitor Workbooks to map Sentinel data - Microsoft Tech Community]()

Deploying user and entity behavior analytics

User and Entity Behavior Analytics (UEBA) represents one of the relatively new technologies provided by SIEM solutions and rely in most cases on machine learning capabilities to track the behavior of users and entities such as hosts and IP addresses and detect deviations from the expected patterns.

Azure Sentinel relies on a range of logs sources such as Azure Active Directory, Windows Active Directory, Office 365, and others to build models around the activity of users and computers and build “insights” around their observed behavior. Azure Sentinel builds several baselines using 10, 30, 90, or 180 days of observed behavior, based on the type of log source. Through UEBA, multiple low fidelity signals can be aggregated to build a timeline of events that can be used to better understand the entity behavior and take more informed decision during further investigations.

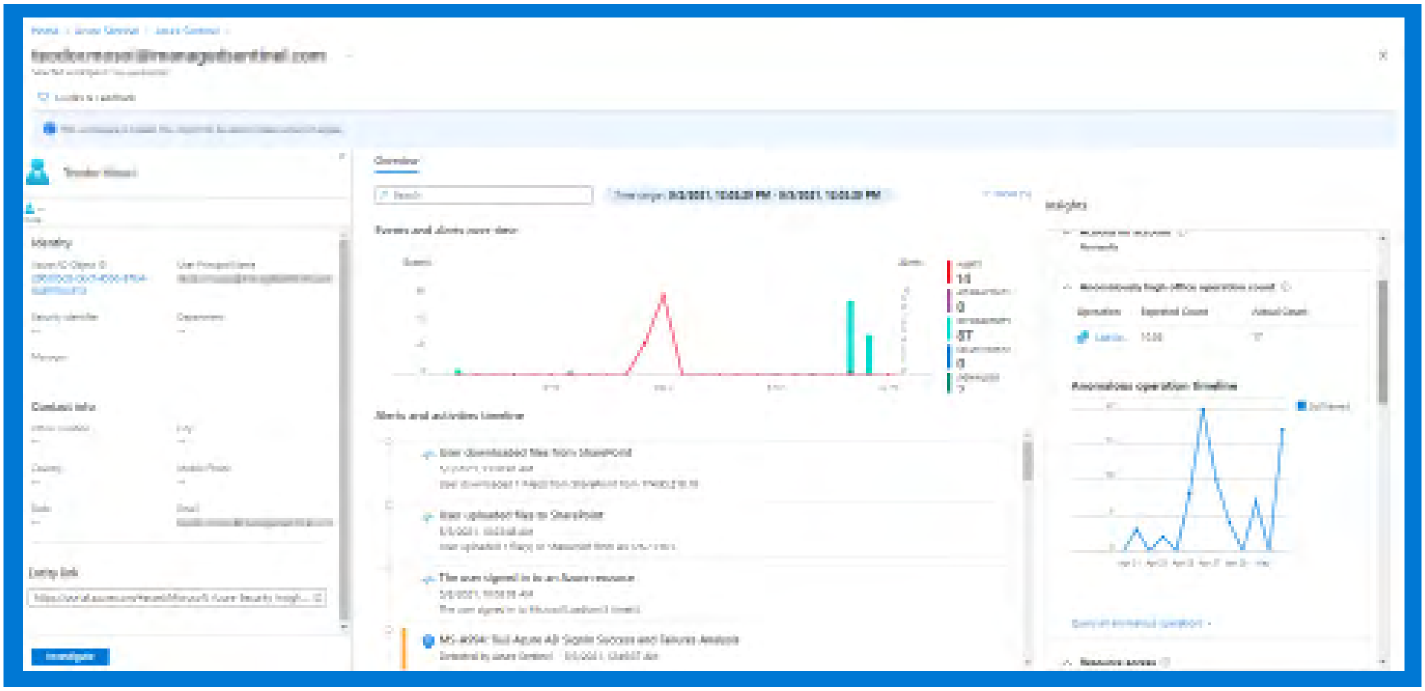

Overview
Events and alerts over time
Alerts and activities timeline
User downloaded files from SharePoint
User uploaded files to SharePoint
The user signed in to an Azure resource
MS-A094: Test Azure AD Signin Success and Failures Analysis
Insights
Anomalous operation timeline
Identity
Contact info
Entity link
Investigate

Fig. 17. User entity Azure Sentinel entity behavior interface

The entity information interface allows for quick pivoting into an Azure Sentinel investigation graph for further details on the relation between the entity and the alerts raised by Azure Sentinel.

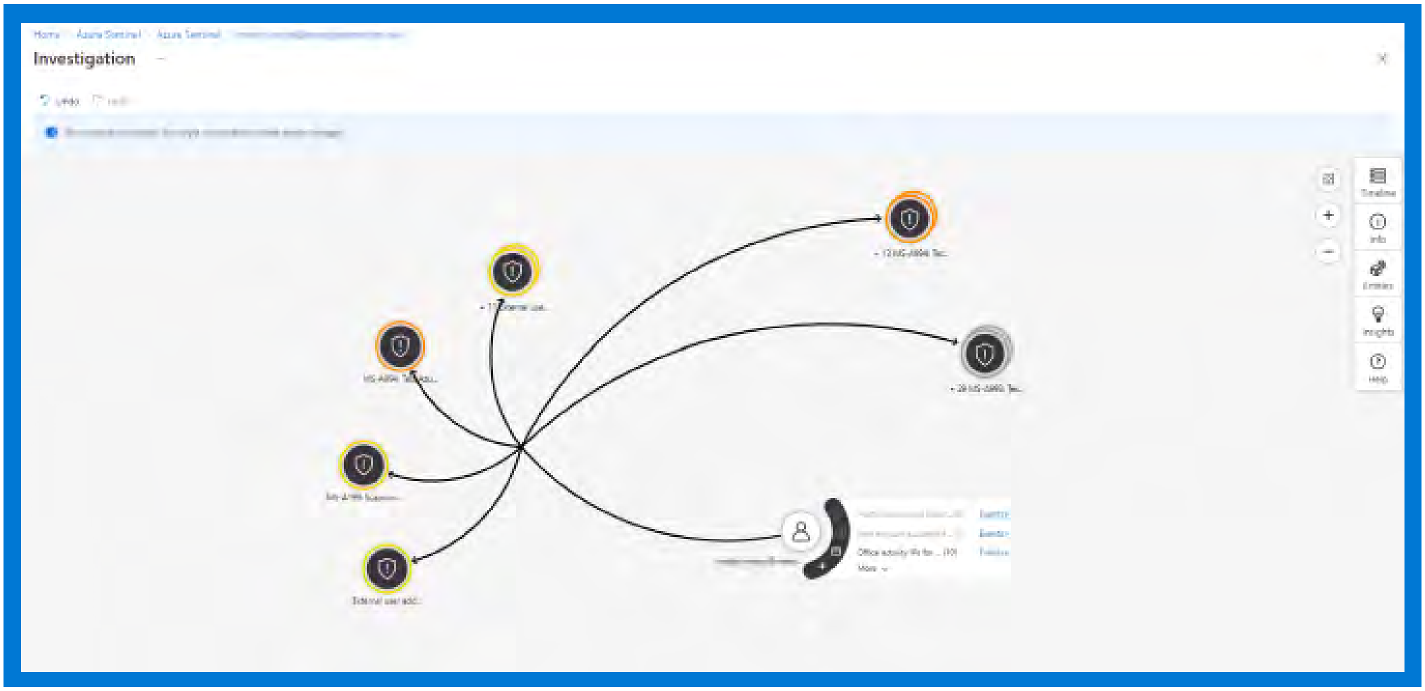


Fig. 18. Azure Sentinel investigation graph

The UEBA data is available as Azure Sentinel tables for full integration with alert rules and SOAR playbooks, allowing for the enrichment of alert metadata and adjustment of incident severity based on the behavior retrieved from UEBA.

In our experience:

- Substantial data is compiled by UEBA engines, which are often underutilized by security organizations in Azure Sentinel deployments. We have found these detections to be highly valuable and recommend utilizing them in security operations use cases.

Additional resources:

- Identify advanced threats with User and Entity Behavior Analytics (UEBA) in Azure Sentinel | Microsoft Docs
- Azure Sentinel UEBA enrichments reference | Microsoft Docs

Deploying notebooks

Based on Jupyter Notebooks, Azure Sentinel Notebooks allow for advanced threat hunting capabilities, using the data collected by Azure Sentinel and the processing capabilities available in multiple programming languages including but not limited to Python and C#/. Net. Any library, such as those related to threat intelligence enrichment and ML/AI available to the selected programming language, is available to use toward the Azure Sentinel log data.

In our experience:

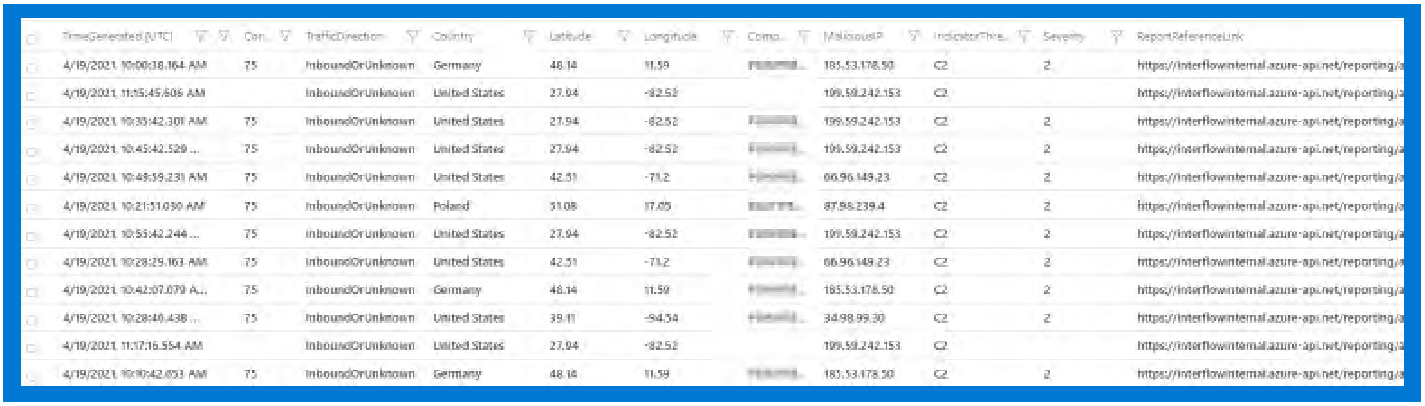
- Notebooks require knowledge of at least one programming language SME and, due to their technical complexity, can have a learning curve, but they are very powerful.
- Msticpy Python library, developed and maintained by Microsoft, is extensively used to perform a wide number of threat intel enrichment tasks within Notebooks, and there are many articles with examples of use for threat hunting scenarios, with some linked here.

