



# Microsoft Sentinel

Dein Leitfaden zur Einrichtung und Konfiguration

- ▶ Analysen mit SIEM/SOAR
- ▶ Integrationen in bestehende IT-Infrastrukturen vornehmen
- ▶ Automatismen und Anti-Threat Maßnahmen definieren



## Über den Autor

### Aaron Siller

Als ich 2014 als IT-Dienstleister startete, stand ich vor denselben Herausforderungen, mit denen heute viele meiner Kunden zu mir kommen: Komplexe Microsoft-Systeme, ständig neue Security-Anforderungen und nie genug Zeit, um alles richtig zu konfigurieren.

Was als klassische IT-Beratung begann, entwickelte sich schnell zu einer klaren Mission: **Microsoft 365**

**Umgebungen sicherer machen, ohne dass Admins dafür Wochenenden opfern müssen.**



Heute werde ich von führenden Instituten wie der Heise Academy und Golem Karrierewelt als Trainer für Microsoft 365 Security eingesetzt. Meine Expertise bestätigt sich in der Zusammenarbeit mit Unternehmen vom handwerklichen Mittelstand bis hin zu internationalen Konzernen. Schau Dir gerne meine Referenzen auf meiner Website an.

 E-MAIL [aaron@siller.consulting](mailto:aaron@siller.consulting)

 WEBSITE [siller.consulting](http://siller.consulting)

 LINKEDIN [Aaron-Siller](#)

 YOUTUBE [Aaron-Siller-YT](#)

## Inhaltsverzeichnis

Microsoft Sentinel.....	6
Herausforderungen der Security Operations (SecOps).....	6
Was sind SIEM und SOAR? .....	7
Cloud-SIEMs verändern die Welt .....	7
Einführung in Microsoft Sentinel .....	8
Architektur .....	9
Log Analytics .....	9
RBAC.....	11
Datenarchitektur .....	15
Protokolltypen.....	15
Aufbewahrung.....	17
Datennormalisierung .....	18
Content Hub .....	22
Lösungen .....	24
GitHub .....	25
Datenaufnahme .....	25
Art der Datenkonnektoren.....	26
Microsoft First-Party-Konnektoren .....	28
Offizielle Konnektoren von Drittanbietern.....	39
Benutzerdefinierte Konnektoren .....	45
Priorisierung von Konnektoren .....	48
Vorfälle und Analyseregeln .....	49
Vorfälle vs. Warnungen .....	50
Einführung in Analyseregeln .....	51
Geplante Analyseregeln .....	56
Near Real Time (NRT) Regeln .....	71
Microsoft Incident Creation Rules .....	72
Fusion.....	75

SOC-ML.....	77
ML-Verhaltensanalyse.....	80
Wie man mit Analyseregeln beginnt.....	80
Hunting mit Microsoft Sentinel.....	81
Hunting über das Portal .....	82
MITRE ATT&CK-Übersicht.....	87
User Entity Behavior Analytics (UEBA).....	88
Aktivieren von UEBA .....	89
UEBA-Daten.....	90
Entitätsseite .....	91
Aktivitäten.....	92
Watchlists .....	94
Bedrohungsinformationen.....	97
Hinzufügen von Indikatoren.....	99
Verwendung von TI in Analyseregeln.....	103
Verwendung der Microsoft Threat Intelligence (TI).....	103
Daten mit Arbeitsmappen visualisieren.....	104
Einführung in Azure Monitor-Arbeitsmappen .....	105
Verwendung von Arbeitsmappe-Vorlagen.....	107
Erstellen deiner eigenen Arbeitsmappe .....	109
Zugriff auf Arbeitsmappen bereitstellen.....	110
Automatisierung von Antworten .....	111
Playbooks .....	111
Einrichten deiner ersten Automatisierung in Microsoft Sentinel .....	122
Microsoft Sentinel Systemzustand.....	131
Datenkonnektoren .....	133
Automatisierung .....	135
Automatisierung .....	136
API .....	136

PowerShell .....	140
ARM-Vorlagen .....	141
Wahl der richtigen Automatisierungsmethode .....	142
Automatisierte Bereitstellung.....	142
Multi-Tenant-Unterstützung in Microsoft Sentinel.....	146
Workspace Manager .....	149
Reagieren auf Incidents mit Microsoft Sentinel.....	150
Aktualisieren von Incidenteigenschaften.....	151
Ansicht zur Incidentuntersuchung .....	152
Aufgaben .....	154
Entitätsseite .....	156
Untersuchungsgraph.....	159
Automatisierungsregeln.....	161
Incident-Zusammenarbeit mit Microsoft Teams.....	161

# Microsoft Sentinel

## Herausforderungen der Security Operations (SecOps)

Wenn du bis hierher in unserem Buch gekommen bist, wirst du gemerkt haben, dass die Absicherung deiner Microsoft 365-Umgebung keine leichte Aufgabe ist. Es gibt eine Menge verschiedener Produkte und Einstellungen, die es zu beachten und zu konfigurieren gilt. Der Schutz deiner (Cloud-)Umgebung besteht nicht nur darin, proaktiv die richtigen Richtlinien und Einstellungen vorzunehmen, sondern auch darin, auf Ereignisse reagieren zu können, wenn sie eintreten – was meiner Meinung nach genauso wichtig (oder sogar noch wichtiger) ist. Du musst wissen, was in deiner Umgebung vor sich geht. Nur so kannst du deine Umgebung umfassend schützen, indem du potenzielle Sicherheitsverstöße erkennst und darauf reagierst.

Indem du deine Umgebung ständig überwachst, weißt du, was in ihr passiert. Zu viele Unternehmen wissen nicht, was in ihrer Umgebung vor sich geht, wodurch sie anfällig für mögliche Sicherheitsverstöße werden. Es ist von höchster Wichtigkeit, ein Auge auf deine Umgebung zu haben. So kannst du bestimmte Schwachstellen erkennen und deine Sicherheitsrichtlinien aktualisieren, um sie zu beheben. Wenn du dich nur auf die Implementierung der Sicherheitsrichtlinien konzentrierst und nicht darauf, wie sie angegriffen werden, haben Angreifer die Möglichkeit, potenzielle Angriffspunkte zu finden.

Innerhalb von Microsoft 365 gibt es eine Vielzahl von Sicherheitsprodukten, mit denen du Sicherheitsrichtlinien zur Überwachung deiner Umgebung einrichten kannst. Behalte das Microsoft Security Portal für Geräte- und E-Mail-bezogene Verstöße im Auge, bleibe immer auf dem Laufenden über Risikoerkennungen innerhalb von Entra ID Identity Protection und verfolge bösartigen Netzwerkverkehr beispielsweise über ZScaler.

In der heutigen Bedrohungslandschaft ist die Überwachung einzelner Produkte nicht mehr praktikabel. Zum einen wird dich das Hin- und Herwechseln zwischen den verschiedenen Portalen, um auf die Vorfälle jedes Produkts zu reagieren, stark ausbremsen – es ist schlicht ineffizient. Die Automatisierung wird schwer umzusetzen sein, da du mit vielen verschiedenen Produkten interagieren musst. Auf der anderen Seite musst du die Möglichkeit haben, deine Daten produktübergreifend zu korrelieren. Wenn du zum Beispiel eine Warnung über verdächtige Aktivitäten auf einem mit Microsoft Defender gesicherten Gerät erhältst, kann dein sicheres Web-Gateway zusätzliche Einblicke darüber liefern, welche Netzwerkverbindungen ein Gerät hergestellt hat. Durch die Kombination von Daten aus zwei Produkten kannst du Warnmeldungen korrelieren und ein vollständiges Bild über laufende Vorfälle in deiner Umgebung erhalten. Um diese beiden Probleme zu lösen, könnte ein SIEM die Lösung für dich sein.

# Was sind SIEM und SOAR?

Microsoft Sentinel ist Microsofts SIEM- und SOAR-Lösung. SIEM steht für *Security Information and Event Management*. Es ist ein Produkt, das alle deine Logs und Ereignisse an einem einzigen Ort sammelt. Von diesem zentralen „Repository“ aus kannst du Ereignisse aus verschiedenen Datenquellen wie Firewall-Logs und Event-Viewer-Logs von einem Server korrelieren und diese Korrelation für zusätzliche Erkenntnisse nutzen.

Mit all diesen Daten an einem einzigen Ort kannst du sie analysieren und eigene Warnregeln erstellen, wenn du benachrichtigt werden möchtest, sobald bestimmte Ereignisse oder Kombinationen von Ereignissen in deiner Organisation auftreten.

Eine der größten Herausforderungen eines SIEM ist die Anzahl der Warnungen und Vorfälle, die daraus resultieren können. Wenn du ein riesiges Security-Operations-Team hast, das mit diesen Warnungen umgehen, über sie nachdenken und die entsprechenden Maßnahmen ergreifen kann, mag das kein Problem sein. Aber in der heutigen Welt sind viele Organisationen unterbesetzt oder verfügen nicht einmal über ein Vollzeit-Security-Operations-Team. Selbst wenn sie eines haben, wäre das gedankenlose Auswerten aller Ereignisse Zeitverschwendug. Wenn du alles, was aus deinem SIEM kommt, effizient handhaben willst, ist das Hinzufügen von SOAR-Funktionen ein Muss.

Neben einem SIEM ist Microsoft Sentinel auch eine SOAR-Lösung. SOAR steht für *Security Orchestration, Automated Response* und ermöglicht automatisierte Aktionen auf Ereignisse, die in deinem SIEM stattfinden. Es gibt zwei Arten von Aktionen, die ein SOAR ausführen kann: Eine davon ist die Anreicherung, also das Agieren auf Daten in deiner Umgebung, um zusätzlichen Kontext zu liefern. Ein gutes Beispiel ist die Bereitstellung zusätzlicher Informationen zu IP-Adressen, mit denen sich Endpunkte verbinden – zum Beispiel Informationen über die Region und den Netzwerksitzer einer IP. Solche Informationen helfen sehr dabei, zusätzlichen Kontext zu Protokollen und Warnungen zu liefern und Aktivitäten besser zu interpretieren. Die zweite Art ist die automatisierte Bedrohungsreaktion: Das SOAR führt automatisch bestimmte (behebende) Aktivitäten durch, wenn ein bestimmtes Ereignis oder eine Warnung in der Umgebung ausgelöst wird. Dies könnte beispielsweise bedeuten, dass ein Workflow gestartet wird, wenn ein neuer Vorfall erstellt wurde, oder ein Gerät automatisch isoliert wird, wenn eine Warnung mit hohem Risiko eingeht. Ein SOAR kann ein separates Produkt von deinem SIEM sein, aber die meisten modernen SIEM-Produkte haben eine eingebaute SOAR-Funktion.

## Cloud-SIEMs verändern die Welt

SIEM-Produkte gibt es schon lange und sie sind im Bereich der Cybersicherheit allgemein bekannt. Der Begriff SIEM wurde erstmals 2005 von Gartner eingeführt. Einige der bekanntesten SIEM-Lösungen sind Splunk, ArcSight und IBMs QRadar. Die meisten der ursprünglichen SIEMs wurden als lokale Anwendung entwickelt, bei der alle Daten und die Logik

im eigenen Rechenzentrum einer Organisation gespeichert wurden. Das ergab Sinn, da alle Logs von der lokalen Umgebung generiert wurden und das Datenvolumen im Vergleich zu heute eher gering war. Die Speicherkapazität für diese Logs sicher bereitzustellen, ist eine Sache – aber die Rechenleistung, um sie vollständig zu analysieren, ist eine andere.

Hier kommt ein Cloud-basiertes SIEM ins Spiel. Fast jedes große SIEM-Produkt hat heute eine Cloud-Version, wie zum Beispiel Splunk Cloud und AWS Control Tower. Auch wenn ein Cloud-SIEM nicht immer die richtige Lösung für jede Organisation ist, bieten sie viele Vorteile, wenn sie richtig eingesetzt werden. Ein offensichtlicher Vorteil ist, dass eine Organisation nicht selbst für die Speicherung und Rechenleistung zur Verarbeitung all dieser Daten sorgen muss. Ein weiterer Vorteil ist, dass die Cloud-Anbieter stark in maschinelles Lernen und KI investiert haben, um Sicherheitsprobleme zu erkennen und zu beheben. Durch die Nutzung der Leistungsfähigkeit der Cloud bindest du das Wissen der großen Unternehmen ein, um zusätzliche Erkenntnisse über deine Umgebung zu gewinnen und so die Effizienz deiner Sicherheitsabläufe zu verbessern.

## Einführung in Microsoft Sentinel

Microsoft hatte lange Zeit keine SIEM-Lösung, was ein fehlendes Puzzleteil im Sicherheitsangebot von Microsoft Azure war. Erst mit der Übernahme des Unternehmens Hexadite begann Microsoft, in ein SIEM-Produkt zu investieren. 2019 wurde die öffentliche Vorschau von Microsoft Sentinel angekündigt – dem ersten nativen Cloud-SIEM- und SOAR-Produkt von Microsoft. Das Produkt wurde im September 2019 allgemein verfügbar und hat in den letzten Jahren einen enormen Popularitätsschub erlebt. Seitdem hat sich Microsoft Sentinel von einem kleinen SIEM/SOAR-Produkt zu einem vollwertigen SIEM entwickelt, das es mit anderen Marktführern wie Splunk und QRadar aufnehmen kann.

Microsoft Sentinel baut auf bestehenden Azure-Ressourcen wie Log Analytics und Logic Apps auf, um ein natives SIEM-Erlebnis in der Cloud zu bieten. Auch wenn Microsoft Sentinel ein relativ neues Produkt ist, gehört es zu den Microsoft-Security-Produkten mit den häufigsten Updates und Verbesserungen.

Ein wichtiges Missverständnis möchte ich zum Schluss noch ausräumen: Microsoft Sentinel ist kein reines Cloud-SIEM-Produkt. Auch wenn es eine Ressource innerhalb der Microsoft-Azure-Cloud ist, bedeutet das nicht, dass die Systeme und Software, die du überwachst, ebenfalls in derselben Microsoft-Cloud laufen müssen. Microsoft Sentinel unterstützt die Datenerfassung aus einer Vielzahl verschiedener Datenquellen – darunter lokale Systeme ebenso wie Cloud-Anbieter wie Amazon AWS und Google GCP.