Additional resources:

- Use notebooks with Azure Sentinel for security hunting | Microsoft Docs
- msticpy - Python Defender Tools - Microsoft Tech Community

Deploying cyber threat intelligence functionality

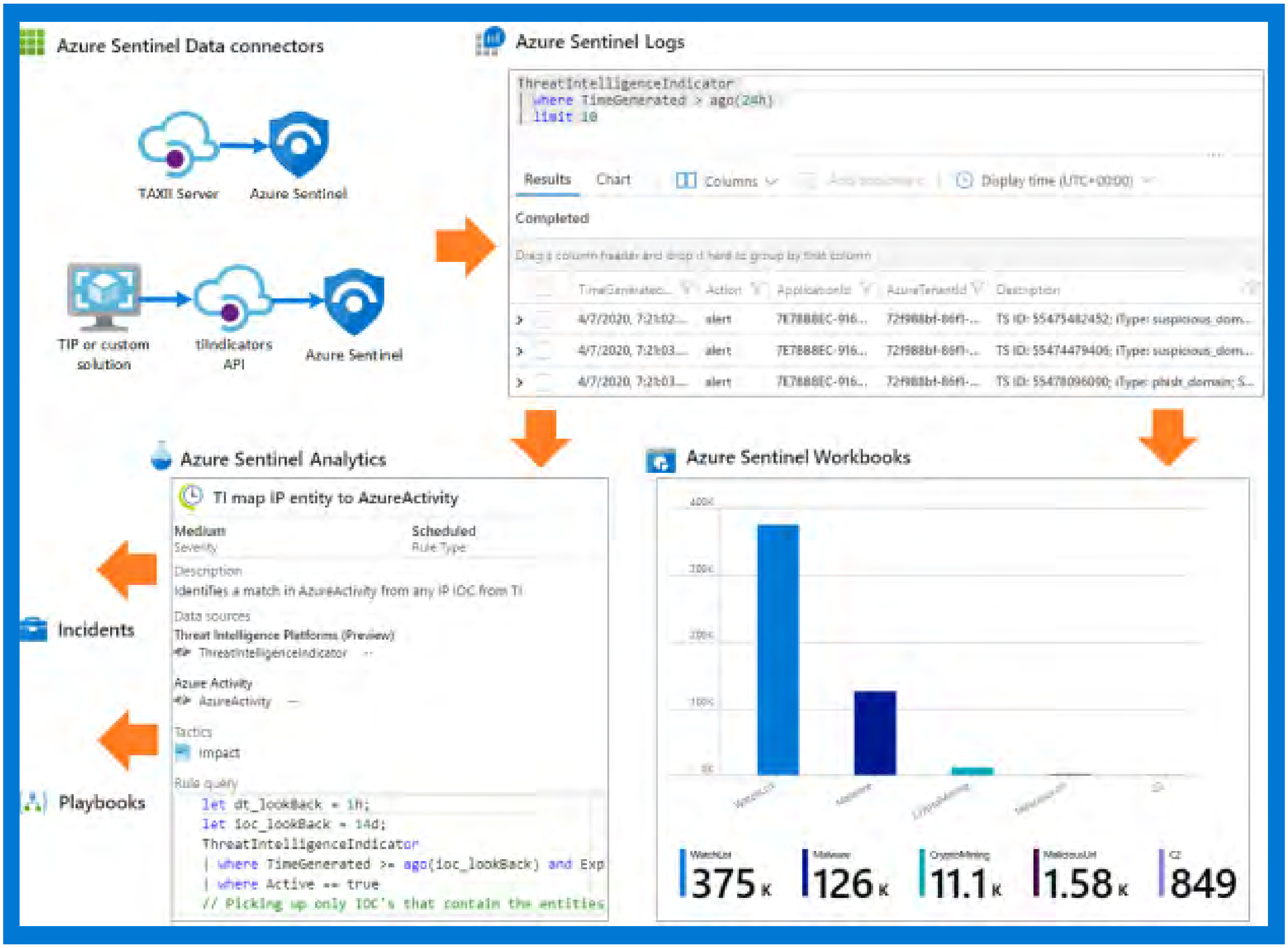
Cyber threat intelligence (CTI) is available from a wide array of sources. These can include open-source data feeds, threat intelligence sharing communities, premium curated feeds, and your own security investigations. CTI can be provided as a formal write up about a given threat actor’s tactics, techniques, and procedures (TTP), underlying goals or motivations, or specific lists of observed domain names, IP addresses, email addresses, and file hashes—the latter are collectively known as indicators of compromise (IOCs) but also known within Azure Sentinel as threat indicator data. CTI can provide valuable contextual information when combined with your own data, helping to speed the time to detect, identify, and triage malicious or anomalous activity.



TimeGenerated [UTC]	Con...	TrafficDirection	Country	Latitude	Longitude	Comp...	MaliciousIP	IndicatorThre...	Severity	ReportReferenceLink
4/19/2021, 10:00:38.164 AM	75	InboundOrUnknown	Germany	48.14	11.59	[redacted]	185.53.178.50	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 11:15:45.606 AM		InboundOrUnknown	United States	27.94	-82.52		199.59.242.153	C2		https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:35:42.301 AM	75	InboundOrUnknown	United States	27.94	-82.52	[redacted]	199.59.242.153	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:45:42.529 ...	75	InboundOrUnknown	United States	27.94	-82.52	[redacted]	199.59.242.153	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:48:59.231 AM	75	InboundOrUnknown	United States	42.51	-71.2	[redacted]	66.96.149.23	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:21:51.030 AM	75	InboundOrUnknown	Poland	51.08	17.05	[redacted]	87.98.239.4	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:55:42.244 ...	75	InboundOrUnknown	United States	27.94	-82.52	[redacted]	199.59.242.153	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:28:29.163 AM	75	InboundOrUnknown	United States	42.51	-71.2	[redacted]	66.96.149.23	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:42:07.079 A...	75	InboundOrUnknown	Germany	48.14	11.59	[redacted]	185.53.178.50	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:28:40.438 ...	75	InboundOrUnknown	United States	39.11	-94.54	[redacted]	34.98.99.30	C2	2	https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 11:17:16.554 AM		InboundOrUnknown	United States	27.94	-82.52		199.59.242.153	C2		https://interflowinternal.azure-api.net/reporting/a
4/19/2021, 10:10:42.053 AM	75	InboundOrUnknown	Germany	48.14	11.59	[redacted]	185.53.178.50	C2	2	https://interflowinternal.azure-api.net/reporting/a

Fig. 19. Example IOCs

CTI can be imported to Azure Sentinel through many methods, but two of the most common are via the built-in TAXII client to source from a wide array of open-source and paid STIX feeds or via an external threat intelligence platform (TIP) and the Azure Sentinel Platforms data connector.



```
ThreatIntelligenceIndicator
| where TimeGenerated > ago(24h)
| limit 10
```

Completed

Drag a column header and drop it here to group by that column

	TimeGenerated...	Action	ApplicationId	AzureTenantId	Description
>	4/7/2020, 7:21:02....	alert	7E7888EC-916...	72f988bf-86f1-...	TS ID: 55475482452; iType: suspicious_dom...
>	4/7/2020, 7:21:03....	alert	7E7888EC-916...	72f988bf-86f1-...	TS ID: 55474479406; iType: suspicious_dom...
>	4/7/2020, 7:21:03....	alert	7E7888EC-916...	72f988bf-86f1-...	TS ID: 55478096090; iType: phish_domain; S...

TI map IP entity to AzureActivity

Medium
Severity

Scheduled
Rule Type

Description
Identifies a match in AzureActivity from any IP IOC from TI

Data sources
Threat Intelligence Platforms (Preview)
ThreatIntelligenceIndicator --

Azure Activity
AzureActivity —

Tactics
Impact

Rule query

```
let dt_lookBack = 1h;
let ioc_lookBack = 14d;
ThreatIntelligenceIndicator
| where TimeGenerated >= ago(ioc_lookBack) and Exp
| where Active == true
// Picking up only IOC's that contain the entities
```