## Architektur

Microsoft Sentinel baut auf bestehenden Azure-Ressourcen auf und verwendet viele verschiedene Komponenten, bei denen Sentinel gewissermaßen als Klebstoff fungiert. Einen Überblick über die allgemeine Microsoft Sentinel-Architektur findest du in Abbildung 13. Auch wenn das auf den ersten Blick überwältigend wirken mag, gehen wir in diesem Abschnitt jede einzelne Komponente durch.

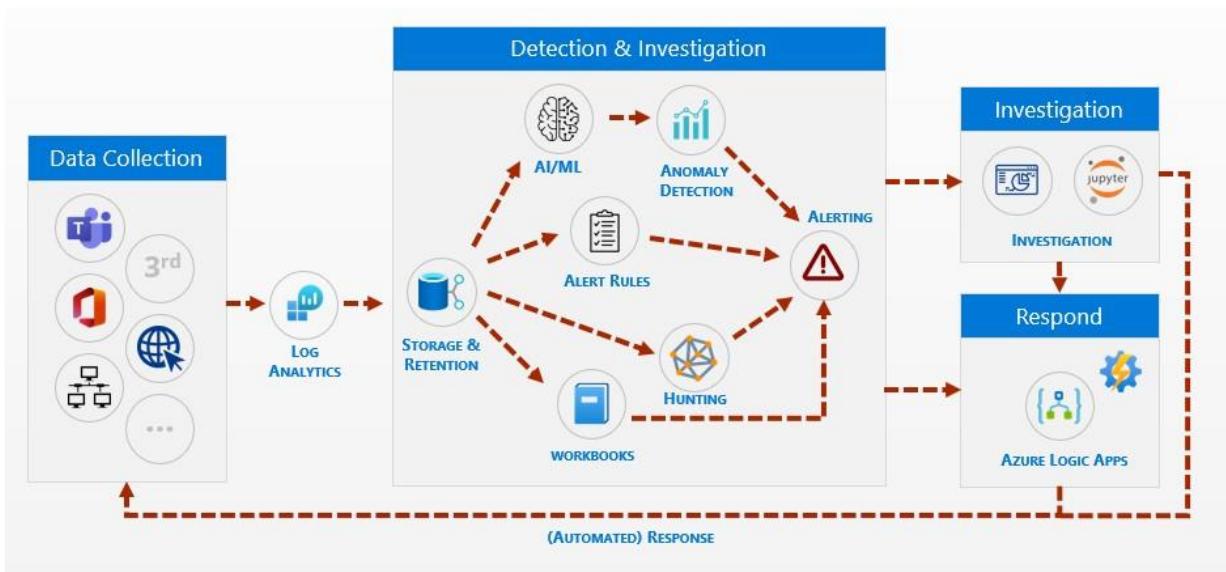


Abbildung 13-1: Übersicht über Microsoft Sentinel

## Log Analytics

Die zentrale Komponente von Microsoft Sentinel ist Log Analytics. Dabei handelt es sich um einen Azure-Dienst, mit dem du Abfragen für Daten erstellen und ausführen kannst, die aus einer oder mehreren Datenquellen gesammelt wurden. Log Analytics ist Teil der Azure Monitor-Produktreihe und verantwortlich für das Speichern der eingehenden Daten sowie das Durchführen von Abfragen. Eine Log Analytics-Umgebung wird als Arbeitsbereich bezeichnet – ein eindeutiger Bereich, in dem Protokolldaten zentral gespeichert werden. Log Analytics wird auch nativ von vielen Azure-Ressourcen verwendet, etwa von Application Insights, Entra ID oder Azure Information Protection.

Zusätzlich zu Log Analytics kannst du verschiedene Lösungen hinzufügen, die zusätzliche Funktionen für einen bestehenden Arbeitsbereich bereitstellen. Ein Beispiel dafür ist die AD-Integritätsbewertung. Wenn du diese Lösung einem Arbeitsbereich hinzufügst, wird auf jedem verbundenen Domänencontroller eine ausführbare Datei installiert. Diese Datei sammelt dann Informationen aus dem Active Directory und sendet sie an den Log Analytics-Arbeitsbereich.

Durch Abfragen und Visualisierungen kannst du daraus wertvolle Erkenntnisse über deine AD-Umgebung gewinnen.

Auch Microsoft Sentinel selbst ist eine Log Analytics-Lösung, die du auf einem vorhandenen Arbeitsbereich aktivierst. Wenn du einen neuen Microsoft Sentinel-Arbeitsbereich einrichten möchtest, brauchst du also zunächst einen Log Analytics-Arbeitsbereich. Sentinel fügt dann die Lösung **SecurityInsights** hinzu, die alle SIEM- und SOAR-Funktionen enthält. Sentinel ist fest an diesen Arbeitsbereich gebunden – das heißt, Log Analytics und Microsoft Sentinel haben eine 1:1-Beziehung. Eine Sentinel-Instanz kann also immer nur einen einzelnen Arbeitsbereich umfassen.

Bevor du deine Sentinel-Umgebung aufbaust, solltest du ein paar Designfragen klären:

1. Wird es einen oder mehrere Log Analytics-Arbeitsbereiche geben?
2. In welcher Region werde ich Microsoft Sentinel erstellen?
3. Möchtest du einen bestehenden Arbeitsbereich verwenden oder einen neuen erstellen?

Die Antworten auf diese Fragen hängen stark von deinen individuellen Anforderungen ab und definieren deine spätere Sentinel-Architektur. Generell empfehle ich, wenn möglich, mit einem einzigen Arbeitsbereich zu arbeiten – also mit einer Sentinel-Umgebung. Wenn du mehrere Arbeitsbereiche anlegst, musst du auch mehrere Sentinel-Instanzen verwalten. Das führt dazu, dass du Regeln, Datenerfassung, Automatisierung und mehr an mehreren Stellen konfigurieren musst, was zusätzliche Komplexität und mehr Aufwand bedeutet. Dennoch gibt es ein paar triftige Gründe, warum mehrere Arbeitsbereiche für dich sinnvoll sein können:

- **Datenhoheit:** Wenn dein Unternehmen internationale Standorte hat und du verschiedenen Datenschutzgesetze einhalten must (wie z. B. die DSGVO), kann es erforderlich sein, Daten innerhalb bestimmter Regionen zu speichern. Das kann dazu führen, dass du separate Sentinel-Instanzen betreiben must – beispielsweise eine in der EU und eine in den USA.
- **Abrechnung:** Wenn dein Unternehmen aus verschiedenen, unabhängigen Teams besteht, kann es notwendig sein, einzelne Ressourcen getrennt abzurechnen. Zum Beispiel könnten die Firewall-Logs vom Netzwerkteam und die EDR-Logs vom Security-Team separat verwaltet und budgetiert werden. Und beide Ressourcen müssen für das jeweilige Team separat abgerechnet werden. Eigene Arbeitsbereiche machen eine getrennte Abrechnung möglich, da diese auf Ressourcenebene erfolgt.
- **Datenbesitz:** Verschiedene Teams könnten jeweils vollständige Kontrolle über ihre Daten benötigen – und damit auch über zugrunde liegende Sentinel-Infrastruktur.

Behalte im Hinterkopf: Mit mehreren Sentinel-Arbeitsbereichen steigt die Komplexität deiner Umgebung erheblich. Wähle diesen Weg nur, wenn es wirklich notwendig ist. Microsoft stellt einen Entscheidungsbaum zur Verfügung, der dir bei dieser Entscheidung helfen kann – du findest ihn [hier](#).

## RBAC

Da Sentinel auf Log Analytics basiert, teilen sich beide Dienste auch bestimmte RBAC-Rollen (Role-Based Access Control). Um also ein fein abgestimmtes Rollenmodell für Sentinel umzusetzen, brauchst du eine Kombination aus spezifischen Sentinel-Rollen und allgemeinen Azure-Rollen. Microsoft Sentinel stellt aktuell vier integrierte Rollen zur Verfügung, die in Tabelle 13-1 aufgeführt sind. Mit ihnen kannst du den Zugriff gezielt steuern. Zusätzlich gewähren auch die Standardrollen von Azure, wie Besitzer oder Mitwirkender, Zugriff auf das SIEM.

Rollenname	Beschreibung
Microsoft Sentinel Reader	Lesezugriff auf alle Aspekte von Microsoft Sentinel (einschließlich der Möglichkeit, Daten abzufragen).
Microsoft Sentinel Responder	Hat dieselben Berechtigungen wie der Reader, kann zudem Vorfälle verwalten (Kommentare hinzufügen, Besitzer & Status ändern...).
Microsoft Sentinel Contributor	Verwalten der zugrunde liegenden Microsoft Sentinel-Infrastruktur (Analytic Rules, Automatisierung, Workbooks...).
Microsoft Sentinel Automation Contributor	Kann Microsoft Sentinel-Automatisierungsregeln erstellen und verwalten. Nur relevant, wenn Kunden über Lighthouse verwaltet werden. Siehe Abschnitt „Arbeiten mit Microsoft Sentinel als MSSP“ für weitere Informationen.

Tabelle 13-1: Microsoft Sentinel-spezifische RBAC-Rollen

Neben den integrierten Microsoft Sentinel-Rollen gibt es noch ein paar andere (Azure-)Rollen, die du beachten solltest. Diese Rollen gelten für einige der Ressourcen, die Microsoft Sentinel im Hintergrund nutzt. Alle Rollen findest du in Tabelle 13-2 unten.

Rollenname	Beschreibung
Log Analytics Reader	Ermöglicht das Lesen von Log Analytics-Daten, ohne Zugriff auf die Microsoft Sentinel-Komponenten.
Log Analytics Contributor	Ermöglicht das Aktualisieren bestimmter Workspace-Einstellungen (z. B. Lösungen, Aufbewahrungsduer...).
Logic Apps Contributor	Ermöglicht das Erstellen und Verwalten von Logic Apps (die für Microsoft Sentinel-Playbooks verwendet werden).

Tabelle 13-2: RBAC-Rollen, die für Microsoft Sentinel gelten

**Hinweis:** Wie du vielleicht bemerkst hast, hat der Microsoft Sentinel Contributor keine Berechtigungen zum Bereitstellen von Arbeitsmappen und Playbooks, da dies Microsoft Sentinel-Ressourcen sind. Um diese Ressourcen bereitzustellen, sind zusätzliche Rollen erforderlich.

Wenn du bestimmte RBAC-Anforderungen hast, die nicht zu einer dieser Rollen passen, gibt es noch ein paar andere Möglichkeiten:

- **Benutzerdefinierte Rollen.** Wenn du spezifische Anforderungen hast, um granulare Rollen zu erstellen und die Anzahl der Aktionen zu begrenzen, die ein Benutzer in Microsoft Sentinel ausführen kann, kannst du benutzerdefinierte Rollen konfigurieren. Alle Microsoft Sentinel-Aktionen sind in der Kategorie [Microsoft.SecurityInsights](#) aufgeführt. Diese umfasst alle RBAC-Aktionen für Microsoft Sentinel. SecurityInsights bezieht sich auf die Lösung, die auf Log Analytics aktiviert wird, wenn du Microsoft Sentinel aktivierst. Damit kannst du beispielsweise bestimmten Benutzern Berechtigungen zum Erstellen von Analyseregeln zuweisen, ohne dass sie bestehende Regeln bearbeiten oder löschen dürfen.
- **Tabellenebene-RBAC.** Alle oben genannten Rollen gewähren Zugriff auf das gesamte Microsoft Sentinel und erlauben keine granularen Einschränkungen. Wenn du nur Zugriff auf bestimmte Tabellen gewähren möchtest, verwende die Berechtigungen auf Tabellenebene. Damit kannst du zum Beispiel Leseberechtigungen auf Tabellen beschränken. Wichtig ist, dass dies nicht für einzelne benutzerdefinierte Tabellen konfiguriert ist - Berechtigungen gelten entweder für alle benutzerdefinierten Tabellen oder für keine.
- **Ressourcenkontext-RBAC.** Wenn du spezifische Anforderungen hast, bei denen ein Benutzer nur Protokolle einer bestimmten Ressource benötigt (z. B. Protokolle für einen bestimmten virtuellen Computer), kannst du dies über ResourceID-Tabelle steuern. Der

Benutzer erhält dann Zugriff auf genau die Protokolldaten, die mit der angegebenen ResourceID verknüpft sind – ganz ohne Zugriff auf den gesamten Log Analytics-Arbeitsbereich. Wenn du benutzerdefinierte Protokolle in Microsoft Sentinel integrierst, kannst du die ResourceID mitgeben, indem du die API, CEF, Syslog oder Logstash entsprechen konfigurierst.

Wenn du gerade mit Microsoft Sentinel beginnst, kann es schwierig sein, die Kosten abzuschätzen, da sie sich nicht leicht vorhersagen lassen. Um die Kosten richtig einzuschätzen, musst du verstehen, wie sie berechnet werden.

Sobald du Daten in Microsoft Sentinel einspeist, nutzt du gleichzeitig zwei Produkte: Azure Log Analytics (Monitor) zum Speichern der Daten und Microsoft Sentinel zur Auswertung. Seit Juli 2023 gibt es ein neues Preismodell, das diese beiden Kostenpunkte zusammenfasst. Zuvor wurden Log Analytics und Microsoft Sentinel separat auf der Azure-Rechnung ausgewiesen, nun erscheinen sie als eine kombinierte Position mit der Bezeichnung „Microsoft Sentinel“. Das erleichtert dir die Kostenkontrolle. Wenn du einen Arbeitsbereich nutzt, der vor Juli 2023 erstellt wurde, kannst du manuell auf das neue Modell wechseln, indem du in den Microsoft Sentinel-Einstellungen „Zu neuer Preisgestaltung wechseln“ auswählst (siehe Abbildung 13-2).