WatchList 375K

Malware 126K

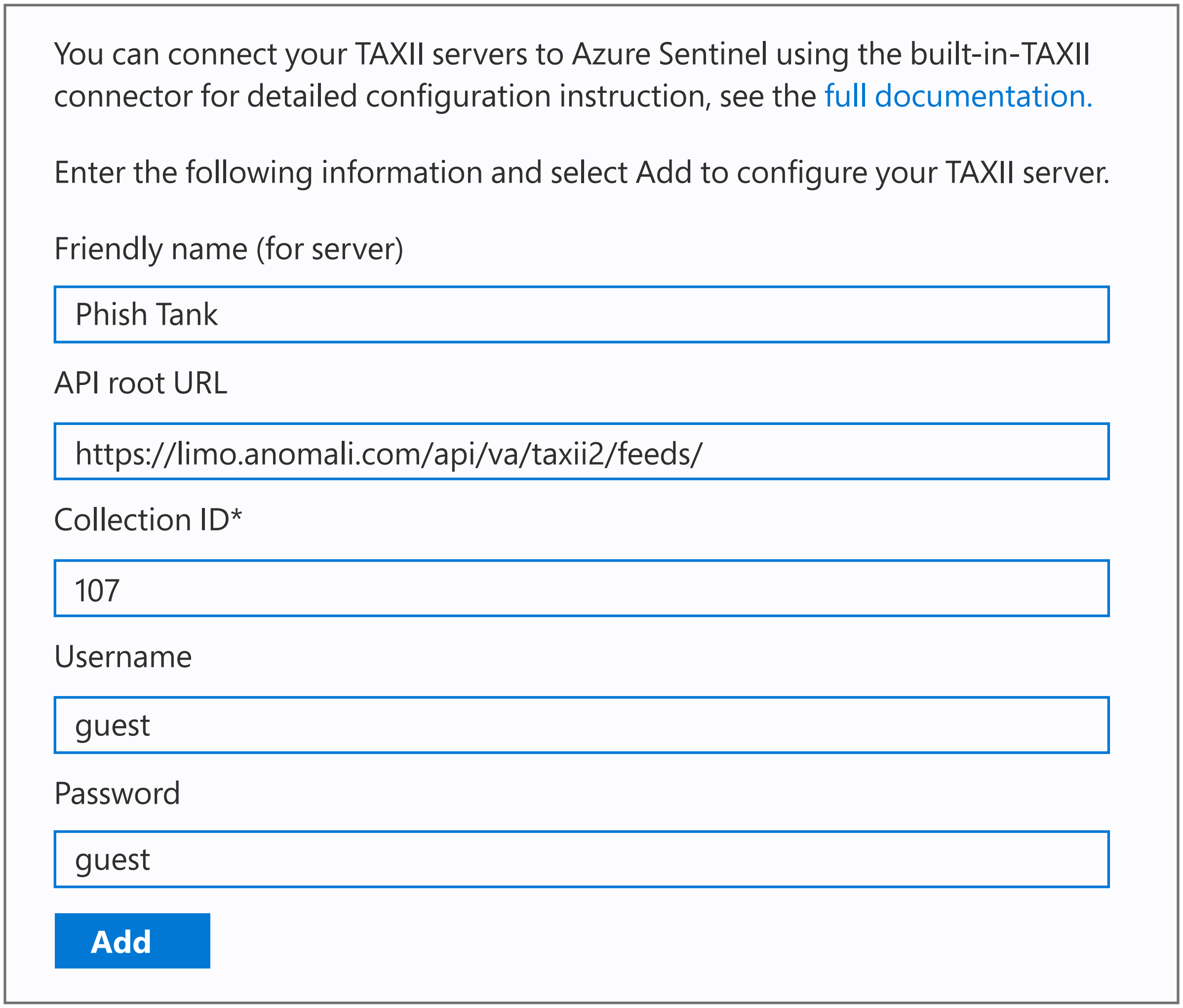
CryptoMining 11.1K

MaliciousUrl 1.58K

C2 849

Fig. 20. Azure Sentinel threat intelligence flow

To import using the Azure Sentinel TAXII data connector, you will need to know the advertised TAXII API root and collection ID. These are generally published in the vendor documentation and often include a discovery endpoint to help identify all available collection IDs.



You can connect your TAXII servers to Azure Sentinel using the built-in-TAXII connector for detailed configuration instruction, see the full documentation.

Enter the following information and select Add to configure your TAXII server.

Friendly name (for server)

Phish Tank

API root URL

https://limo.anomali.com/api/va/taxii2/feeds/

Collection ID*

107

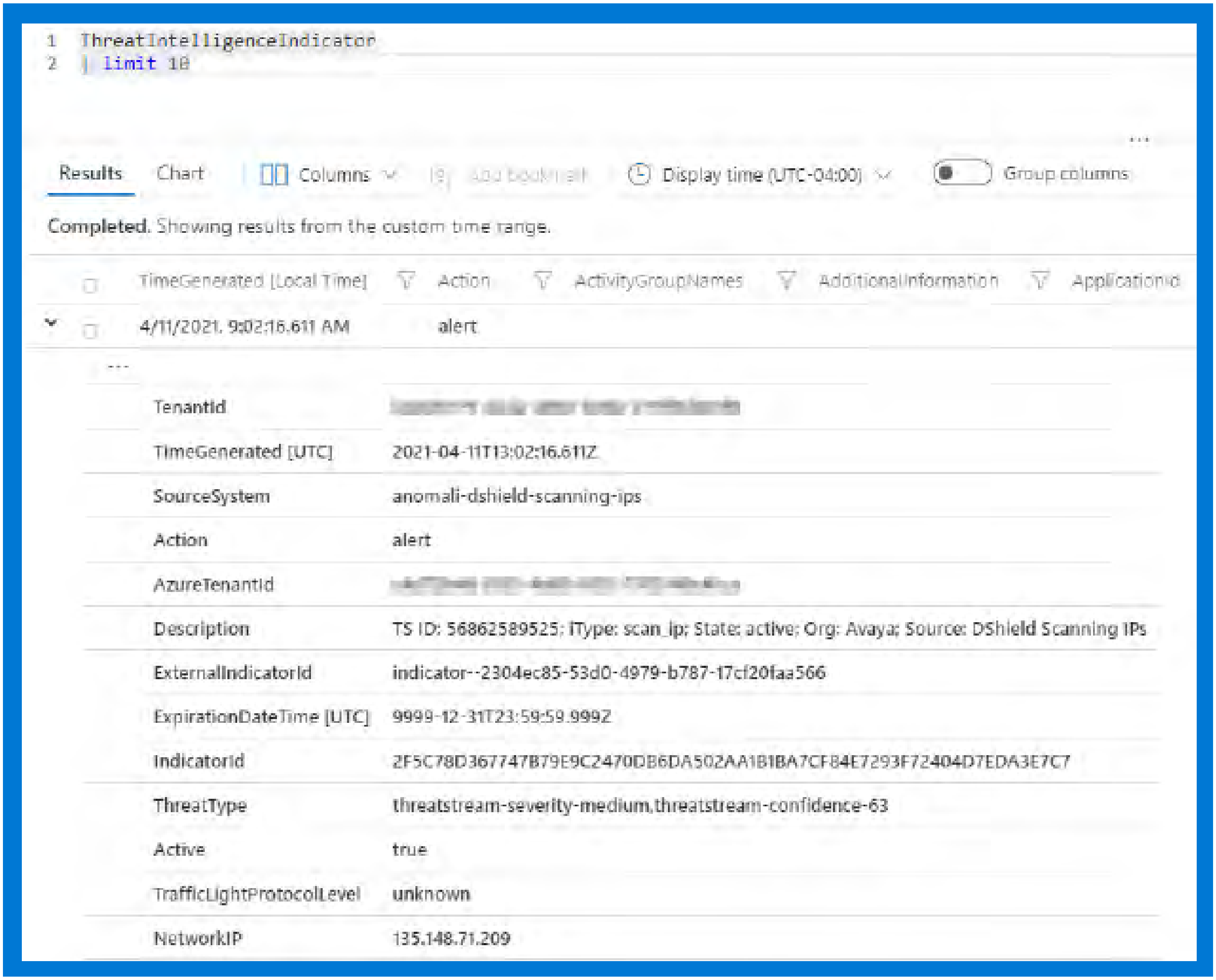
Username

guest

Password

guest

Fig. 21. Configuring a TAXII data connector



```
ThreatIntelligenceIndicator
| limit 10
```

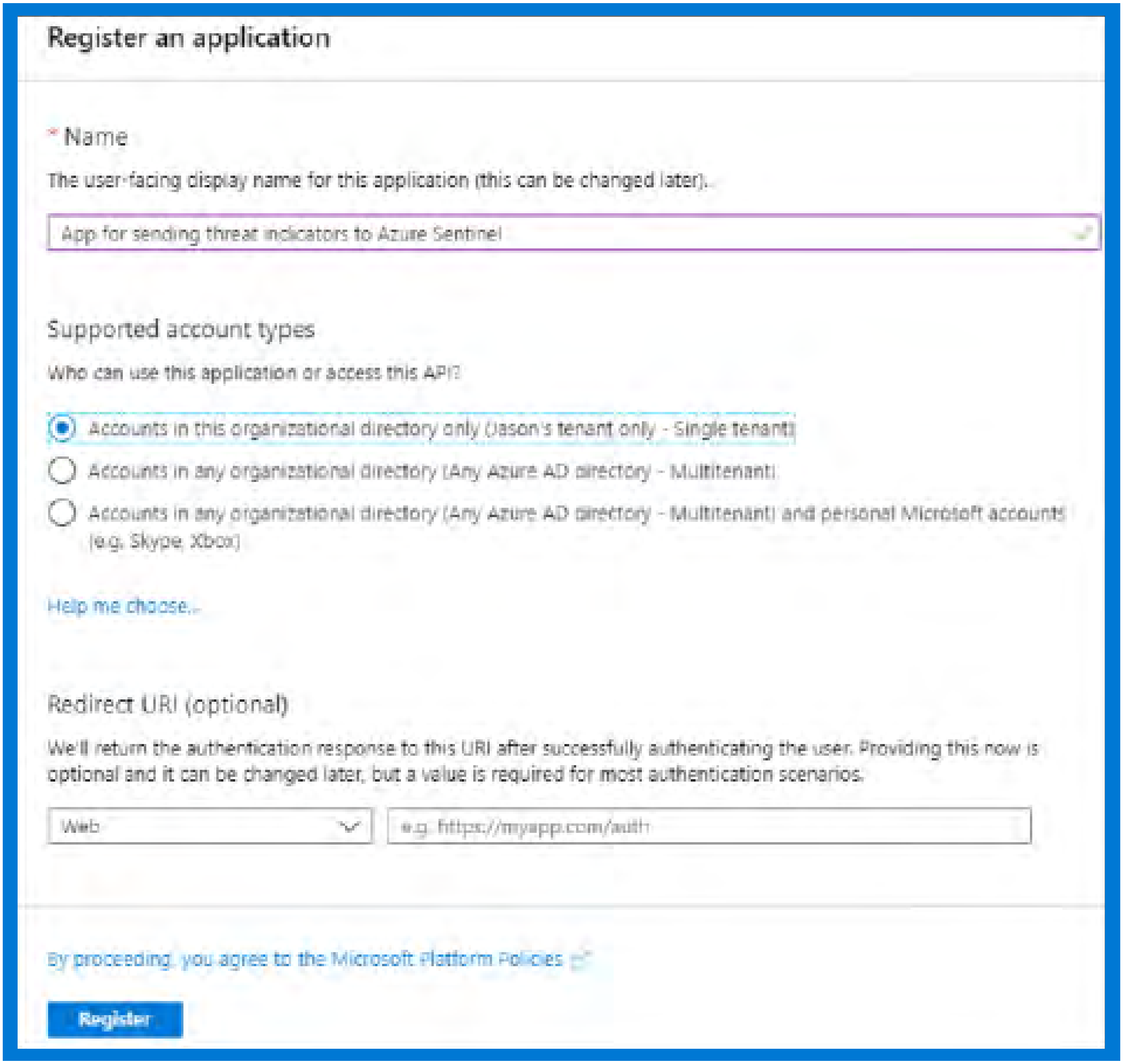
Completed. Showing results from the custom time range.

TimeGenerated [Local Time]	Action	ActivityGroupNames	AdditionalInformation	ApplicationId
4/11/2021, 9:02:16.611 AM	alert			

TenantId	[redacted]
TimeGenerated [UTC]	2021-04-11T13:02:16.611Z
SourceSystem	anomali-dshield-scanning-ips
Action	alert
AzureTenantId	[redacted]
Description	TS ID: 56862589525; iType: scan_ip; State: active; Org: Avaya; Source: DShield Scanning IPs
ExternalIndicatorId	indicator--2304ec85-53d0-4979-b787-17cf20faa566
ExpirationDateTime [UTC]	9999-12-31T23:59:59.999Z
IndicatorId	2F5C78D367747B79E9C2470DB6DA502AA1B1BA7CF84E7293F72404D7EDA3E7C7
ThreatType	threatstream-severity-medium,threatstream-confidence-63
Active	true
TrafficLightProtocolLevel	unknown
NetworkIP	135.148.71.209

Fig. 22. Sample threat intelligence indicator record (ingested via TAXII data connector)

TIPs are external platforms that allow organizations to aggregate feeds from a variety of sources and curate the collected data to deduplicate redundancies. These platforms generally include mechanisms to apply these IOCs to security solutions such as blocking known bad websites at your firewall or stopping malicious file hashes at your endpoints. Azure Sentinel can ingest this same data using the TIPs data connector. This uses an API and Application (client) ID, Directory (tenant) ID, and client secret from your TIP to connect and send threat indicators for use in Azure Sentinel.



Register an application

* Name

The user-facing display name for this application (this can be changed later).

App for sending threat indicators to Azure Sentinel

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Jason's tenant only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

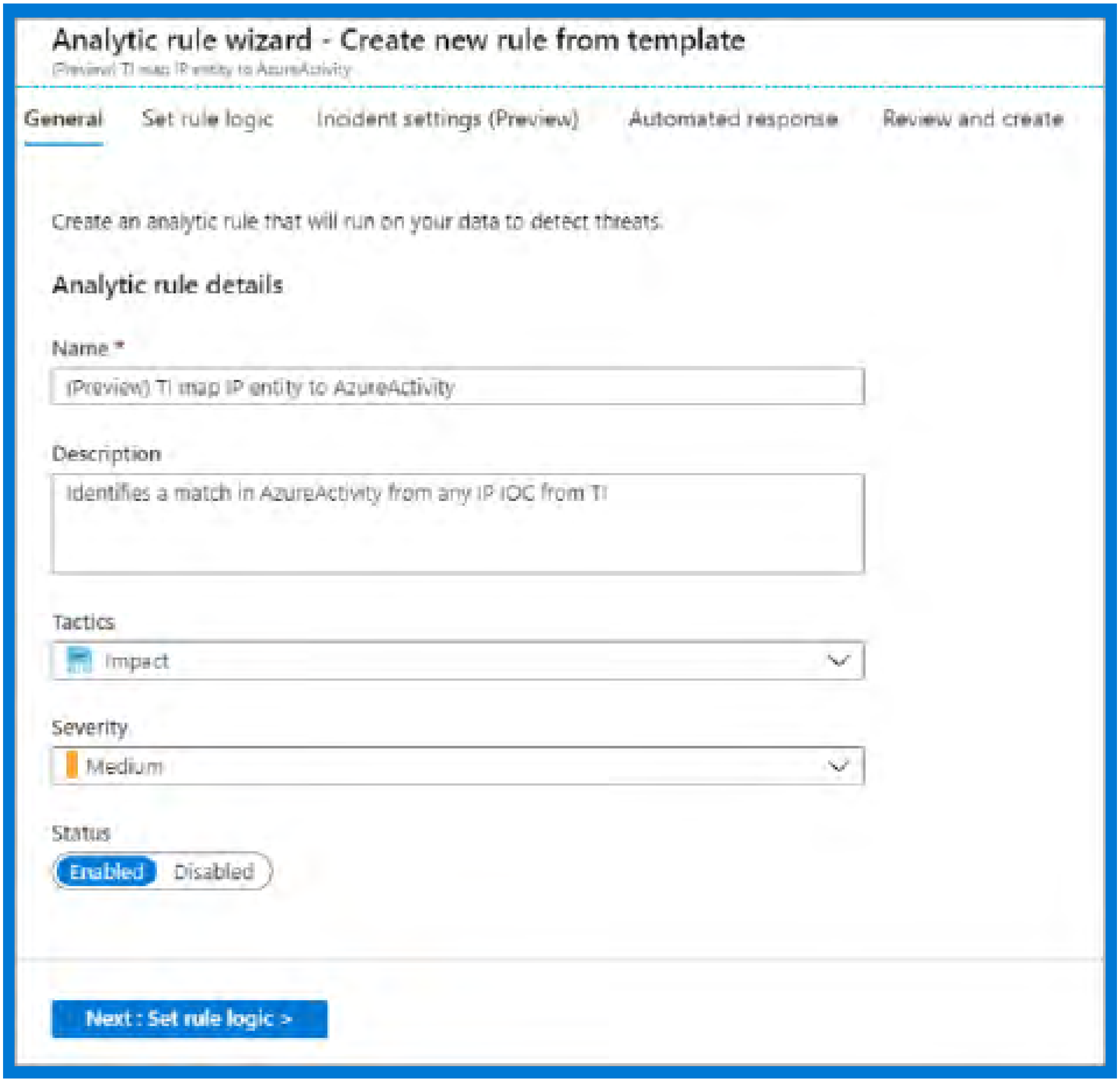
Web | e.g. https://myapp.com/auth

By proceeding, you agree to the Microsoft Platform Policies

Fig. 23. Connecting Azure Sentinel to a TIP

The most important use case for threat indicators in Azure Sentinel is to drive the analytics that correlate events with IOCs to generate security alerts, incidents, and automated responses. Azure Sentinel Analytics is used to create rules that trigger on schedule to generate alerts. You start with a query that includes your required parameters, configure the frequency, and the results that will generate security alerts and incidents. Azure Playbooks, based on workflows built into Azure Logic Apps, can help automate and orchestrate your response to IOCs. For example, you can add contextual information to an incident with the enriched data added as comments, including rich markup and HTML formatting, to aid your analysts and incident responders before they have even opened the incident.

Azure Sentinel includes dozens of built-in analytic rules. The data from TAXII or TIP ingest options are stored initially in the Azure Sentinel table Threat Intelligence Indicator, but the built-in rules are easily modified to reference your own custom tables.


Analytic rule wizard - Create new rule from template

(Preview) TI map IP entity to AzureActivity

Create an analytic rule that will run on your data to detect threats.

Analytic rule details

Name *: (Preview) TI map IP entity to AzureActivity

Description: Identifies a match in AzureActivity from any IP IOC from TI

Tactics: Impact

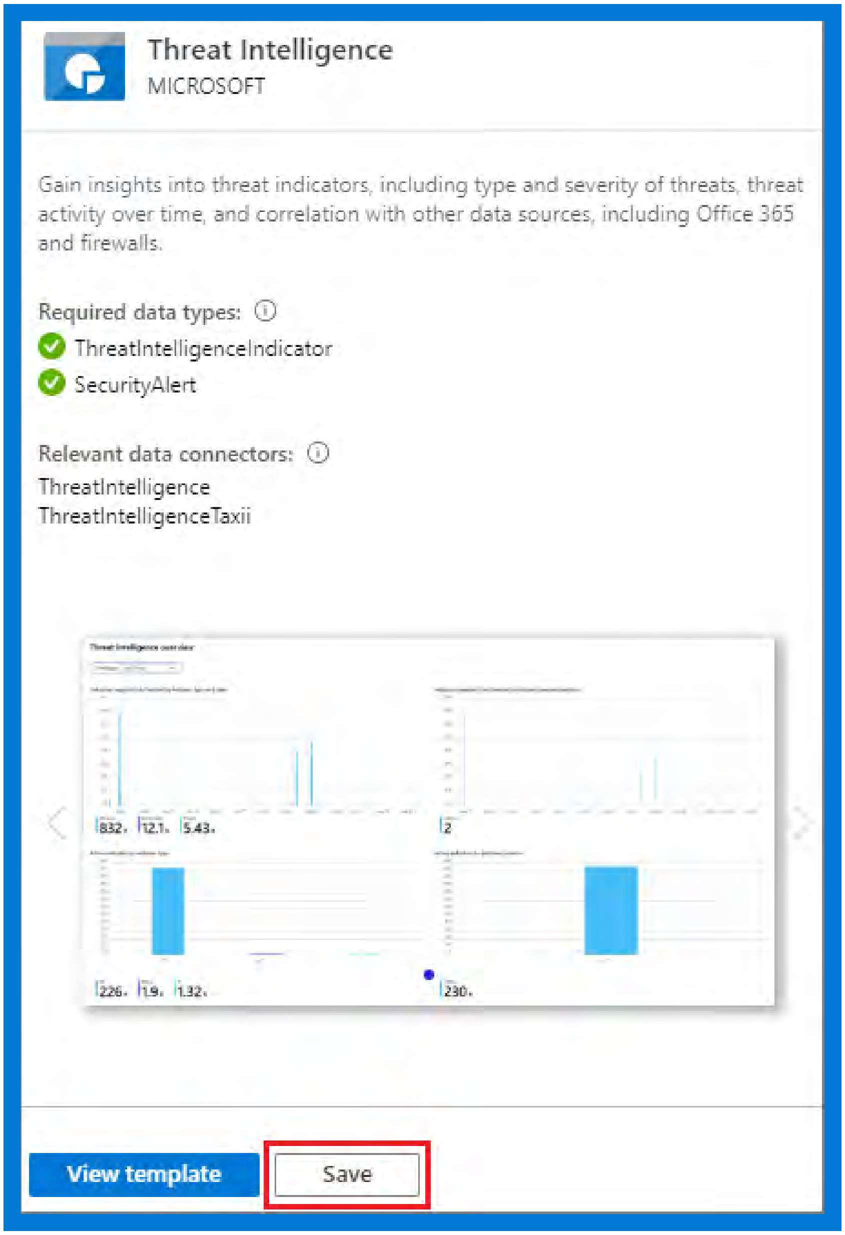
Severity: Medium

Status: [✓ Enabled] Disabled

Fig. 24. Creating a new Azure Sentinel analytics rule from a template

Azure Sentinel's Threat Intelligence Workbook is an excellent starting point to begin visualizing your threat indicator data. The Threat Intelligence Workbook shows you all alerts and incidents related to your TI sources and can help to illustrate those that provide the most value for your organization.

Of course, the provided templates offer an entry point for you to customize the included templates to meet your specific business needs or create entirely new dashboards that can combine data from multiple sources to maximize the visualization of your data.