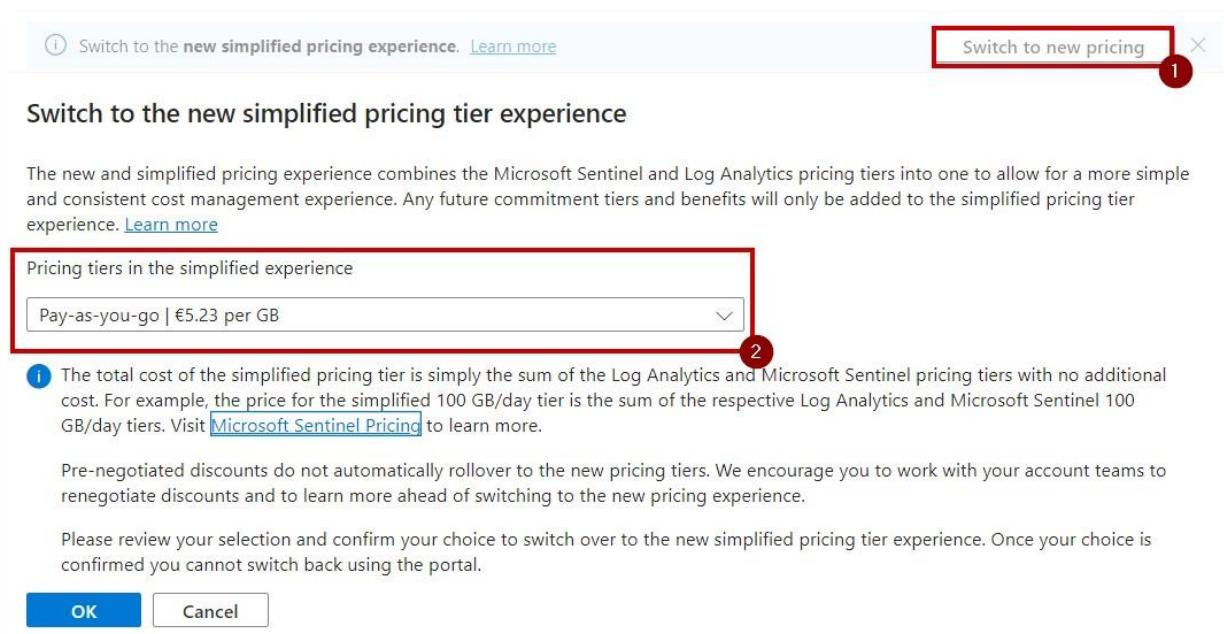


Abbildung 13-2 Wechsel zur neuen Kostenerfahrung

Beim Wechsel solltest du sicherstellen, dass du die Verpflichtungsstufe übernimmst, die du derzeit verwendest – diese wird später im Kapitel noch erklärt. Es gibt keine Nachteile beim Wechsel zum neuen Modell, da sich die Preise nicht verändern.

Es gibt jedoch einige Ausnahmen. Bestimmte Microsoft-First-Party-Ressourcen kannst du ohne zusätzliche Kosten einspeisen, dazu gehören:

- Azure-Aktivitätsprotokolle
- Office 365-Überwachungsprotokolle (Teams, SharePoint und Exchange)
- Warnungen von Defender for Cloud, Microsoft Defender und Identity Protection

Darüber hinaus gibt es einen besonderen Vorteil für Microsoft 365 E5-Kunden: Wenn du Microsoft 365 Defender Advanced Hunting-Protokolle einspeist, erhältst du 100 MB kostenlose Datenaufnahme pro Benutzer und Monat. Dieses Volumen wird dir als Azure-Guthaben gutgeschrieben. Ursprünglich war dieses Angebot zeitlich begrenzt, wurde aber im November 2021 dauerhaft verfügbar gemacht. Die Aktivierung erfolgt automatisch, es gelten jedoch bestimmte Einschränkungen - weitere Informationen findest du auf der [offiziellen Microsoft-Website](#).

Während manche Microsoft-Quellen kostenlos sind, ist die Nutzung von Drittanbieterlösungen kostenpflichtig. Die Microsoft Sentinel-Lösung für SAP ist zum Beispiel mit einer zusätzlichen Gebühr verbunden, die sich nach der Anzahl der angebundenen SAP-Systeme richtet. Das ist eine bedauerliche Entwicklung und ich hoffe persönlich, dass diese Abrechnungsmethode auf SAP beschränkt bleibt. Wenn künftig jede Lösung zusätzliche Gebühren erfordert – zusätzlich zu den regulären Erfassungs- und Aufbewahrungskosten – kann das deine Kostenstruktur erheblich belasten. In meiner Arbeit als Berater habe ich erlebt, dass Kunden diese SAP-Lösung getestet haben, aber oft enttäuscht von den hohen Kosten waren, die mit der Integration verbunden sind.

Die Standardaufbewahrungszeit von Azure Log Analytics beträgt 30 Tage, danach werden die Daten automatisch gelöscht. Wenn du Microsoft Sentinel auf Log Analytics aktiviert hast, erhältst du kostenlos eine Aufbewahrung von bis zu 90 Tagen. Log Analytics unterstützt eine maximale Aufbewahrung von 730 Tagen für Analyseprotokolle und bis zu 7 Jahren für Archivprotokolle. (Die verschiedenen Protokollarten werden im nächsten Abschnitt erklärt.)

**Hinweis:** Die Aufbewahrungszeit ist standardmäßig nicht auf 90 Tage eingestellt. Wenn du deine erste Microsoft Sentinel-Umgebung einrichtest, musst du die Aufbewahrungszeit manuell auf 90 Tage aktualisieren, um von diesem Vorteil zu profitieren.

Wenn du eine große Menge an Daten einspeist, hast du zwei Möglichkeiten:

- Wenn du 100 GB pro Tag oder mehr einspeist, kannst du Verpflichtungsstufen aktivieren. Verpflichtungsstufen sind mit einer festen monatlichen Gebühr verbunden, was es für dich einfacher macht, deine SIEM-Kosten abzuschätzen. Die Aktivierung erfolgt über die Einstellungsseite von Microsoft Sentinel. Es versteht sich von selbst, dass dies nur dann sinnvoll ist, wenn du wirklich große Datenmengen verarbeitest. Wenn du eine Verpflichtungsstufe aktivierst, ohne dieses Datenvolumen zu erreichen, zahlst du für Speicherplatz, den du nicht nutzt. Alle Datenaufnahmen, die über die Zuteilung der von dir erworbenen Stufe hinausgehen, werden zu ermäßigten Sätzen abgerechnet.

- Wenn du ein wirklich intensiver SIEM-Nutzer bist und 1 TB oder mehr pro Tag einspeist, empfiehlt Microsoft [\[dedizierte Cluster\]{.underline}](#). Dedizierte Cluster sind eine Funktion für Azure Monitor, die aber auch für Microsoft Sentinel anwendbar ist, da es im Hintergrund Azure Monitor/Log Analytics verwendet. 1 TB Daten ist natürlich eine riesige Menge und bevor du diese Menge erreichst, solltest du dich wirklich fragen, ob du all diese Daten brauchst.

## Datenarchitektur

Der Kern jedes SIEM-Produkts sind die gesammelten Daten bzw. Protokolle. Microsoft Sentinel verwendet Log Analytics als Grundlage und ermöglicht damit eine Vielzahl von Funktionen – von Archiv- bis Analyseprotokollen.

## Protokolltypen

Log Analytics kennt insgesamt drei Protokolltypen (zwei davon wurden Anfang 2022 hinzugefügt):

- **Analyseprotokolle:** Analyseprotokolle sind die gebräuchlichsten Protokolltypen in Log Analytics. Die meisten Daten in Microsoft Sentinel werden in diesem Format eingespeist, da sie die umfangreichsten Funktionen bieten.
- **Basisprotokolle:** Basisprotokolle sind nur acht Tage lang verfügbar und werden danach entfernt oder ins Archiv verschoben. Sie können nicht in Analyseregeln verwendet werden und unterstützen nur eine begrenzte Anzahl von KQL-Befehlen.
- **Archivprotokolle:** Archivprotokolle dienen der Langzeitspeicherung. Analyse- und Basisprotokolle können nach einer festgelegten Anzahl an Tagen ins Archiv verschoben werden. Archivprotokolle unterstützen keine Echtzeitabfragen über KQL.

Die Einführung dieser Typen hat anfangs für Verwirrung gesorgt, und es ist nicht immer klar, welchen Typ du wann verwenden solltest. Wichtig ist: Analyseprotokolle sind der Standard und in den meisten Fällen die beste Wahl. Sie bieten viele Funktionen und ermöglichen eine native Aufbewahrung von bis zu 730 Tagen – ganz ohne Archivierung.

Basisprotokolle haben nur begrenzte Funktionen und wichtige Einschränkungen:

- Sie können nicht in Analyseregeln verwendet werden.
- Die Aufbewahrung ist auf 8 Tage begrenzt, danach können sie ins Archivübertragen werden.
- Die unterstützen KQL-Operatoren sind stark eingeschränkt.

Du fragst dich vielleicht, warum du Basisprotokolle überhaupt verwenden solltest. Die Antwort ist einfach: Sie sind günstiger – etwa 1 USD pro GB im Vergleich zu 2 USD für Analyseprotokolle.

Das macht sie ideal, wenn du große Mengen von Daten einspielst, die du nur im Hintergrund während einer Untersuchung brauchst.

Ein Beispiel wären rohe Firewall-Protokolle. Sie sind vielleicht nicht sehr nützlich, wenn du bereits Microsoft Defender for Endpoint- und Proxy-Logs einspeist, können aber im Rahmen einer forensischen Analyse dabei helfen, Lücken zu schließen.

Basisprotokolle werden derzeit nur unterstützt, wenn du benutzerdefinierte Datenerfassungsregeln in Verbindung mit dem Azure Monitor-Agenten verwendest, wie [hier](#) dokumentiert. Vorher musst du sie per API-Aufruf aktivieren. Dabei holst du die Konfiguration der Tabelle ab, die du anpassen möchtest, und setzt den Plan auf [basic](#) setzt. Beachte: Die Einspeisung von Basisprotokollen wird nicht auf die Verpflichtungsstufe in Microsoft Sentinel angerechnet.

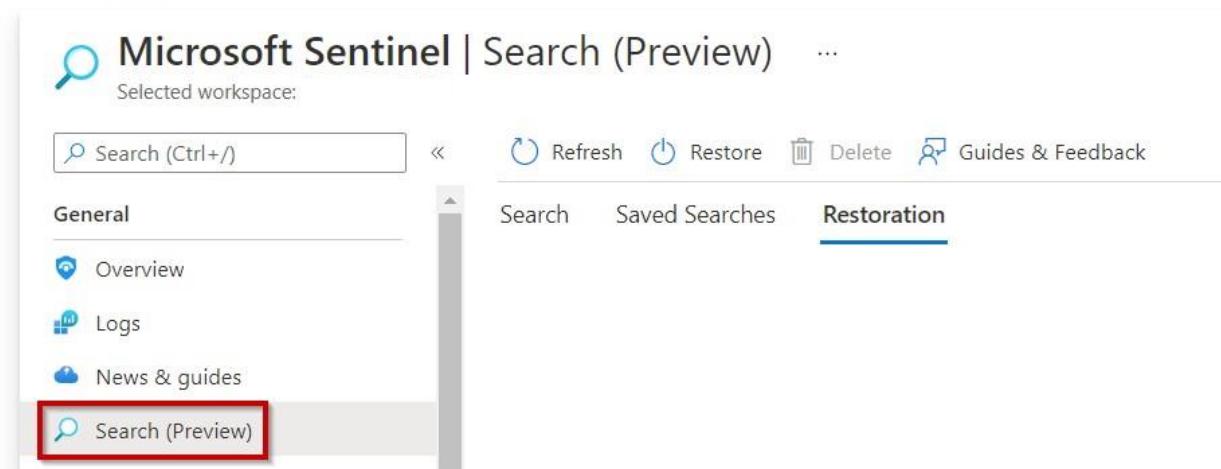
Während Basisprotokolle maximal acht Tage gespeichert werden können, lassen sich Analyseprotokolle bis zu 730 Tage behalten. Danach kannst du sie entweder löschen oder ins Archiv verschieben. Archivierte Protokolle sind deutlich günstiger in der Aufbewahrung und können bis zu sieben Jahre gespeichert werden – im Gegensatz zu zwei Jahren bei Analyseprotokollen. Ideal, wenn du die Daten nicht regelmäßig brauchst, aber im Ernstfall darauf zugreifen möchtest.

Archivprotokolle können standardmäßig nicht direkt abgefragt werden. Es gibt aber zwei Alternativen:

- **Suchjobs** ermöglichen dir die Such nach einem bestimmten Begriff in einer bestimmten Tabelle. Diese Abfragen haben andere technische Einschränkungen als normale KQL-Abfragen, unterstützen große Datenmengen und laufen bis zu 24 Stunden, bevor sie abgebrochen werden. Sie eignen sich perfekt, wenn du gezielt nach etwas suchst.
- Die Datenwiederherstellung erlaubt dir, eine bestimmte Tabelle für einen definierten Zeitraum aus dem Archiv wieder in die Analyseebene zu verschieben. So kannst du aktiv in den Daten suchen und gezielt Informationen extrahieren.

Beide Varianten – Suchjobs und Wiederherstellung – kannst du über die Azure-Management-API oder direkt im Microsoft Sentinel-Portal nutzen.

Im Portal gibt es eine neue Ansicht namens „Suche“, die drei Registerkarten umfasst, wie in Abbildung 13- gezeigt. Darüber kannst du neue Suchen starten, Tabellen wiederherstellen oder vergangene Suchanfragen einsehen.



The screenshot shows the Microsoft Sentinel interface with the title "Microsoft Sentinel | Search (Preview)". A search bar at the top left contains the placeholder "Search (Ctrl+;)". To its right are buttons for Refresh, Restore, Delete, and Guides & Feedback. Below the search bar is a navigation menu with tabs: General (selected), Search, Saved Searches, and Restoration. Under the General tab, there are four items: Overview, Logs, News & guides, and Search (Preview). The "Search (Preview)" item is highlighted with a red box.