Threat Intelligence

MICROSOFT

Gain insights into threat indicators, including type and severity of threats, threat activity over time, and correlation with other data sources, including Office 365 and firewalls.

Required data types: ⓘ

✓ ThreatIntelligenceIndicator

✓ SecurityAlert

Relevant data connectors: ⓘ

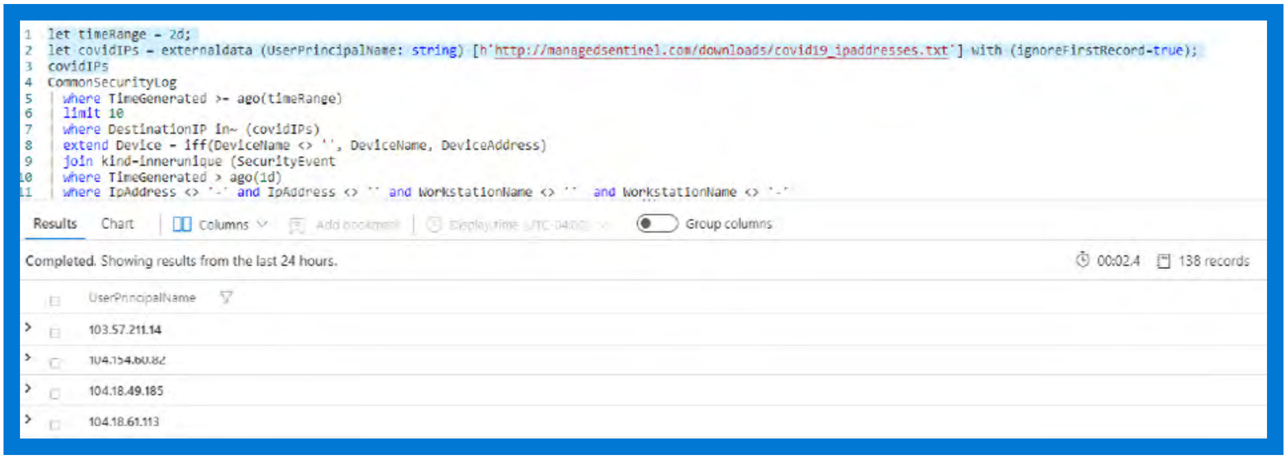
ThreatIntelligence

ThreatIntelligenceTaxii

Fig. 25. Preparing to customize the Azure Sentinel Threat Intelligence Workbook

Azure Sentinel also provides several additional ways to ingest third-party TI:

- Custom logs. If the TI is available as a local file (i.e., a comma separated variable [CSV] file), the MMA can be used to collect the updated data and send it to an Azure Sentinel custom log. In addition to MMA functionality, any custom log can be brought into Azure Sentinel using other log management solutions that integrate with Azure Log Analytics, such as Logstash and Fluentd.
- Watchlists. As one of the Azure Sentinel features, watchlists allow for the importing of CSV data and its use in an Azure Sentinel table to integrate with alert rules and queries.
- External data. If the TI data is available as a URL (e.g., downloadable CSV files, Azure blobs), the KQL function externaldata() can be used to download it on demand and use it as a temporary Azure Sentinel table.



```
let timeRange = 2d;
let covidIPs = externaldata (UserPrincipalName: string) [h'http://managedsentinel.com/downloads/covid19_ipaddresses.txt'] with (ignoreFirstRecord=true);
covidIPs
CommonSecurityLog
| where TimeGenerated >= ago(timeRange)
| limit 10
| where DestinationIP in~ (covidIPs)
| extend Device = iff(DeviceName <> '', DeviceName, DeviceAddress)
| join kind=innerunique (SecurityEvent
| where TimeGenerated > ago(1d)
| where IpAddress <> '-' and IpAddress <> '' and WorkstationName <> ''  and WorkstationName <> '-'
```

Completed. Showing results from the last 24 hours.

00:02.4 | 138 records

UserPrincipalName
103.57.211.14
104.154.60.82
104.18.49.185
104.18.61.113

Fig. 26. Sample use of KQL externaldata() for retrieval of Covid19-related IOCs

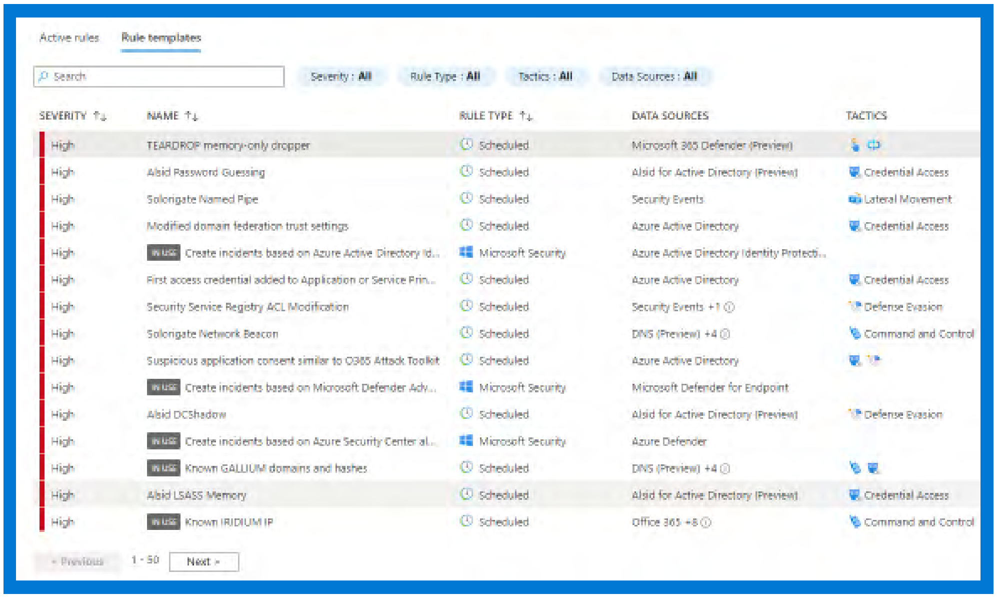
Additional resources:

- Threat indicators for cyber threat intelligence in Azure Sentinel
- Bring your threat intelligence to Azure Sentinel - Microsoft Tech Community
- Connect threat intelligence data to Azure Sentinel | Microsoft Docs
- Anomali Limo Free Intel Feed
- Tutorial: Set up automated threat responses in Azure Sentinel

Deploying alert rules

As the core functionality of an SIEM, the configuration of detection rules is a critical component of any Azure Sentinel deployment. Azure Sentinel includes many built-in alert rules templates, covering the array of typical log source, with new alert rules added regularly.

Additional rule templates can be obtained through the Azure Sentinel Community, where both Microsoft and third-party contributors publish new content. The proposed alert rules are reviewed by the community and those found valuable published in Azure Sentinel.



SEVERITY ↑↓	NAME ↑↓	RULE TYPE ↑↓	DATA SOURCES	TACTICS
High	TEARDROP memory-only dropper	Scheduled	Microsoft 365 Defender (Preview)	
High	Alsid Password Guessing	Scheduled	Alsid for Active Directory (Preview)	Credential Access
High	Solorigate Named Pipe	Scheduled	Security Events	Lateral Movement
High	Modified domain federation trust settings	Scheduled	Azure Active Directory	Credential Access
High	IN USE Create incidents based on Azure Active Directory Id...	Microsoft Security	Azure Active Directory Identity Protecti...	
High	First access credential added to Application or Service Prin...	Scheduled	Azure Active Directory	Credential Access
High	Security Service Registry ACL Modification	Scheduled	Security Events +1 ⓘ	Defense Evasion
High	Solorigate Network Beacon	Scheduled	DNS (Preview) +4 ⓘ	Command and Control
High	Suspicious application consent similar to O365 Attack Toolkit	Scheduled	Azure Active Directory	
High	IN USE Create incidents based on Microsoft Defender Adv...	Microsoft Security	Microsoft Defender for Endpoint	
High	Alsid DCShadow	Scheduled	Alsid for Active Directory (Preview)	Defense Evasion
High	IN USE Create incidents based on Azure Security Center al...	Microsoft Security	Azure Defender	
High	IN USE Known GALLIUM domains and hashes	Scheduled	DNS (Preview) +4 ⓘ	
High	Alsid LSASS Memory	Scheduled	Alsid for Active Directory (Preview)	Credential Access
High	IN USE Known IRIDIUM IP	Scheduled	Office 365 +8 ⓘ	Command and Control

Fig. 27. Azure Sentinel alert rule templates

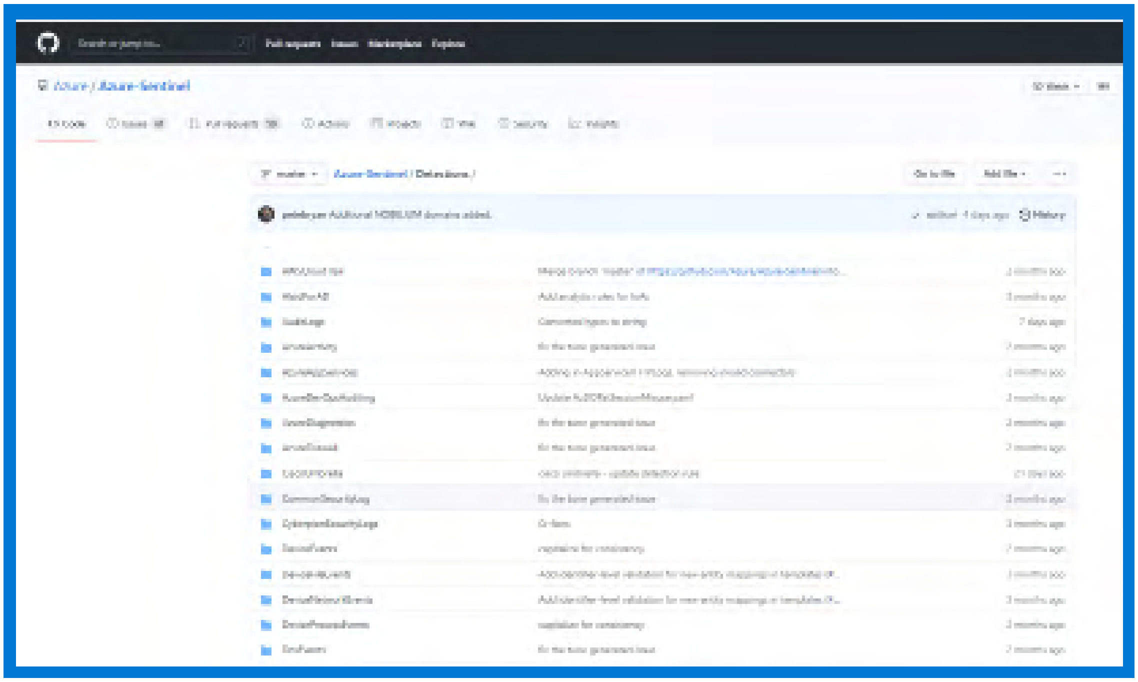

Pull requests Issues Marketplace Explore
Azure / Azure-Sentinel
Code
Azure-Sentinel / Detections /
Go to file
Add file
History

Fig. 28. Azure Sentinel Community GitHub repository

Rule templates can be deployed as Azure Sentinel scheduled alerts in the Azure Sentinel portal. These are fully customizable for use cases that extend beyond out-of-the-box detections. You may wish to avoid the temptation to enable too many at once, though, in that false positives are possible. In those cases, review the alert rule criteria and its triggering criteria to find how to tune the rule so that the detection results match original expectations. Continue to enable more rules, in controlled batches, over the course of the deployment.