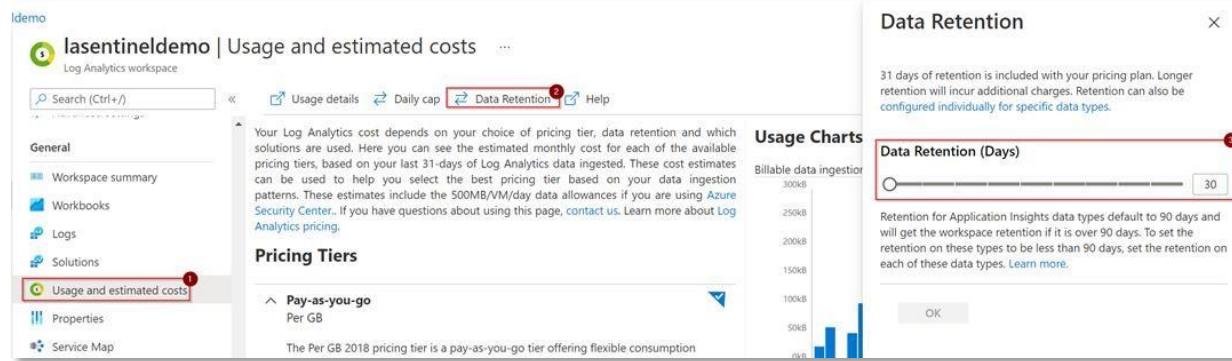
Abbildung 13-3: Die Suchansicht in Microsoft Sentinel

Beachte: Für jede dieser Aktionen entstehen Kosten. Bei Suchjobs zahlst du für die gescannten Daten (GB), bei der Wiederherstellung für die wiederhergestellten Datenmenge und zusätzlich für jeden Tag, den du die wiederhergestellten Daten speicherst.

## Aufbewahrung

Gerade bei Audit- und Sicherheitsprotokollen verlangen viele Organisationen eine längere Aufbewahrung – mitunter mehrere Monate oder Jahre. Die meisten integrierten Microsoft-Sicherheitslösungen speichern Daten nur 30 Tage lang – manchmal bis zu 90 Tage oder ein Jahr mit einer E5-Lizenz. Wenn du darüber hinausgehen möchtest, kannst du die Protokolle nach Microsoft Sentinel exportieren, um sie dort länger aufzubewahren.

Die standardmäßige Aufbewahrung in Log Analytics ist nicht ideal: Sie beträgt 30 Tage. In Verbindung mit Microsoft Sentinel kannst du sie auf bis zu 730 Tage verlängern. Gehe dazu in den Log Analytics-Arbeitsbereich, wähle **Nutzung und geschätzte Kosten > Tägliche Aufbewahrung**, wie in Abbildung 13-4 gezeigt. In der geöffneten Ansicht kannst du die gewünschte Aufbewahrungsduer festlegen. 90 Tage sind dabei ein No-Brainer – für Sentinel-Kunden ist das kostenlos.



The screenshot shows the Azure Log Analytics workspace interface. In the top navigation bar, there are links for 'Search (Ctrl+ /)', 'Usage details', 'Daily cap', 'Data Retention' (which is highlighted with a red box), and 'Help'. On the left, a sidebar lists 'General' (Workspace summary, Workbooks, Logs, Solutions, Usage and estimated costs), 'Properties', and 'Service Map'. The main content area displays 'Pricing Tiers' with a section for 'Pay-as-you-go' per GB. Below this is a chart titled 'Usage Charts' showing billable data ingestion over time. A modal window titled 'Data Retention' is overlaid, stating '31 days of retention is included with your pricing plan. Longer retention will incur additional charges. Retention can also be configured individually for specific data types.' It contains a slider for 'Data Retention (Days)' set to 30, with a note that Application Insights data types default to 90 days. An 'OK' button is at the bottom right of the modal.

Abbildung 13-4: Einstellen der Aufbewahrungszeit für deinen Log Analytics-Arbeitsbereich

Oft wirst du bestimmte Tabellen länger aufbewahren wollen als andere. Du könntest zwar mehrere Sentinel-Instanzen einrichten, aber einfacher ist es, die Aufbewahrung auf Tabellenebene zu konfigurieren. So kannst du abweichende Aufbewahrungsfristen pro Tabelle definieren. Das geht jedoch nicht im Portal, sondern nur per API oder über eine ARM-Vorlage. Eine vollständige Anleitung findest du auf unserem [Blog](#). Aber Achtung: Diese Methode kann schnell unübersichtlich werden, da du auch die Übersicht über die konfigurierten Fristen nur per API erhältst.

**Hinweis:** Die Entscheidung über die passende Aufbewahrungszeit solltest du nicht allein treffen. Kläre diese Anforderungen immer mit dem Management und der Rechtsabteilung ab, um die organisatorischen und gesetzlichen Vorgaben zu erfüllen.

Das Portal bezieht sich bei der Aufbewahrung nur auf Analyseprotokolle. Wenn du diese Fristen verlängern oder weitere Protokolle aufbewahren willst, musst du Archivprotokolle aktivieren. Wie bei der Aufbewahrung auf Tabellenebene funktioniert auch das nur per API – die genaue Anleitung findest du [hier](#).

## Datennormalisierung

Wenn du mit vielen verschiedenen Datenquellen arbeitest, haben diese oft nicht die gleiche Formatierung. Ein Switch von HP hat möglicherweise eine Spalte namens „SourceIP“, während ein Cisco-Switch eine Spalte namens „SourceIPAddress“ hat. Auch wenn beide die gleichen Daten enthalten, können sie auf mehrere Spalten mit unterschiedlichen Namen verteilt sein. Das macht das Abfragen von Daten extrem schwierig, da du viele Variablen wie zum Beispiel die Spaltennamen berücksichtigen musst.

Datennormalisierung ist der Prozess, bei dem alle Daten aus verschiedenen Quellen in ein einheitliches Format gebracht werden. Das erleichtert dir das Schreiben von Abfragen erheblich. Ein SIEM hat zwei Möglichkeiten, Daten zu normalisieren:

- Sicherstellen, dass die Daten vor der Aufnahme normalisiert werden
- Normalisierung zur Abfragezeit

Beide Methoden haben ihre Vor- und Nachteile. Die erste erfordert, dass du die Daten vorab filterst und sicherstellst, dass die richtigen Spalten verwendet werden. Das kann die Datenaufnahme verlangsamen. Die Normalisierung zur Abfragezeit kann hingegen zu Leistungseinbußen führen, da die Ausgabe jedes Mal angepasst werden muss, wenn du eine Abfrage ausführst.

Microsoft Sentinel unterstützt beide Methoden:

- Die Normalisierung vor der Abfragezeit erfolgt durch native Tools oder mit einer benutzerdefinierten Lösung wie Logstash oder eine API.
- Die Normalisierung zur Abfragezeit erfolgt über sogenannte "Parser", die es dir ermöglichen, mehrere Datenquellen mit einer einzigen Abfrage abzufragen – mithilfe des Advanced Security Information Model" oder kurz ASIM.

## Normalisierung bei der Aufnahme

Die fehlende native Unterstützung für die Normalisierung bei der Aufnahme war früher eine große Einschränkung von Microsoft Sentinel – bis ein Update für Log Analytics Anfang 2022 veröffentlicht wurde. Seitdem kann Sentinel Datenerfassungsregeln (Data Collection Rules, DCRs) nutzen, um Daten bereits bei der Aufnahme zu normalisieren. Das bietet dir klare Vorteile:

- Du kannst mit KQL-Abfragen filtern, welche Daten gar nicht erst aufgenommen werden.
- Spalten lassen sich vor der Aufnahme umbenennen oder entfernen.
- Die Zieltabelle kann aktualisiert werden.

Ein häufiger Irrglaube ist, dass die Transformation bei der Aufnahme nur für Quellen funktioniert, die über den Azure Monitor-Agenten angebunden sind. Das ist falsch. Auch für sogenannte Legacy-Workloads kannst du die sogenannte "[Arbeitsbereichstransformation](#)" nutzen. Allerdings unterstützt [nicht jede Tabelle](#) diese Funktion.

## Transformationen

Wenn du eine Datenerfassungsregel erstellst, kannst du Transformationen in Form von KQL-Abfragen definieren. Diese ändern die Daten, bevor sie aufgenommen werden.

Ein besonders interessanter Anwendungsfall: Du kannst unerwünschte Daten vor der Aufnahme entfernen. Das geschieht über einfache KQL-Abfragen. In der gleichen Abfrage kannst du auch bestimmte Spalten entfernen oder umbenennen. Ich selbst nutze diese Filtermöglichkeit häufig,

um die aufgenommenen Datenmengen – und damit die Kosten – gering zu halten und nur die Informationen zu speichern, die ich wirklich brauche.

Leider unterstützen einige Tabellen wie AzureDiagnostics oder jene aus Microsoft 365 Defender diese Filterung nicht. Diese sind oft sehr „laut“, und es wäre wünschenswert, bei der Aufnahme filtern zu können – vielleicht kommt das ja in Zukunft?

## Zieltabelle

Ein weiteres praktisches Feature von Datenerfassungsregeln: Du kannst die Zieltabelle definieren, in die deine Daten geschrieben werden. Viele Tools wie Logstash schreiben Daten in benutzerdefinierte Tabellen. Mit DCRs kannst du stattdessen native Tabellen verwenden. Das hat entscheidende Vorteile:

- Du profitierst von Regelvorlagen von Microsoft und der Community.
- Die Daten können in Machine Learning-Algorithmen wie Fusion einfließen.
- Sie lassen sich in Standardfunktionen wie UEBA (User Entity Behavior Analytics) und Dashboards integrieren.
- Berechtigungen auf Tabellenebene lassen sich nur bei integrierten Tabellen nutzen, nicht bei benutzerdefinierten.

DCRs sind nicht auf eine einzelne Tabelle beschränkt – du kannst auch mehrere Ziele definieren. Ein gutes Beispiel: Firewall-Protokolle. Diese enthalten oft große Datenmengen und verursachen hohe Kosten, wenn du sie vollständig aufnimmst. Mit DCRs kannst du bestimmte Daten als Analyseprotokolle und andere als Basisprotokolle einspeisen. DNS-Ereignisse sind zum Beispiel sehr umfangreich, aber oft nicht entscheidend für eine Analyseregel – im Gegensatz zu IPS-Datenverkehr. Mit DCR kannst du DNS-Ereignisse in eine Basisprotokoll-Tabelle senden, während IPS-Ereignisse in Analyseprotokolle eingehen – ideal für Warnmeldungen. Die DNS-Daten bleiben dennoch verfügbar, falls du sie für spätere Untersuchungen brauchst.

Microsoft Sentinel unterstützt zwei Wege zur Normalisierung bei der Aufnahme:

- Durch Datenerfassungsregeln (DCRs) oder
- Durch benutzerdefinierte Protokollaufnahme individueller Ausgabe.

Details zur Konfiguration folgen in einem späteren Kapitel. Wenn du jetzt mehr erfahren möchtest, kannst du in der [öffentlichen Dokumentation](#) nachlesen.

Neben der offiziellen Unterstützung kannst du weiterhin Logstash oder API-basierte Lösungen nutzen, um normalisierte Daten direkt in Microsoft Sentinel einzuspeisen.

## ASIM

Die Normalisierung zur Abfragezeit entwickelt sich stetig weiter. Die aktuellsten Informationen findest du auf [GitHub von Microsoft Sentinel](#), da dort die neuesten Parser-Versionen veröffentlicht werden. Aktuell gibt es Parser für folgende Bereiche:

- Netzwerksitzung
- DNS-Aktivität
- DHCP-Aktivität
- Prozessereignis
- Authentifizierungsereignis
- Registrierungsereignis
- Dateiaktivität
- Benutzerverwaltung
- Websitzungen
- Auditereignis

Eine Übersicht, welche Protokolle zu welchem Parser gehören, findest du in der [offiziellen Microsoft-Dokumentation](#).

Einige Parser sind standardmäßig in Sentinel enthalten – etwa für DNS, Netzwerksitzungen und Websitzungen. Andere musst du manuell über eine ARM-Vorlage von GitHub bereitstellen. Nach der Bereitstellung kannst du die Parser direkt verwenden. Die Abfragen basieren auf dem Namensschema im\*`<Parser>`\*. Damit kannst du z. B. alle DNS-Ereignisse in deinem Workspace abfragen.

Netzwerksitzungen und Websitzungen, wie [hier](#) dokumentiert. Wenn du zusätzliche Parser bereitstellen möchtest, steht eine einzige ARM-Vorlage [auf GitHub](#) zur Verfügung. Nach Abschluss der Bereitstellung kannst du die verschiedenen Parser verwenden, um die Datennormalisierung innerhalb von Microsoft Sentinel zu erkunden. Der Name eines Parsers verwendet das folgende Format im\*`<Parser>`\*. Das bedeutet, dass du die folgende Abfrage verwenden kannst, um alle DNS-Ereignisse innerhalb deines Arbeitsbereichs abzurufen:

### imDNS

Da die Normalisierung dafür sorgt, dass deine Abfragen unabhängig von der Datenquelle funktionieren, ist die Unterstützung vieler Formate essenziell. Microsoft arbeitet kontinuierlich daran, mehr Produkte zu unterstützen. Du kannst aber auch selbst eigene Parser erstellen. [Durch Aktualisierung der Log Analytics-Funktionen](#) und so sicher stellen, dass auch deine benutzerdefinierten Protokolle einheitlich nutzbar sind.

Sobald du die integrierten Parser bereitgestellt und eigenen Quellen eingebunden hast, kannst du die Normalisierungsschemata direkt in deinen Abfragen nutzen. Viele [integrierter Abfragen](#) unterstützen diese Schemata bereits und es kommen laufend neue hinzu.

Auch wenn die Datennormalisierung oft übersehen wird, ist sie eine zentrale Funktion in Microsoft Sentinel. Sie ermöglicht es dir, neue Datenquellen unkompliziert zu integrieren, ohne ständig Abfragen anpassen zu müssen. Wenn du neu bei Sentinel bist, solltest du dich unbedingt zuerst mit der Normalisierung beschäftigen – so stellst du sicher, dass du deine Arbeit von Anfang an auf einem soliden Fundament aufbaust.

## Content Hub

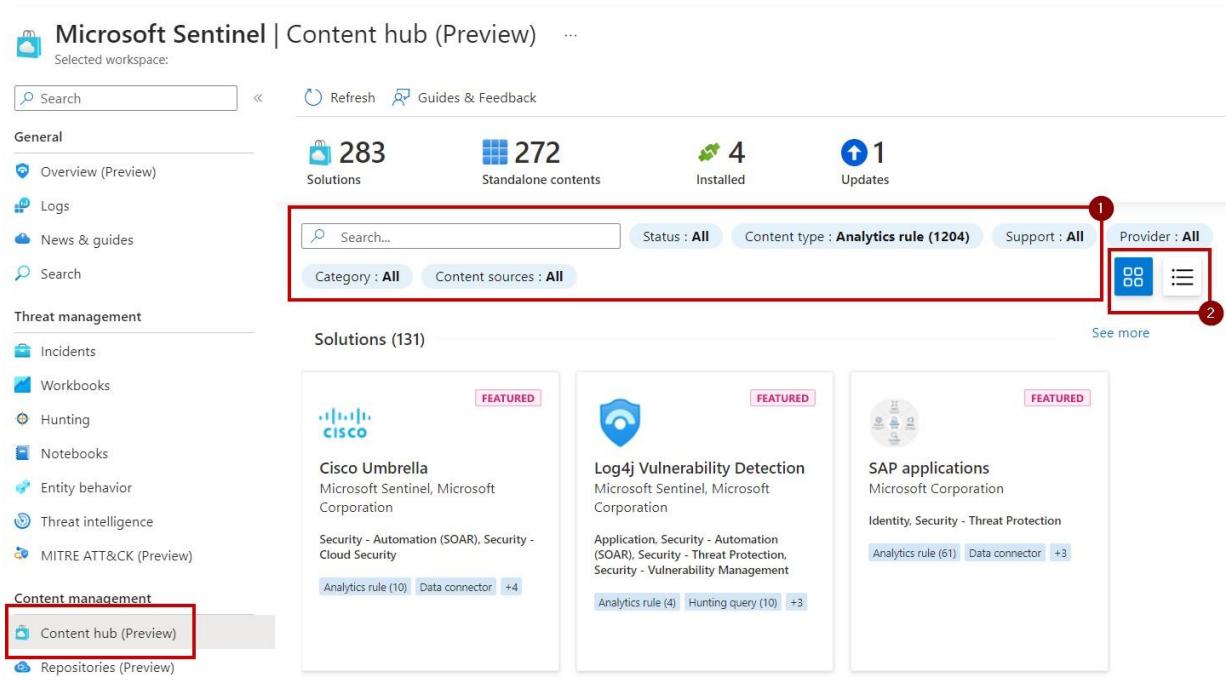
Microsoft Sentinel verfügt über eine Vielzahl von Vorlagen, die du bereitstellen und erkunden kannst. All diese Inhalte sind im sogenannten „Content Hub“ zentral zusammengefasst. Bei der ersten Einführung von Microsoft Sentinel hatte jede Ressource ihre eigene Vorlagenbibliothek – eine Ansicht für Analyseregel-Vorlagen, eine für Datenkonnektoren usw. Das hat sich mit der [OOTB \(Out-of-the-Box\)-Inhaltszentralisierung](#) geändert.

Auch wenn ich Microsofts Gründe für den Wechsel zu einem zentralen Bereich für Vorlagen nachvollziehen kann, habe ich ein paar Probleme mit dem aktuellen Prozess:

- Ich finde es umständlich, zunächst Inhalte installieren zu müssen, um anschließend in den Vorlagen zu suchen und sie manuell zu aktivieren. Das ist ein zusätzlicher Schritt, den es vorher nicht gab.
- Du kannst die KQL-Abfrage einer Analyseregel nicht einsehen, ohne sie vorher zu installieren. Da der Content Hub so umfangreich ist, fällt es mir schwer, das zu finden, was ich brauche. Das wird teilweise durch die Verwendung von Filtern erleichtert.

Im Content Hub findest du sowohl sogenannte „Lösungen“ als auch eigenständige Inhalte. Lösungen bündeln mehrere Ressourcen in einem einzigen Paket, was die Bereitstellung deutlich vereinfacht. Eigenständiger Inhalt hingegen ist eine einzelne Ressource, zum Beispiel eine Analyseregel, die du individuell aktivieren kannst.

Den Content Hub erreichst du über das Menü in Microsoft Sentinel auf der linken Seite – wie in Abbildung 13-5 dargestellt.



The screenshot shows the Microsoft Sentinel Content hub interface. On the left, there's a sidebar with navigation links: General (Overview, Logs, News & guides, Search), Threat management (Incidents, Workbooks, Hunting, Notebooks, Entity behavior, Threat intelligence, MITRE ATT&CK (Preview)), and Content management (Content hub (Preview) - highlighted with a red box, and Repositories (Preview)). At the top, there's a search bar and filter buttons for Status: All, Content type: Analytics rule (1204), Support: All, and Provider: All. Below these are summary counts: 283 Solutions, 272 Standalone contents, 4 Installed, and 1 Update. A large red box highlights the search bar and filter area. Another red box highlights the 'Content hub (Preview)' link in the sidebar. To the right, there's a section titled 'Solutions (131)' with three cards: 'Cisco Umbrella' (FEATURED), 'Log4j Vulnerability Detection' (FEATURED), and 'SAP applications'. Each card includes a brief description, provider information, and analytics/hunting query counts.

Abbildung 13-5: Content Hub von Sentinel

Wenn du dich zum Content Hub durchgeklickt hast, wirst du vielleicht von der Fülle an Informationen überrascht. Das kann anfangs recht unübersichtlich wirken, weil so viele Inhalte verfügbar sind, dass die Navigation schwierig ist. Um dir die Suche zu erleichtern, empfehle ich dir folgende Filter:

1. Verwende die integrierten Filter, um den gewünschten Inhaltstyp (Datenkonnektoren oder Analyseregeln) einzuzgrenzen. Nutze zusätzlich die Eigenschaft "Anbieter", um zu sehen, ob der Inhalt von Microsoft oder einem Drittanbieter stammt.
2. Wechsel von der Listen- zur Tabellenansicht. Ich bevorzuge die Tabellenansicht, weil sie mehr Informationen auf einmal anzeigt – ideal wenn du Inhalte vergleichen oder gezielt durchsuchen willst.

**Unterstützung:** Nicht alle Inhalte im Content Hub werden von Microsoft unterstützt. Einige stammen von Drittanbietern oder Community-Mitgliedern unterstützt. Während das bei Analyseregeln oder Hunting Queries kein Problem ist, solltest du bei der Bereitstellung von Datenkonnektoren oder Playbooks von Drittanbietern besonders aufmerksam sein. Es gibt keine klare Richtlinie, welcher Support verfügbar ist. Also pass auf, wenn du eine nicht unterstützte Konfiguration haben möchtest.

## Lösungen

Lösungen in Microsoft Sentinel sind Sammlungen aus Datenkonnektoren, Analyseregeln, Arbeitsmappen und Suchabfragen, die von Microsoft oder Drittanbietern bereitgestellt werden. Sie ermöglichen dir eine einfache Integration externer Produkte in Microsoft Sentinel.

Ein Beispiel ist die von Microsoft bereitgestellte Lösung „Ubiquiti Unifi“. Sie kombiniert folgende Ressourcen:

- Datenkonnektor
- Parser
- Arbeitsmappen
- Analyseregeln
- Suchabfragen

Mit der Bereitstellung dieser Lösung stehen dir all diese Komponenten in deiner Microsoft Sentinel-Umgebung zur Verfügung. Der enthaltene Datenkonnektor erscheint anschließend auf der Registerkarte „Datenkonnektoren“ und kann so konfiguriert werden, dass er Daten von deinen Ubiquiti-Geräten abruft.

Um alle verfügbaren Lösungen einzusehen, wähle im Microsoft Sentinel-Portal den Punkt „Content Hub“.

**Hinweis:** Auch wenn Lösungen eine komfortable Möglichkeit bieten, mehrere Ressourcen auf einmal bereitzustellen, bedeutet das manchmal auch, dass Inhalte nicht dort erscheinen, wo du sie erwartest. Der Datenkonnektor für Unifi wird beispielsweise standardmäßig nicht in der allgemeinen Übersicht der Datenkonnektoren angezeigt. Wenn du also nach einem bestimmten Datenkonnektor suchst, solltest du unbedingt auch die verfügbaren Microsoft Sentinel-Lösungen durchsuchen, bevor du deinen eigenen Konnektor entwickelst.

Neue Lösungen werden regelmäßig hinzugefügt. Zwei besonders interessante Beispiele sind:

- Das **Trainingslabor**: Diese Lösung speist Validierungsdaten in Microsoft Sentinel ein. Ideal wenn du Sentinel testest und eine realitätsnahe Datenumgebung benötigst.
- **Deception Honey Tokens**: Mit dieser Lösung von Microsoft kannst du Honeytokens in Key Vaults erstellen, die eine Warnung auslösen, wenn sie unerlaubt aufgerufen werden. Die Einrichtung ist zwar etwas komplex - sie umfasst eine Arbeitsmappe, Playbooks, eine Funktion und eine App-Registrierung – doch ist es erfreulich zu sehen, dass solche spezialisierten Szenarien im Hub verfügbar sind.

**Hinweis:** Zu Beginn waren Microsoft Sentinel-Lösungen nicht verfügbar, wenn Sentinel in einem CSP-Azure-Abonnement betrieben wurde. Inzwischen wurde das Produkt dahingehend aktualisiert - alle Lösungen stehen jetzt auch für CSP-Kunden zur Verfügung.

## GitHub

Im weiteren Verlauf dieses Kapitels wirst du häufiger auf GitHub-Verweise stoßen. Wenn du im Microsoft Sentinel-Portal zur Registerkarte „Community“ navigierst, wirst du direkt offiziellen [Microsoft Sentinel GitHub Repository](#) weitergeleitet. Dieses Repository ist eine großartige Ressource – sowohl Microsoft als auch Community-Mitglieder stellen dort Inhalte zur Verfügung, etwa:

- Analyseregeln
- Datenkonnektor
- Suchabfragen
- Playbooks
- U.v.m

Ich empfehle dir dringend, das GitHub-Repository regelmäßig zu besuchen. Es enthält zahlreiche Inhalte, die im Content Hub selbst nicht verfügbar sind, und bietet dir die Möglichkeit, deine Sentinel-Implementierung um weitere Funktionen zu erweitern.

Alle Ressourcen aus dem Content Hub werden übrigens ebenfalls auf GitHub gespeichert. Du findest dort also sowohl Duplikate aus dem Portal als auch zusätzliche Inhalte, die exklusiv über GitHub bereitgestellt werden. Einige davon – etwa Playbooks oder Arbeitsmappen – enthalten sogar einen „Deploy to Azure“-Button, über den die Ressourcen direkt in deiner Azure-Umgebung bereitgestellt werden können.

Wenn du eigene Inhalte für Microsoft Sentinel erstellt hast, ist das GitHub-Repository auch eine gute Möglichkeit, sie mit der Community zu teilen. Das Sentinel-Team prüft regelmäßig eingehende Pull Requests und fügt qualifizierte Inhalte hinzu – vorausgesetzt, du [befolgst einige Richtlinien](#).

## Datenaufnahme

Nachdem du deinen Microsoft Sentinel-Arbeitsbereich eingerichtet hast, ist der nächste Schritt das Einspeisen von Daten über sogenannte Datenkonnektoren. Sie ermöglichen es dir, Protokolle aus verschiedenen Quellen in Sentinel zu integrieren, um Abfragen zu erstellen, Analyseregeln zu nutzen und Sicherheitsereignisse zu untersuchen.

Bevor wir uns die verschiedenen Datenkonnektoren ansehen, die Microsoft Sentinel bietet, solltest du mit einigen Begriffen vertraut sein, die in diesem Kapitel verwendet werden. Eine Übersicht findest du in Tabelle 13-3.

Begriff	Beschreibung
Syslog	Syslog steht für „System Logging Protocol“ und ist ein Standard zum Protokollieren von Nachrichten. Ein Client (z. B. ein Netzwerkgerät) sendet Logs an einen zentralen Syslog-Server, der sie speichert oder weiterleitet.
CEF	CEF steht für Common Event Format und dient dazu, Syslog-Nachrichten zu parsen. Mit CEF wird sichergestellt, dass Daten bereits an der Quelle normalisiert werden und korrekt am Ziel ankommen. test
MMA	MMA (Microsoft Monitoring Agent) ist eine Anwendung, die Logs von Windows und Linux sammelt und an einen Log-Aggregator (z. B. OMS oder ein Azure Log Analytics Workspace) weiterleitet.
Azure Monitor Agent (AMA)	Zukünftiger Ersatz für den MMA-Agenten, der zusätzliche Funktionen wie granulare Filterung von Windows-Ereignissen unterstützt. Der Agent befindet sich derzeit in der Vorschau und unterstützt nur Azure- (Arc-)Server. test

Tabelle 13-3: Einführung von Begriffen, die für die Datenaufnahme gelten

## Art der Datenkonnektoren

Innerhalb von Microsoft Sentinel unterscheide ich zwischen verschiedenen Arten von Datenkonnektoren:

- Microsoft First-Party-Konnektoren
- Offizielle Drittanbieter-Konnektoren
- Benutzerdefinierte Konnektoren

Microsoft-First-Party-Konnektoren sind Konnektoren, die für Microsoft-Produkte entwickelt und direkt von Microsoft bereitgestellt werden. Dazu gehören sowohl Konnektoren für Cloud-Dienste wie Entra ID und Office 365 als auch für lokale Quellen wie DNS oder Sicherheitsereignisse.

Neben den Microsoft-Konnektoren gibt es auch eine Vielzahl an Drittanbieter-Konnektoren. Ein häufiges Missverständnis ist, dass Microsoft Sentinel nur für Microsoft-Cloud-Produkte geeignet ist – das stimmt so nicht. Zwar lag der Fokus zu Beginn stark auf der Microsoft-Cloud, doch inzwischen hat sich das stark erweitert. Monatlich werden neue Drittanbieter-Konnektoren veröffentlicht, darunter für andere Cloud-Anbieter wie AWS oder Google, aber auch für lokale Systeme wie Aruba ClearPass oder Cisco ASA.

Innerhalb der Drittanbieter-Konnektoren lassen sich verschiedene Kategorien unterscheiden:

- **Integrierte Konnektoren** sind für Produkte gedacht, die einen nativen Konnektor direct im Produkt integriert haben. Du musst in diesem Fall keine zusätzlichen Ressourcen bereitstellen – es genügt das Produkt so zu konfigurieren, dass es seine Daten direkt an Microsoft Sentinel sendet.
- Konnektoren auf Basis von **CEF** oder **Syslog** eignen sich oft für lokale (Netzwerk-)Produkte. Die Datenkonnektor-Seite enthält in diesem Fall eine Dokumentation, wie du sowohl das Drittanbieterprodukt als auch den Syslog-/CEF-Server (Forwarder) korrekt einrichtest.
- Andere Produkte bieten **API-basierte Konnektoren**. Für diese Konnektoren wird in der Regel eine zusätzliche Azure-Ressource benötigt – zum Beispiel eine Logic-App oder eine Azure-Funktion. Diese holt die Daten vom Drittanbieter-Produkt ab und überträgt sie in Microsoft Sentinel.