All alert rules deployments should be included in regular reviews of their value and adjustments necessary to make them more relevant to the organization. As a cloud-based solution, Azure Sentinel provides frequent access to new content, and the alert rule deployment and tuning lifecycle should match the dynamics of your threat landscape. A weekly review is frequently recommended during the initial deployment, followed by monthly reviews as the environment enters a more stable state.

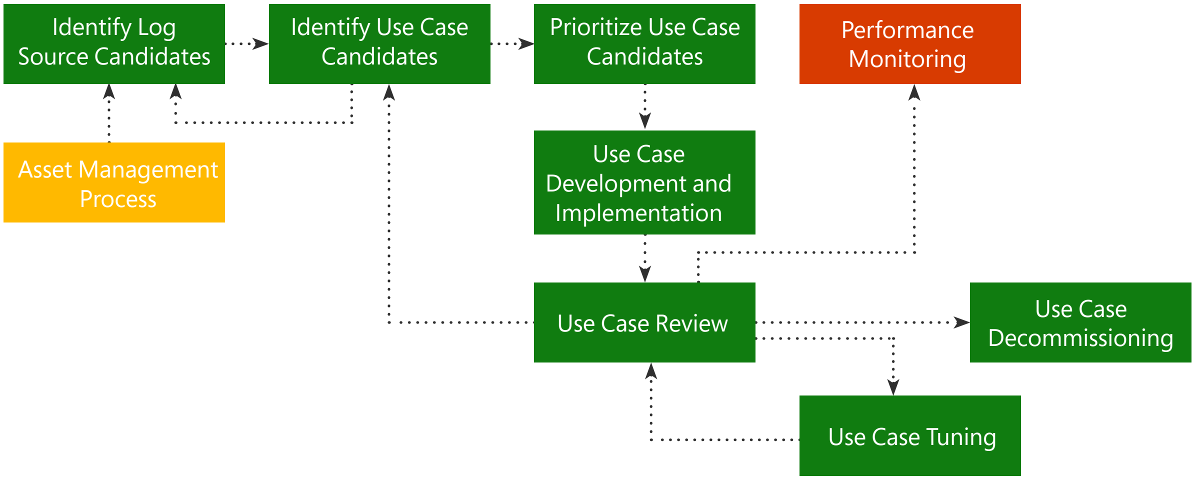

Identify Log Source Candidates
Identify Use Case Candidates
Prioritize Use Case Candidates
Performance Monitoring
Asset Management Process
Use Case Development and Implementation
Use Case Review
Use Case Decommissioning
Use Case Tuning

Fig. 29. Sample SIEM use-case lifecycle

If Azure Sentinel is deployed or managed by a third party, such as a Managed Security Services Provider (MSSP) providing Azure Sentinel management services, additional alert rules might be available from their catalog along with more automated deployment methods or monitoring and tuning advice. Given the criticality of alert rules to core SIEM functionality, seeking consulting advice from third-party SIEM specialists may be advisable.

In our experience:

- Avoid the temptation to enable too many alert rules at once to avoid a potential influx of false positive alerts. If left alone, these can cause "the boy who cried wolf" alert fatigue among your analysts. Start with a handful, tune the associated alert rule logic to match the expectations of your organization, and then enable more.
- Alert tuning may feel a bit like a dark art at first but take one alert rule at a time and step through the logic in the flow and consider the trigger against the associated fields. You will likely find *the* field (or even an entire table) that should be added/removed to eliminate the false positive and have the alert rule reflect your specific use case better. Remove (or add) that section and test. The more times you run through this exercise, the better you will learn your own data and the faster this will become.
- Custom tables and custom data are generally not included in default alert rules; these may need to be added or have custom rules created to maximize the effectiveness of the newly sourced data.
- Professional services deployments are a good way to quickly deploy your instance but be sure to knowledge transfer such that staff learn the "what, when, and how" about alert rule tuning for future use cases.

Some helpful links about learning Azure Sentinel, including alert rule tuning:

- Microsoft Learning Path: Cloud-native security operations with Azure Sentinel
- Become an Azure Sentinel Ninja: The complete level 400 training
- Join discussions about Azure Sentinel

Migration from existing SIEM solutions

As Azure Sentinel is a relatively new entrant to the SIEM market, most projects involve not only the deployment of Azure Sentinel but also the migration and cutover from another legacy SIEM. Based on our experience leading numerous SIEM migration projects from a wide variety of SIEM solutions to Azure Sentinel, we have compiled some recommended approaches presented in the following three scenarios.

Scenario 1

Legacy SIEM solution in place

This involves migrating from an existing, older, possibly obsolete, SIEM where the vendor may not have kept up with the latest developments or the architecture is difficult/expensive to scale with growing log volume. In many cases, the existing SIEMs are already overloaded with a large volume of logs or a significant licensing expense without providing expected or satisfactory security visibility. Quite often, these scenarios involve a need for additional, potentially significant costs related to a required increase in licensing or a need to upgrade aging hardware.

Use case migration

The legacy use cases will have to be inventoried and the ones that are deemed to provide value should be documented and converted in Azure Sentinel alert rules or playbooks. Fortunately, for the most common ones, Azure Sentinel already provides support through a large array of built-in alert rule templates.

Due to differences in the overall SIEM design and the major shift from on-premises to cloud-based technology, the effort to convert the existing SIEM use cases may vary, so the requirements should be documented as abstractly as possible to allow for the differences between the analysis/alerting processes. It is common to become focused on reproducing a specific legacy alert in Azure Sentinel verbatim rather than target the intent of the original use case.

In our experience:

- Many existing use cases in production SIEM solutions have an equivalent in the existing rule base for Azure Sentinel or can be re-created simply with a KQL query.
- Many organizations performing a migration to Azure Sentinel choose to completely discard existing SIEM use cases due to lack of maintenance and relevance with the current log sources and start fresh in Azure Sentinel.
- SIEM analysts experienced with a specific legacy platform may desire training and assistance when transitioning concepts applicable to legacy technology over to cloud-native solutions. Partnering the analysts with professional services during the implementation of the new Azure Sentinel deployment may help streamlined adoption of the new platform.

Particular attention must be provided to any log collection agents present in the legacy SIEM solution. The typical agents are deployed on endpoints such as Windows or Linux servers, with each SIEM solution having its own design and, therefore, requiring potentially significant changes about how such agents are deployed and monitored. For example, a project may include pull versus push agents, agentless solutions, centralized collection of logs using Windows Event Log Forwarder, or Linux syslog versus native SIEM log collection method.

To avoid last moment major redesigns, the differences in agent deployment and log collection must be very well understood and the relevant infrastructure staged to accommodate. A small pilot or development environment should be deployed to address challenges that might not be easily visible on paper.

One very important difference between legacy on-premises SIEMs and Azure Sentinel is the need to send the log data to the cloud. This requires the proper network infrastructure in place with internet or VPN connections that can provide sufficient bandwidth as well as opening the necessary firewall ports for outbound access to Azure. Azure Sentinel provides multiple options for aggregating the logging data using a single collector or gateway before sending to the cloud repository for analysis.

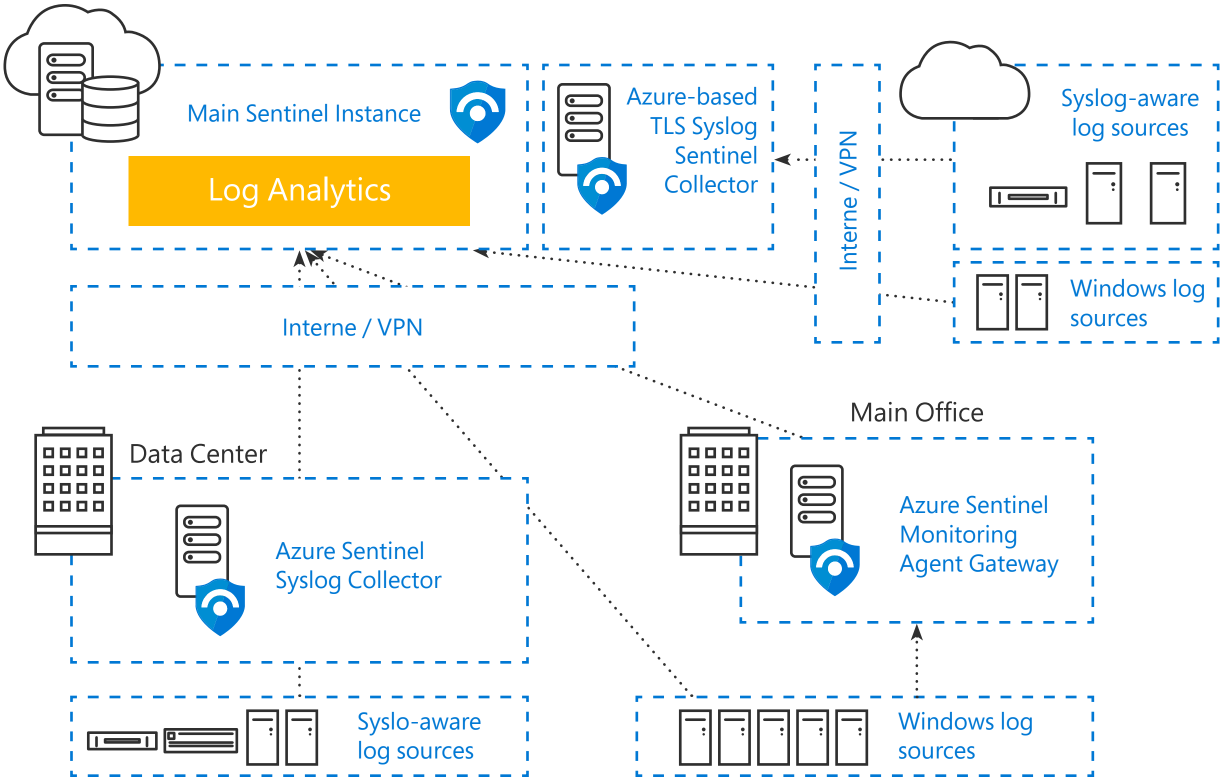


Fig. 30. Azure Sentinel collectors for on-premises log sources

A testing plan should be established for the removal of existing agents, configuration of MMA, and the migration of syslog-based log sources from the legacy collector to the Azure Sentinel. In some situations, the collectors can be recycled, but it is seldom recommended; choose fresh installs wherever feasible.