Auch wenn die Anzahl der offiziellen Datenkonnektoren begrenzt ist, heißt das nicht, dass du keine Daten aus anderen Anwendungen einspeisen kannst. Viele Produkte lassen sich mithilfe benutzerdefinierter Konnektoren integrieren. Ein benutzerdefinierter Konnektor kann ebenfalls auf CEF oder einer API basieren. Der wesentliche Unterschied besteht darin, dass du diesen Konnektor selbst entwickelst und er nicht offiziell von Microsoft unterstützt wird. Fällt ein offizieller Drittanbieter-Konnektor aus, kannst du dich an den Support wenden. Bei einem benutzerdefinierten Konnektor liegt die Verantwortung bei dir – du musst das Problem selbst analysieren und beheben.

Das Microsoft-Sentinel-Team stellt auf einer Übersichtsseite alle verfügbaren Datenkonnektoren auf der [folgenden Seite](#) zur Verfügung. Dort kannst du direkt erkennen, welche Konnektoren out of the box verfügbar sind, welche Implementierungsschritte erforderlich sind und von wem der jeweilige Konnektor bereitgestellt wurde – Microsoft oder der Hersteller des Produkts, mit dem du eine Verbindung herstellen möchtest.

Es versteht sich von selbst, dass du offizielle Konnektoren bevorzugen solltest, bevor du einen eigenen benutzerdefinierten Konnektor entwickelst. Wenn du einen Konnektor findest, der deine gewünschte Datenquelle unterstützt und deinen Anforderungen entspricht, solltest du diesen zuerst ausprobieren.

**Hinweis:** Einige Konnektoren werden ausschließlich über Microsoft Sentinel-Lösungen bereitgestellt. Das bedeutet, dass sie nicht in der allgemeinen Übersicht der Datenkonnektoren auftauchen. Wenn du nach einem bestimmten Konnektor suchst, solltest du daher auch die verfügbaren Lösungen prüfen. Weitere Informationen dazu findest du im Abschnitt über Lösungen am Ende dieses Kapitels.

Während die Liste der Datenkonnektoren ständig wächst (sie hatte früher nur zwei Punkte), sehe ich im Moment eine Haupteinschränkung:

- Wenn du Daten über CEF oder Syslog einspeisen möchtest, musst du dafür eigenen Server dafür einrichten. Die Wartung und Absicherung dieses Servers kann mühsam sein, wenn du nicht über ausreichend Ressourcen verfügst. Ein Vorteil wäre, wenn Microsoft Sentinel eine SaaS-basierte Lösung anbieten würde, bei der Microsoft den Server für dich verwaltet und du nur für die Nutzung bezahlst. Das würde es besonders kleineren Unternehmen erleichtern, CEF/Syslog-Quellen schneller zu integrieren. Das ist aktuell jedoch nicht der Fall.

**Hinweis:** Wenn du Daten in Microsoft Sentinel einspeist, solltest du mit einem Begriff **Erfassungsverzögerung** vertraut sein. Dieser beschreibt die Zeitspanne zwischen dem Ereigniszeitpunkt und dem Zeitpunkt, an dem die Daten in Microsoft Sentinel verfügbar sind. Es ist wichtig, den richtigen Zeitstempel zu verwenden, da du sonst bestimmte Erkennungen verpassen könntest. Einige Tipps und Tricks zum Umgang mit Erfassungsverzögerungen findest du im [Microsoft Sentinel Blog](#).

## Microsoft First-Party-Konnektoren

Da Microsoft Sentinel auf dem bestehenden Azure-Ökosystem aufbaut, lässt es sich nahtlos in Azure- und Microsoft 365-Ressourcen integrieren. Microsoft bietet viele First-Party-Konnektoren, mit denen du Microsoft-(Cloud-)Daten mit wenigen Klicks einspeisen kannst. Eine vollständige Liste findest du in der offiziellen [Dokumentation](#).

Es Ein paar Dinge zu Microsoft-Konnektoren möchte ich dir besonders hervorheben.

### Entra ID

Wie du oben gelesen hast, ist das Verbinden der Entra ID-Protokolle sehr einfach, da es mit wenigen Klicks funktioniert.

**Hinweis:** Zum Zeitpunkt des Schreibens heißt der Konnektor im Microsoft Sentinel-Portal noch "Azure Active Directory". Ich gehe davon aus, dass sich dieser Name in den nächsten Monaten ändern wird.

Innerhalb des Entra ID-Konnektors gibt es sieben verschiedene Datentypen, die in Tabelle 13-4 beschrieben sind.

Datentyp	Beschreibung
Anmeldeprotokolle	Interaktive Anmeldeprotokolle von Benutzern
Prüfprotokolle	Protokolliert Aktivitäten in Entra ID (Änderungen an Benutzern, Gruppen, Conditional-Access-Richtlinien...)
Nicht-interaktive Anmeldeprotokolle	Anmeldeprotokolle von Benutzern ohne direkte Benutzerinteraktion
Service-Principal-Anmeldeprotokolle	Anmeldeprotokolle für Service Principals (Anwendungen, die sich per Client-Secret oder Zertifikat anmelden)
Managed-Identity-Anmeldeprotokolle	Protokolle für verwaltete Identitäten (Azure-Ressourcen mit systemzugewiesener Identität)
Bereitstellungsprotokolle	Protokolle zur Bereitstellung von Benutzern/Gruppen in externen Anwendungen
ADFS-Anmeldeprotokolle	Anmeldeprotokolle von ADFS
Benutzerrisikoereignisse	Ereignisse zu risikobehafteten Benutzern aus Identity Protection
Riskante Benutzer	Von Entra ID Identity Protection als riskant eingestufte Benutzer
Riskante Service Principals	Von Entra ID Identity Protection als riskant eingestufte Service Principals
Service-Principal-Risikoereignisse	Ereignisse zu risikobehafteten Service Principals aus Identity Protection

Tabelle 13-4: Im Entra ID-Konnektor enthaltene Tabellen

Der Entra ID-Datenkonnektor erfordert eine Entra ID P1- oder P2-Lizenz. Wenn du keine solche Lizenz hast, kannst du den Konnektor nicht aktivieren.

**Zusätzliche Analyseregeln:** Die integrierten Analyseregeln konzentrieren sich auf die regulären SignInLogs und schließen oft nicht-interaktive oder Dienstprinzipal-Anmeldungen aus. Wenn du gerade erst beginnst, ist es sinnvoll, eigenen Abfragen zu erstellen, die diese speziellen Datenquellen berücksichtigen. Ein Beispiel für Dienstprinzipale findest du in [einem meiner Blog-Beiträge](#).

## Azure-Aktivität

Mit dem Azure-Aktivitätsdatenkonnektor kannst du das Azure-Aktivitätsprotokoll an Microsoft Sentinel streamen. Da dieses Protokoll abonnementspezifisch ist, musst du sicherstellen, dass du die Protokolle **aller** deiner Abonnements an Microsoft Sentinel weiterleitest – sonst verpasst du möglicherweise Daten aus neuen Abonnements.

Der integrierte Konnektor basiert auf einer Azure Policy, mit der du die Einstellung abonnementspezifisch automatisch für alle deine Abonnements konfigurieren kannst. Eine Azure Policy kann auf mehreren Ebenen gültig sein:

- Verwaltungsgruppe
- Abonnements

Eine Verwaltungsgruppe fasst ein oder mehrere Abonnements zusammen. Standardmäßig sind sie nicht aktiviert, aber es wird empfohlen, sie [einzurichten](#). Du kannst eine Stammverwaltungsgruppe einrichten, die alle deine Abonnements umfasst. Wenn du dann die Azure Policy dieser Gruppe zuweist, stellst du sicher, dass auch alle neu erstellten Abonnements so konfiguriert sind, dass ihre Aktivitätsprotokolle automatisch an Microsoft Sentinel gesendet werden. Auch wenn du den Konnektor auf Abonnementebene aktivieren kannst, empfehle ich dir, ihn auf der Stammverwaltungsgruppe zu konfigurieren, um Datenlücken zu vermeiden.

## Microsoft Defender for Cloud

Wenn du Microsoft Defender for Cloud (früher Azure Defender) verwendest, solltest du ihn mit Microsoft Sentinel verbinden. Dadurch werden Warnungen, die von Defender erstellt wurden, automatisch mit Microsoft Sentinel synchronisiert. Der Konnektor unterstützt auch eine bidirektionale Synchronisierung – Warnungen, die in Sentinel geschlossen werden, gelten dann auch in Defender als geschlossen und umgekehrt.

Die bidirektionale Synchronisierung kannst du auf Abonnementebene konfigurieren, indem du unter **Datenkonnektoren** den Konnektor **Microsoft Defender for Cloud** auswählst. In der Registerkarte **Konfiguration** aktivierst du die Warnungssynchronisierung in der Statusspalte und schaltest die bidirektionale Synchronisierung ein.

Connect Microsoft Defender for Cloud to Azure Sentinel  
Mark the check box of each Microsoft Defender for Cloud subscription whose alerts you want to import into Azure Sentinel, then select **Connect** above the list.

The connector can be enabled only on subscriptions that have at least one Microsoft Defender for Cloud plan enabled on Azure Security Center, and only by users with Security Reader permissions on the subscription.

Connect | Disconnect | Enable bi-directional sync | Disable bi-directional sync | Enable Microsoft Defender for all subscriptions >

Search

Subscription ↑↓	Status	Bi-directional sync	Microsoft Defender plans
<input checked="" type="checkbox"/> Azure subscription 1	Connected	Enabled	None enabled <a href="#">Enable all &gt;</a>
<input checked="" type="checkbox"/> Visual Studio Enterprise Subscription – MPN	Connected	Enabled	None enabled <a href="#">Enable all &gt;</a>

Abbildung 13-6: Konfigurieren der Warnungssynchronisierung von Microsoft Defender for Cloud

## Microsoft 365 Defender

Mit der Einführung von Microsoft 365 Defender wurde auch ein neuer Konnektor veröffentlicht, der die Integration mit Microsoft Sentinel verbessert. Dieser bietet zwei Vorteile:

1. Bidirektionale Synchronisierung von Vorfällen zwischen Microsoft 365 Defender und Microsoft Sentinel
2. Erfassung der Microsoft 365 Defender Advanced Hunting-Protokolle in Microsoft Sentinel

Die bidirektionale Synchronisierung ist besonders nützlich, weil sie es ermöglicht, Vorfälle in beiden Portalen ohne doppelte Arbeit zu verwalten. Die bisher genutzte „Microsoft Incident Creation Rule“ wird dadurch überflüssig.

**Hinweis:** Die SLA für die Synchronisierung beträgt 10 Minuten. Es kann also bis zu 10 Minuten dauern, bis ein Vorfall in Sentinel erscheint, nachdem er in Defender erstellt wurde. Beachte dabei: Der "**CreatedTime**"-Zeitstempel in Sentinel entspricht dem Erstellungszeitpunkt in Defender – nicht dem Importzeitpunkt in Sentinel. Weitere Informationen dazu findest du auf [meinem Blog](#).

Neben der Synchronisierung von Vorfällen unterstützt dieser Konnektor auch die Erfassung von Advanced Hunting-Protokollen. Dazu gehören alle Protokolle von Microsoft 365 Defender: Microsoft Defender for Endpoint, Microsoft Defender for Identity, Cloud Apps und Alert Evidence. Die Einspeisung dieser Protokolle in Microsoft Sentinel ist praktisch, da du so Ereignisse zwischen Endpunkten, Netzwerkprotokollen und externen Anwendungen korrelieren kannst. Du solltest jedoch beachten, dass die Einspeisung dieser Protokolle – je nach Umgebung – sehr teuer werden kann. Es ist wichtig zu wissen, dass Microsoft Rabatte gewährt, wenn du diese Daten in Microsoft Sentinel einspeist. Dieser Rabatt wird im Abschnitt „Kosten“ dieses Kapitels behandelt.

Im Oktober 2022 wurden auch die Entra ID Identity Protection-Warnungen in Microsoft 365 Defender integriert. Das bedeutet, dass diese Warnungen jetzt auch über den Defender-Konnektor in Microsoft Sentinel eingespeist werden können. Wenn du bisher den separaten

Identity Protection-Konnektor genutzt hast, entstehen dadurch **doppelte Warnungen**. Du solltest dich daher entscheiden, ob du die Identity Protection-Warnungen **entweder** über den separaten Konnektor **oder** über den Defender-Konnektor einspeist. Der Vorteil bei der Integration in Microsoft 365 Defender liegt darin, dass die Warnungen automatisch mit anderen zusammenhängenden Ereignissen zu einem Vorfall korreliert werden. Auch wenn das hilfreich ist, bevorzuge ich weiterhin die Einspeisung über den separaten Konnektor. Der Grund: Diese Warnungen sind oft sehr umfangreich und müssen individuell angepasst werden. Wenn du den Identity Protection-Datenkonnektor separat nutzt, kannst du eigene KQL-Abfragen mit granularen Bedingungen erstellen, um gezielt festzulegen, wann daraus ein Vorfall wird. Wie du die Integration in Microsoft 365 Defender deaktivierst, erfährst du in Kapitel 12.

## Office 365

Während der Microsoft 365 Defender-Konnektor die Microsoft Defender for Office 365-Warnungen und -Vorfälle enthält, ist das Office 365 Unified Audit Log dort nicht enthalten. Das Office 365 Unified Audit Log wird teilweise über den Office 365-Konnektor eingespeist. Dieser Konnektor enthält drei separate Datensätze:

- Exchange
- SharePoint
- Teams

Die Erfassung dieser Datensätze kann separat aktiviert werden. Du könntest zum Beispiel die Erfassung von Exchange-Daten aktivieren, ohne SharePoint- oder Teams-Daten zu erfassen.

**Hinweis:** Dieser Konnektor enthält nicht das gesamte Office 365 Audit Log. Wenn du das gesamte Audit Log einspeisen möchtest, kannst du dies tun, indem du einen benutzerdefinierten Konnektor über eine Azure-Funktion oder eine Logik-App erstellst. Ein gutes Beispiel findest du auf der [Microsoft Sentinel GitHub Seite](#).