The existing SIEM can be kept in place until all resources have been migrated and the use cases fully tested in Azure Sentinel. Special consideration must be given to the data stored in the legacy SIEM, with provisions made to allow access to historical data based on log retention policies or compliance needs. For example, the Payment Card Industry Data Security Standard (PCI DSS) requires access to the prior year of in-scope log data.

In our experience:

- Care should be taken to manage risk in cases where the legacy SIEM is approaching license/ support renewal, the organization has an imminent audit, or the organization has insufficient resources to devote to properly managing the migration. Decisions under pressure or duress can result in risk to the enterprise.
- Evaluate the true nature of logs ingested by the legacy SIEM. For example, it may ingest 20 GB/ day due to some limitation, but the true log volume may be much greater. Miscalculation of these estimates can lead to unexpected costs during the Azure Sentinel deployment.
- If a legacy SIEM has been maintained by a third party, such as an MSSP, be sure to check into any applicable contractual agreements around service exit provisions.

Scenario 2

Legacy SIEM in place and third-party analytics platform in deployment

Another migration scenario is when the legacy SIEM is already in the process of being replaced by a new solution, but there are limitations around connecting with Azure Cloud resources, and there are critical Azure-based projects going live that need full SIEM coverage. Azure Sentinel can be deployed quickly to provide security analytics for such situations.

Most Azure-based resources have the capability to stream logs directly into a Log Analytics workspace and, through that, make the logs readily available to Azure Sentinel. This allows a possible reduction in time (from weeks/months to days) required to onboard raw logs from Azure resources into an SIEM. Depending on the volume of logging data, additional bandwidth costs related to cloud egress traffic can be avoided.

For certain Microsoft/Azure log sources, the ingestion of logging data into Azure Sentinel is free of charge so additional savings can be obtained by reducing the load on the on-premises SIEM. One such example is the onboarding of Office 365 logs—a non-billable log in Azure Sentinel.

Azure Sentinel has the capability to export new incidents into Events Hub and make the data available for third-party analytical platforms. Modern SIEM solutions typically can retrieve events from a data streaming platform like Azure Events Hub.

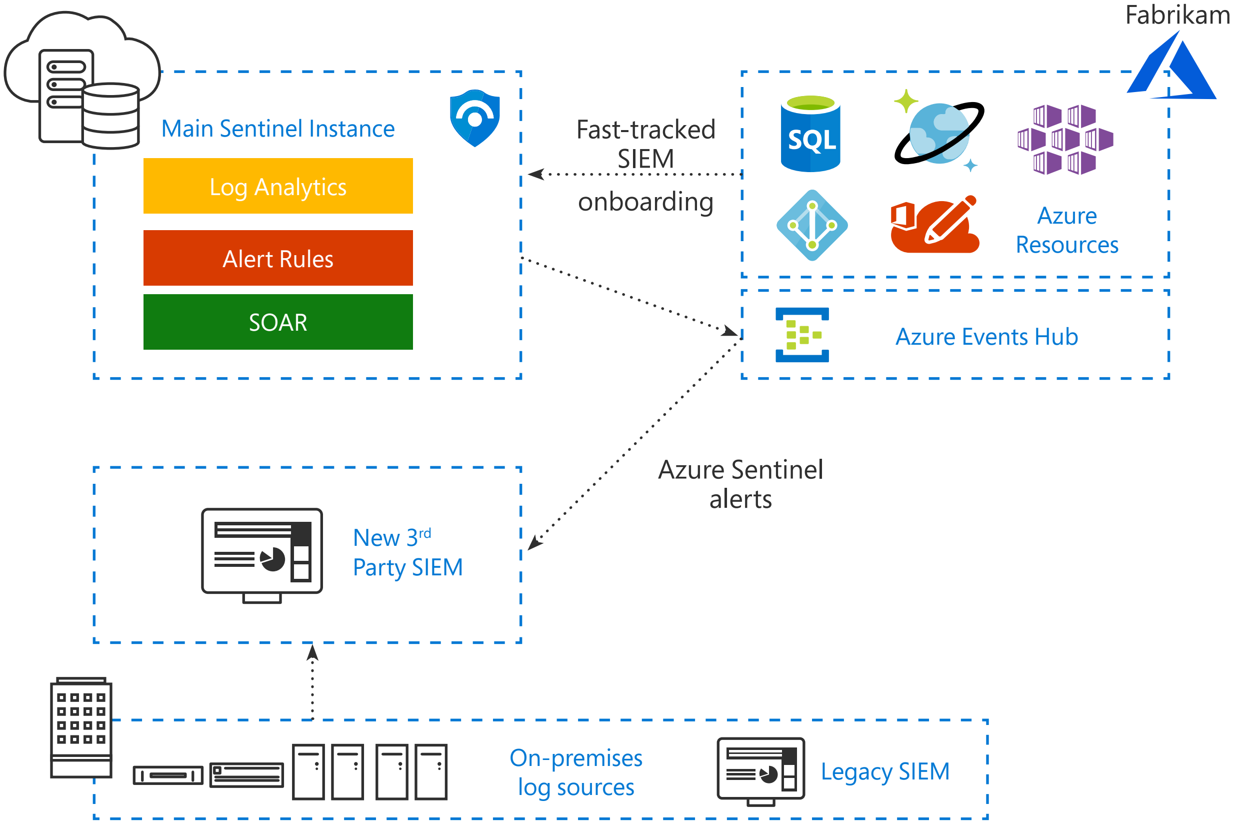


Fig. 31. Deployment of Azure Sentinel SIEM for fast-tracked analytics for Azure-based resources

A long-term strategy must be developed to guide day-to-day decisions about onboarding log sources in Azure Sentinel versus the third-party solution. In many cases the argument for aggregation to a single repository is the ability to correlate data, but in quite a few situations especially for isolated application environments, the aggregation does not bring enough additional value to justify the increase in cost and complexity.

Scenario 3

Migration from a limited logging/analytics platform such as syslog to Azure Sentinel

Organizations with basic log collection/analytics capabilities are either a size that enforced a limited IT security budget or a larger organization that has an established raw log collection infrastructure but needs to add advanced security analytics capabilities.

For new deployments, the value of each type of log source (from a security perspective) must be analyzed, and the ingestion of raw logs into the new SIEM must be prioritized on a cost/value basis. Keep in mind that Azure Sentinel log ingestion is free of charge for Office 365 logs, for M365 security incidents, and several other log sources (see the full details earlier).

It is critical to onboard log sources gradually, understand their value from a security analysis perspective, and decide on their ingestion and retention strategy. Some common log sources that need increased scrutiny from a cost/value perspective are:

- Firewall traffic logs (allowed and denied traffic)
- Windows Active Directory logs (Windows Security event logs)
- Azure Diagnostics (raw logs from Azure resources)
- Web and WAF logs

The risk appetite, compliance requirements, and available budget vary from organization to organization, but these are the typical factors that affect the decisions related to log ingestion and retention.

If a raw log collection and retention capability exists, it can be preserved for long-term retention or to meet compliance requirements. If necessary, older logs can be ingested in Azure Sentinel on an impromptu basis to perform more advanced analytics through KQL or Azure Sentinel Notebooks.

Additional resources:

- Webinar: Best practices converting detections rules from existing SIEM to Azure Sentinel
- How do you export QRadar offenses to Azure Sentinel
- Best practices for migrating detection rules from ArcSight, Splunk and QRadar to Azure Sentinel - Microsoft Tech Community
- Splunk to Kusto Query Language map

Azure Sentinel—Business Considerations

Cost management

Cost management for Azure Sentinel is inextricably tied to organizational risk management. Ingesting incremental data into Azure Sentinel from various network components allows for the creation of analytic rules to detect attacker behavior. However every byte of data ingested into Log Analytics carries a cost. In this section, we aim to provide some practical guidance for building and evaluating a business case for adopting Azure Sentinel.

Evaluating your data ingestion against use cases

The expense of a given log source ingested into Azure Sentinel should be evaluated against a commensurate benefit—the data aids in visibility for cyberattacks, can detect or prevent a data breach, or is simply required due to regulatory compliance. The ingestion of many types of log sources can incur additional expense for your Azure Sentinel deployment; therefore, a degree of cost/benefit analysis is required. For example, ingesting all data from the APIs of the top 100 employee-accessed SaaS applications could aid in certain investigations, but cost would likely be prohibitive and would likely outweigh the potential benefit via increased security visibility, threat mitigation, or fulfilling a compliance requirement.

Generally, we advise to identify the applications that pertain to significant degrees of business risk. Project teams will want to survey all possible data sources and analyze variables such as log volume or anticipated risk mitigation. A log source with an unusually high volume of log data weighed against a relatively small number of potential security use cases could serve as an example to exclude from project scope and Azure Sentinel budget. Ultimately, this decision lies in the hands of the stakeholders and decision makers and should require formal sign off for decisions by the enterprise risk management owners.

Log ingestion strategies

Any network device, endpoint, or application has the potential to generate log data, from basic information to verbose debugging-level details. The value of the information captured in the logs depends on the type of log source; its association with users, computers, and applications; and the level of detail built into the logging mechanism by the device or application vendor.

The requirements around logging level, the type of information captured in the logs, and their retention requirements are driven by the organization's information security policies that are the result of the cybersecurity governance process. In most cases, the policies are based on industry best practices and compliance requirements with standards (e.g., PCI, International Organization for Standardization [ISO] 27001, National Institute of Standards and Technology [NIST]). For many such standards, the policies around logging are left relatively vague and sometimes subject to an auditor interpretation. As a principle, any logs related to the CIA triad are in scope of information security policies for logging, but most of the "must-have" requirements are related to auditing of user activities and integrity of IT systems and their processed data. Log collection must be treated as any security control and be driven by feasibility and its overall contribution to the organization's security stance. A common logging policy is to have the logs available for online analysis for 90 days with 1 year offline/archived to slower or less expensive storage.

From a security perspective, the logs historically captured were those from perimeter security controls, such as firewalls and proxy servers, authentication/authorization servers, or Windows Active Directory Security event logs. With relatively few connections to untrusted locations, bandwidth limitations, plain text, or unencrypted traffic, limited internet resources and virtually nonexistent logging from endpoint security controls resulted in a log analysis strategy centered around ingestion and processing of perimeter security controls.

The exponential increase in internet usage has introduced many new devices with internet connectivity: mobile, Bring-Your-Own-Device (BYOD), Internet of Things (IoT), SaaS. The commensurate adoption of encryption gradually reduced the visibility of traditional security controls, with the endpoints increasingly becoming the single location where the activities of potentially malicious actors are detectable. Combined with the increase in logging data, the legacy approach to log analytics is no longer feasible. The "log everything" approach leads to either unmanageable costs or poor performance via quickly overwhelmed SIEM solutions. Our suggested approach is to analyze each type of log source in detail and weigh the costs versus benefits of ingestion for each type of log identified.

The analysis should consider not only the log source but also the logged field entries and their value from a threat detection perspective. Consider the level of effort to detect use cases in relation to compensating controls in the dynamic of the threat landscape that may provide equal or better visibility for the same cost. The following is a sample high-level analysis of log source volume versus threat detection value.

Log source	Log volume	Value for threat detection
Firewalls allowed traffic	High	Medium-low
Firewalls denied traffic	High	Low
Firewalls VPN	Medium	Medium
Intrusion prevention/detection system (IPS/IDS)	Low	High
URL filtering	High	Medium
Email security	Low	High
Windows Security events	Medium-High	High
AAA (Radius, terminal access controller access control system [TACACS])	Low	High
Cloud IAM	Medium	High
LAN/WAN	Low	Low
Cloud PaaS	High	Medium
Websites access	High	Low
Database audit tools	Low	High
Endpoint detection and response (EDR) (alerts/incidents)	Low	High
EDR (raw logs)	High	Low
Cloud security controls	Low	High
Vulnerability scanning	Low	High
File integrity	Low	High
Privileged Access Management (PAM)	Low	High
SD-WAN	High	Low
Multi-Factor Authentication (MFA)	Medium-Low	Medium

Detailed analysis example:

Firewall-allowed traffic

- Volume: High. An organization with 1,000 end users may generate 20 GB–30 GB of firewall log per day.
- Typical log data collected. Timestamp, source IP, destination IP, protocol/application, traffic (bytes/packets), firewall action (allow), firewall interfaces, firewall rule, and user.
- Typical alert rules. Matching with known malicious IPs, geolocation, volume of traffic (anomalies), number of concurrent connections, potential command and control (C&C) beaconing, and applications/protocols used.
- Value for threat detection: Low. Malicious actors rattle the cages of the internet, endlessly scanning for vulnerabilities. This, combined with dynamic IP address assignment, creates a scenario where one can expect several hit matches from known malicious IPs inbound. Matches outbound are noteworthy but, without other correlating log sources, are difficult to investigate.
- Optimization options. Firewall logging rules adjustments, filtering of types of log entries recorded by firewalls, filtering at syslog collector level, and adjustment of firewall logging format.

EDR (alerts/incidents)

- Volume: Low. Only potentially malicious activities are logged, should not create a significant volume of logs.
- Typical log data collected. Timestamp, endpoint host name, username, process (child/parent), hashes (SHA256, MD5), file names, operations, network connections (destination IP, host name, URL), remediation status, severity, confidence, threat name, threat description URL, and recommended remediation.
- Typical alert rules. Incidents matching high and critical severities, incidents with no remediation, repetitive remediations, and incidents affecting multiple users/hosts.
- Optimization options: Tuning of EDR solution, exclusion lists of known or benign incidents.

Windows security events

- Volume: Medium–High. Depending on the organization’s requirements around collecting data for user identity governance, the required logs may vary.
- Typical log data collected. User sign in/sign out, user creation, user disabled/enabled, password changes, group creation, group membership, process creation, and file access audit.
- Typical alert rules. Anomalous user logins, patterns matching brute force attacks/password spraying, user creation, activation of disabled users, addition to high-sensitive or privileged groups, and suspicious processes.
- Optimization options. Filtering of security events collected, configuration of logging policies (via group policies), filtering at the connector level, and filtering at the SIEM solution-level.

If certain log sources have the potential to generate large volumes of data (and associated analytics costs), additional compensating controls can be considered to provide the same or similar visibility at a lower cost. For example, detections around the processes collected from Windows Security Event logs are typically overlapping with detections provided by EDR solutions. An organization can decide to disable the logging of process creation on Windows servers based on

Azure Defender for Servers being deployed and covering potential malicious activities based on monitoring of processes created. Disabling the process creation audit may result in 50%70% Windows Security event log volume. In many cases, the compensating controls offer superior detection alerts and are more frequently updated by the solution vendor.

Real-world experience:

- Legacy log collection/retention policies enforcing the collection of high-volume, low-value logs. Reevaluate and consider updating corporate security policy.
- Misunderstanding logged data quality. Review both your log sources and inherent log fields for value in threat detections.
- Compliance may require long log retention volumes (e.g., 1 year for PCI logs). Consider reviewing your architecture for ways to reduce compliance scope.
- Log ingest technologies, especially syslog products, have a variety of filtering and parsing options. Assess your architecture to ensure your current solution meets your needs.
- Obtain senior management support to honestly review the value of your current logged sources. Staff may otherwise be unwilling to assume the risk of offering tuning recommendations.

Budgeting for Azure Sentinel costs

With appropriate attention to cost management, Azure Sentinel is a highly cost-effective SIEM solution and provides substantial benefits over physical and premises-based virtualized solutions. However, as with the move of all IT infrastructure to the cloud, assumptions and modes of operation that held true for the on-premises world need to be re-examined and potentially adjusted.

Azure data ingestion costs can be difficult to project, particularly for data sources that produce logs at variable rates based on factors external to the organization. For example, log data from internet-facing web application firewalls may see a significant spike in volume based on the sudden popularity of a company website, causing Azure ingestion costs to spike in parallel. This may not be an event that the IT organization could have foreseen but will need to be reckoned with for subsequent budget cycles.

Case study–Software company

During deployment of Azure Sentinel, an organization had decided to log inbound denies on an internet-facing firewall that was logging to Azure Sentinel, against recommendations. Shortly after deployment, the organization was targeted by a DDoS attack, causing a large and sudden increase in log volumes. The cost for data ingestion quickly spiked, and adjustments were quickly made to adjust logging levels. While there is value in a record of attacker IP addresses, ingesting large volumes of log data to a cloud service like Azure Sentinel is likely not the most cost-effective method of obtaining this information.

For more on "Economic Denial of Sustainability" attacks, visit:

https://www.managedsentinel.com/2020/10/12/detect_edos_attack_sentinel/

Enumerating in-scope log sources and phasing deployment projects over time

A complete and comprehensive view of all organizational assets is often an unobtainable goal for security teams; however, having a clear view to the in-scope and out-of-scope log sources at the outset of a project will be critical to managing costs for Azure Sentinel over multiple budget cycles. Phasing in the onboarding of new log sources over time is also advised to drive a measured ramp up of log ingestion costs rather than a large spike and inevitable pull-back.

An example of project phasing focusing on users, endpoints, and cloud infrastructure:

Phase	Log sources onboarded
1	M365 log sources Azure log sources
2	Security infrastructure (firewalls, IDS/IPS, NAC)
3	Hybrid infrastructure (servers, L2/L3 devices, storage)
4	SaaS applications

Collecting log samples

Collecting log samples from in-scope log sources is an important and often overlooked step in preparing scope documentation and preparing multi-year budgetary requirements for Azure Sentinel deployments. Analysis of log types, logging behavior, and data volumes for in-scope sources will serve several purposes beyond cost analysis, including baselining and tuning analytic rules that may be set up for detecting anomalous logging behavior or developing log parsing in Log Analytics. Log samples will likely need to be collected from a variety of sources, which should be included in the pre-engagement schedule by the project team.

Common log sources that may be required to sample for analysis include:

- OS security and event logs requiring AMA/MMA
- Syslog or CEF logs from on-premises infrastructure
- Azure Diagnostics logs
- Logs from SaaS applications available via API calls

Ongoing cost monitoring and evaluation

Monitoring costs for log ingestion on an ongoing basis is a critical task in ensuring the ongoing viability of the Azure Sentinel platform for any organization. Azure cost management at a general level is a broad and well-covered topic,[7] so here we will provide more targeted recommendations that can be implemented by security teams within the Azure Sentinel solution.

Using KQL queries

In addition to using KQL queries for analytic and threat hunting rules, they can also be developed to monitor unusual deviations in log ingestion volumes or types and provide alerting for further investigation. Many log source types produce data volumes that follow predictable patterns, such as following business hours or seasonal activity. Creating statistical models and configuring analytic rules to provide notification for deviations from normal log volumes is a simple way to provide immediate notice to administrators or analysts to review spikes in data ingestion before they make a meaningful impact on a monthly Azure invoice.

Trending and visualizations in workbooks or in external tools such as PowerBI can provide an elegant way to spot upward trends in data ingestion by source and allow system owners to take action to keep costs in line with budgets.

Conclusion and Resources

We hope that this whitepaper has been informative and helpful for security practitioners and CISOs pursuing deployments of Azure Sentinel in their own organizations. The contents and recommendations provided have been developed by our team in over 120+ Azure Sentinel deployments around the globe in a variety of industries, as a Microsoft Gold and Threat Protection Advanced Specialization partner.

Many thanks to the teams at Microsoft that have supported us, and the inspiring and forward-thinking customers we have the privilege to work with every day.

Additional resources

Microsoft training paths available at: https://docs.microsoft.com/en-us/learn/

Azure Sentinel in the Azure Marketplace: https://azuremarketplace.microsoft.com/en-us/marketplace/apps?search=azure%20sentinel&page=1

Produced in partnership with BlueVoyant LLC

See More

BlueVoyant®