## Sicherheitsereignisse

Es gibt drei Möglichkeiten, Ereignisprotokolle von Windows-Computern in Microsoft Sentinel einzuspeisen:

- Sicherheitsereignisse-Konnektor
- Windows-Sicherheitsereignisse-Konnektor
- Windows-Ereignisprotokolle-Konnektor

Der Sicherheitsereignisse-Datenkonnektor ist vom Microsoft Monitoring Agent abhängig, um Windows-Ereignisse einzuspeisen. Dieser Konnektor unterstützt sowohl Server als auch Clients, unabhängig von ihrem Standort (in Azure oder lokal gehostet). Der Hauptnachteil dieses

Konnektors ist, dass es nur eine begrenzte Anzahl von Konfigurationsmöglichkeiten gibt. Die Auswahl der einzuspeisenden Daten erfolgt durch Auswahl eines der folgenden Werte:

- Alle Ereignisse
- Allgemein
- Minimal
- Keine

Eine Übersicht über die spezifischen Ereignis-IDs, die mit jedem Wert eingespeist werden, findest du [hier](#).

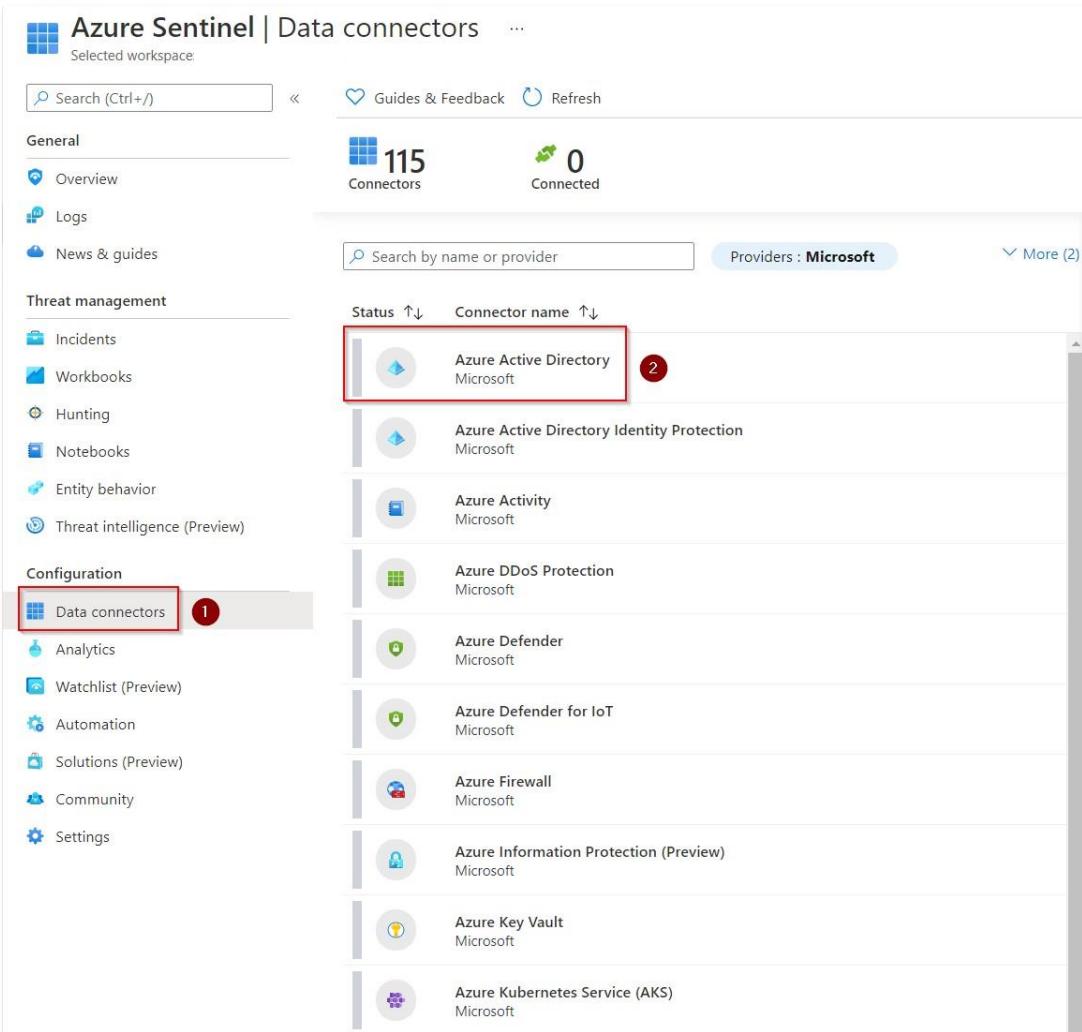
Der Windows-Sicherheitsereignisse-Konnektor basiert auf dem Azure Monitor-Agenten, der es dir ermöglicht, benutzerdefinierte XPath-Abfragen zu erstellen. XPath-Abfragen sind Abfragen, mit denen du angeben kannst, welche Art von Ereignissen erfasst werden sollen. Damit kannst du nach bestimmten Ereignis-IDs filtern und sicherstellen, dass du nur die benötigten Daten einspeist. Der Azure Monitor-Agent wird nur für Server unterstützt, die über Azure Arc verwaltet werden. Es gibt keine Unterstützung für lokale Clients.

**Hinweis:** Wenn du keinen der Agenten verwendest, empfehle ich dir, mit dem Azure Monitor-Agenten (AMA) zu beginnen. Er ist der neueste Agent und wird den MMA-Agenten in Zukunft wahrscheinlich ersetzen. Lokale Server lassen sich einfach in Azure Arc integrieren, was ein kostenloser Dienst ist. Auch wenn Azure Arc erforderlich ist, ist das also kein K.O.-Kriterium.

Die letzte Möglichkeit, Ereignisse in Microsoft Sentinel einzuspeisen, ist die Verwendung der Windows-Ereignisweiterleitung. Während die ersten beiden Lösungen die Installation eines Agenten auf jedem Endpunkt erfordern, der Protokolle senden soll, verwendet die Windows-Ereignisweiterleitung einen zentralen Server zum Sammeln der Protokolle. Endpunkte müssen – zum Beispiel über GPO – so konfiguriert werden, dass sie ihre Protokolle an den Forwarder senden. Dieser Forwarder wird dann mit einem Azure Monitor-Agenten konfiguriert, der die Protokolle an Microsoft Sentinel sendet. Das kann in großen Organisationen oder in Szenarien nützlich sein, in denen bestimmte Endpunkte keine direkte Verbindung zum Internet haben.

## Die ersten Ressourcen mit Microsoft Sentinel verbinden

Wenn du gerade erst mit Microsoft Sentinel anfängst, ist es eine gute Wahl, mit der Aktivierung der Microsoft-Cloud-Konnektoren zu beginnen, da diese eine einfache Konfiguration bieten. Der erste Schritt besteht darin, den Datenkonnektor aus dem Content Hub zu installieren. Navigiere dazu zum Content Hub, suche nach deinem Datenkonnektor (der entweder eigenständig oder Teil einer Lösung sein kann), und klicke auf **Installieren**. Nach der Installation kannst du den Datenkonnektor verbinden, indem du im Microsoft Sentinel-Portal zu **Datenkonnektoren** navigierst und den Konnektor auswählst, den du aktivieren möchtest – wie in Abbildung 13-7 gezeigt.



The screenshot shows the Azure Sentinel Data connectors page. On the left, there's a navigation sidebar with sections: General (Overview, Logs, News & guides), Threat management (Incidents, Workbooks, Hunting, Notebooks, Entity behavior, Threat intelligence (Preview)), Configuration (Data connectors, Analytics, Watchlist (Preview), Automation, Solutions (Preview), Community, Settings). The 'Data connectors' item is highlighted with a red box and a red number '1'. In the main area, there are two summary statistics: 115 Connectors and 0 Connected. Below these are search and filter controls: 'Search by name or provider' and 'Providers : Microsoft' (with a dropdown for 'More (2)'). A table lists ten Microsoft connectors, each with a status icon, name, and provider. The first item, 'Azure Active Directory Microsoft', is highlighted with a red box and a red number '2'.

Status ↑	Connector name ↑
	Azure Active Directory Microsoft
	Azure Active Directory Identity Protection Microsoft
	Azure Activity Microsoft
	Azure DDoS Protection Microsoft
	Azure Defender Microsoft
	Azure Defender for IoT Microsoft
	Azure Firewall Microsoft
	Azure Information Protection (Preview) Microsoft
	Azure Key Vault Microsoft
	Azure Kubernetes Service (AKS) Microsoft

Abbildung 13-7: Microsoft Sentinel-Datenkonnektoren

Nachdem du den gewünschten Datenkonnektor ausgewählt hast, wird auf der rechten Seite deines Bildschirms eine Detailseite eingeblendet (wie in Abbildung 13-8 gezeigt). Hier findest du Details zum Konnektor, zum Beispiel:

1. Konnektor-Status
2. Zugehörige Microsoft Sentinel-Inhalte: Dazu gehören Arbeitsmappen, Suchabfragen und Analyseregeln, die als Vorlagen zur Verfügung stehen und von dir aktiviert werden können. Das kann ein großer Vorteil sein, wenn du gerade erst mit Microsoft Sentinel anfängst, denn diese Vorlagen geben dir einen Vorsprung mit den Daten, die du als Beispiel zum Erstellen deiner eigenen Logik verwenden kannst.
3. Empfangene Daten. Detaillierter Status zu den Daten, die der Konnektor einspeist.
4. Datentypen. Hier werden die Tabellen angezeigt, in die Daten eingespeist werden.
5. Wenn du diesen Konnektor verbinden möchtest, klicke unten auf dem Bildschirm auf die Schaltfläche **Konnektor öffnen**.

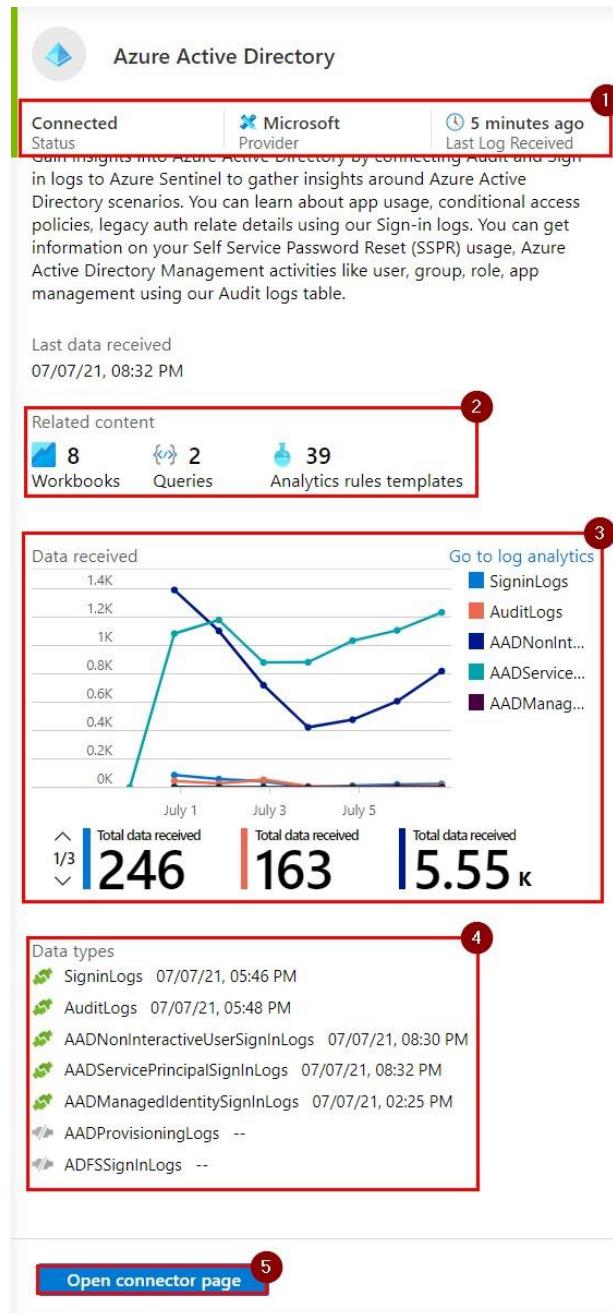


Abbildung 13-8: Details eines Datenkonnektors

Wenn du die Detailseite des Konnektors geöffnet hast, siehst du die spezifischen Voraussetzungen und Konfigurationen dieses Datenkonnektors. Die Voraussetzungen zeigen einige der Dinge, die erforderlich sind, bevor dieser Konnektor aktiviert werden kann, und hängen stark davon ab, welchen Konnektor du zu aktivieren versuchst. Zum Beispiel:

1. **Arbeitsbereichsberechtigungen:** Dies ist für jeden Konnektor erforderlich, da nur Administratoren mit Schreibberechtigungen (wie Microsoft Sentinel Contributor) Datenkonnektoren aktivieren können.
2. **Mandantenberechtigungen:** Diese gelten speziell für Microsoft-Cloud-Konnektoren. Neben den Berechtigungen für das Ziel (Arbeitsbereich) sollte der Administrator, der den Konnektor aktiviert, auch die notwendigen Berechtigungen für das Quellsystem haben. Bei Microsoft 365-Produkten (wie Entra ID, Microsoft Defender, MDCA) bedeutet das eine Entra ID-Rolle wie Globaler Administrator oder Sicherheitsadministrator. Beim Verbinden von Azure-Ressourcen sind Schreibberechtigungen für die jeweiligen Azure-Ressourcen erforderlich.
3. **API/Authentifizierungsschlüssel:** Wird zur Authentifizierung bei Produkten von Drittanbietern verwendet.

Neben den Voraussetzungen zeigt die Konnektorseite auch die Schritte an, die zur Aktivierung dieses Konnektors erforderlich sind. Diese findest du in der Ansicht **Konfiguration**. Im Beispiel des Entra ID-Konnektors, das in Abbildung 13-9 dargestellt ist, kannst du wählen, welche Arten von Protokollen in Microsoft Sentinel eingespeist werden.

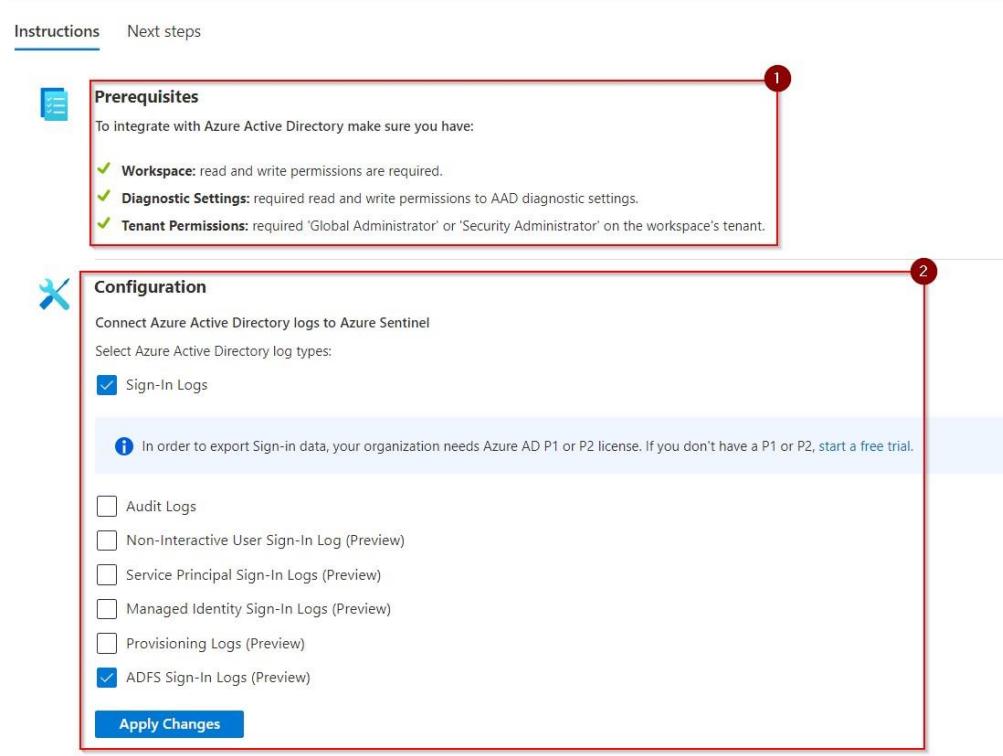


Abbildung 13-9: Konfigurationsansicht eines Datenkonnektors

Nachdem du die Änderungen übernommen hast, ändert sich der Konnektorstatus in **Verbunden**, und die Daten werden in Microsoft Sentinel angezeigt. Beachte, dass es eine Weile

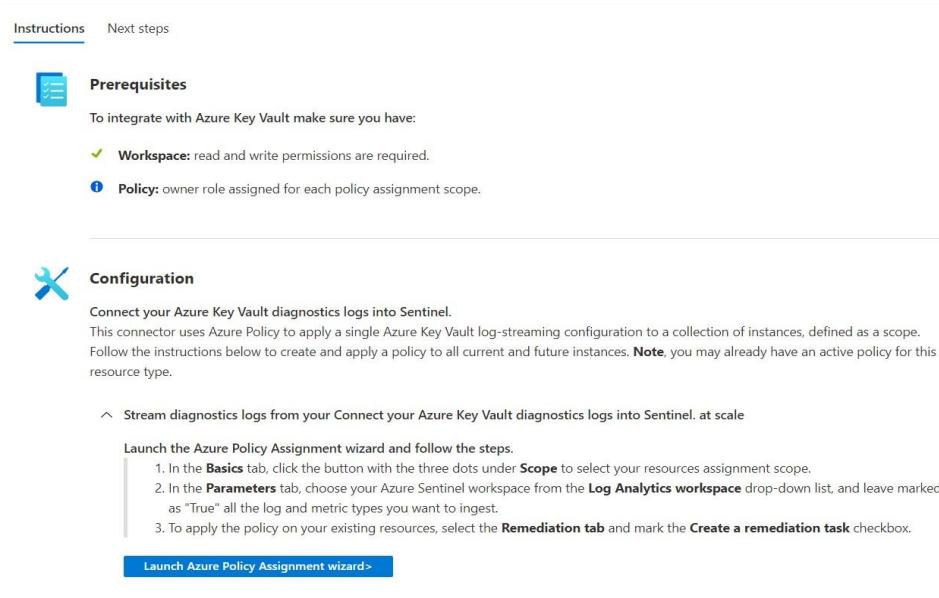
dauern kann (5–30 Minuten), bis die Änderungen im Azure-Portal sichtbar sind und Protokolle in Log Analytics verfügbar werden.

## Azure-Ressourcen verbinden

Die Konfiguration von Microsoft 365-Konnektoren unterscheidet sich von Azure-spezifischen Datenkonnektoren. Es gibt zwei Möglichkeiten, Azure-Ressourcen in Microsoft Sentinel einzuspeisen. Eine besteht darin, die Diagnoseeinstellungen für jede Ressource manuell zu konfigurieren. Das ermöglicht es dir, die Daten gezielt zu testen. Die andere Möglichkeit ist die Verwendung von **Azure Policy**.

Die Standardmethode für die meisten Azure-Ressourcen ist die Aktivierung der Diagnoseeinstellungen über eine Azure Policy. Azure Policy ist ein Azure-Dienst, mit dem du Azure-Ressourcen im großen Stil konfigurieren kannst, indem du Eigenschaften definierst, an die sich eine Ressource halten soll. Durch die Verwendung von Azure Policy ist es bequem, alle gewünschten Azure-Ressourcen innerhalb deines Mandanten an Microsoft Sentinel weiterzuleiten – ohne dass manuelle Aktionen erforderlich sind. So wird auch sichergestellt, dass neu erstellte Ressourcen ihre Protokolle automatisch an Microsoft Sentinel weiterleiten.

Abbildung 13-10 zeigt ein Beispiel für die Funktion von Azure Key Vault. Auf der Datenkonnektor-Seite kannst du auf **Azure Policy-Zuweisungs-Assistent starten** klicken. Dadurch wird eine Azure Policy-Definition mit der erforderlichen Konfiguration (Ressourcentyp, Ziel-Log Analytics-Arbeitsbereich) vorkonfiguriert. Diese Funktion ermöglicht es dir, auf einfache Weise neue Richtlinien zu erstellen, die zur Einspeisung von Daten in Microsoft Sentinel verwendet werden können.



The screenshot shows the 'Prerequisites' section of the Azure Policy Assignment Wizard. It lists requirements: 'Workspace' (read and write permissions required) and 'Policy' (owner role assigned for each policy assignment scope). Below this, the 'Configuration' section is shown, detailing how to connect Azure Key Vault diagnostics logs to Sentinel using Azure Policy. It includes steps to launch the wizard and follow the configuration steps, which involve selecting resources, choosing a Log Analytics workspace, and applying the policy to existing resources. A blue button at the bottom right of the configuration area says 'Launch Azure Policy Assignment wizard>'.

Abbildung 13-10: Verbinden von Azure-Ressourcen mit Azure Policy

Auch wenn in der Konnektorliste viele Azure-Ressourcen enthalten sind, sind nicht alle dort aufgeführt. Das bedeutet jedoch nicht, dass sie nicht in Microsoft Sentinel eingespeist werden können. Nehmen wir als Beispiel einen **API Management-Dienst**. Dieser Dienst wird verwendet, um deine benutzerdefinierten APIs zu skalieren und abzusichern – und er wird viel Traffic verarbeiten.

Um andere Azure-Ressourcen zu integrieren (oder die Verwendung einer Azure Policy zu vermeiden), navigiere zur entsprechenden Ressource im Azure-Portal und wähle **Diagnoseeinstellungen**. Anschließend klickst du auf **Diagnoseeinstellungen hinzufügen**, um die Protokollweiterleitung zu konfigurieren – wie in Abbildung 13-11 dargestellt.

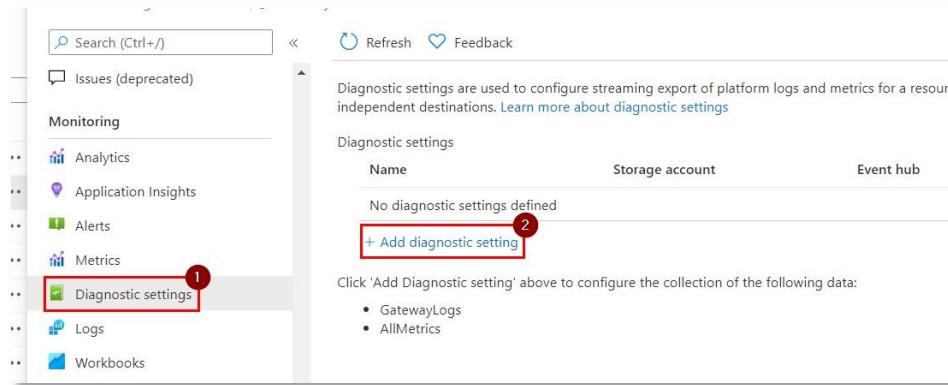


Abbildung 13-11: Manuelle Konfiguration der Diagnoseeinstellungen

Im Bildschirm für die Diagnoseeinstellungen kannst du:

1. Die genauen Protokolle auswählen, die du weiterleiten möchtest (die verfügbaren Optionen unterscheiden sich je nach Azure-Ressource).
2. Das Ziel der Protokolle konfigurieren. Hier solltest du den Log Analytics-Arbeitsbereich auswählen, in dem sich deine Microsoft Sentinel-Lösung befindet.

Nachdem du die Diagnoseeinstellungen konfiguriert hast, werden die Daten in deinen Log Analytics-Arbeitsbereich fließen.

Einer der Gründe, warum Azure-Ressourcen wie diese nicht in der offiziellen Dokumentationsliste enthalten sind, ist, dass Microsoft für die von diesen Ressourcen generierten Daten keine integrierten Arbeitsmappen und Regeln bereitstellt. Wenn du also Microsoft Sentinel-Ressourcen für diese Art von Ressourcen erstellen möchtest, musst du deine eigenen Anwendungsfälle entwickeln.

**Hinweis:** Wenn du Azure Policy für diese Art von Ressourcen verwenden möchtest, solltest du die verfügbaren Azure Policy-Definitionen im Azure-Portal durchsuchen. Wenn sie dort nicht vorhanden sind erstelle eigene. Wenn sie nicht im Microsoft Sentinel-Portal aufgeführt sind, bedeutet das nicht, dass es nicht möglich ist!

## Offizielle Konnektoren von Drittanbietern

Neben Microsoft-Konnektoren gibt es auch viele Drittanbieter-Produkte, die native Konnektoren für Microsoft Sentinel anbieten. Diese Konnektoren wurden vom Microsoft Sentinel-Produktteam validiert und enthalten Parser und Beispielinhalte wie Arbeitsmappen und Suchabfragen.

Die Integration eines Drittanbieter-Konnektors hängt stark vom jeweiligen Anbieter und Produkt ab, lässt sich aber in ein paar Kategorien unterteilen:

- **In das Quellprodukt integriert:** Das bedeutet, dass eine native Integration zwischen dem Quellprodukt und Microsoft Sentinel verfügbar ist. Das Produkt verfügt über eine Option, Protokolle direkt an Microsoft Sentinel zu exportieren.
- **CEF/Syslog.** Viele Anwendungen und Geräte (typischerweise Netzwerkgeräte) unterstützen CEF oder Syslog. Die meisten der aktuellen Drittanbieter-Konnektoren fallen in diese Kategorie.
- **API-basiert.** Cloud-Anwendungen unterstützen oft nicht die Generierung von Protokollen über CEF oder Syslog, bieten aber die Möglichkeit, Protokolle über eine API zu exportieren. In diesem Szenario wird eine API verwendet, um das Quellprodukt nach Daten abzufragen, die dann mithilfe der Data Collector-API von Log Analytics an Microsoft Sentinel gesendet werden.

### Integriert

Integrierte Konnektoren sind am einfachsten und komfortabelsten zu verwenden, da sie keine zusätzlichen Ressourcen wie eine Azure-Funktion oder einen Syslog-Server erfordern. Zum jetzigen Zeitpunkt ist die Unterstützung für integrierte Konnektoren begrenzt, wird aber in Zukunft voraussichtlich wachsen. Ein Beispiel für einen integrierten Konnektor ist die “BETTER Mobile Threat Defense”, mit der du Protokolle wie Warnungen und Gerätprotokolle an Microsoft Sentinel senden kannst. Wie du sehen kannst, ist im Quellprodukt (BETTER MTD) nur eine minimale Konfiguration notwendig, was eine einfache Einbindung in Microsoft Sentinel ermöglicht.



## Configuration

1. In **Better MTD Console**, click on **Integration** on the side bar.
2. Select **Others** tab.
3. Click the **ADD ACCOUNT** button and Select **Azure Sentinel** from the available integrations.
4. Create the Integration:
  - set **ACCOUNT NAME** to a descriptive name that identifies the integration then click **Next**
  - Enter your **WORKSPACE ID** and **PRIMARY KEY** from the fields below, click **Save**
  - Click **Done**
5. Threat Policy setup (Which Incidents should be reported to **Azure Sentinel**):
  - In **Better MTD Console**, click on **Policies** on the side bar
  - Click on the **Edit** button of the Policy that you are using.
  - For each Incident types that you want to be logged go to **Send to Integrations** field and select **Sentinel**
6. For additional information, please refer to our [Documentation](#).

Abbildung 13-12: Beispiel für die Konfiguration eines integrierten Datenkonnektors

## CEF/Syslog

CEF- und Syslog-Konnektoren sind mit Abstand die beliebtesten Arten von Datenkonnektoren. Da diese Protokolle schon lange existieren, unterstützen viele Produkte sie standardmäßig – was es bequem macht, sie auch in Microsoft Sentinel zu nutzen.

Wenn du dir die bestehenden Datenkonnektoren in Microsoft Sentinel ansiehst, wirst du feststellen, dass ein Großteil davon auf CEF/Syslog basiert. Der Unterschied zwischen einem „offiziellen“ und einem regulären CEF/Syslog-Konnektor besteht in folgenden Punkten:

- Datenkonnektoren in Microsoft Sentinel sind von Microsoft validiert.
- Beispieldaten (wie Arbeitsmappen und Abfragen) sind verfügbar.
- Anweisungen zur Einrichtung des Konnektors sind verfügbar.
- Im Falle von Syslog ist ein Parser verfügbar.

**AMA vs MMA:** Bei der Bereitstellung eines CEF- oder Syslog-Konnektors ist es wichtig, die Abkündigung des MMA-Agenten im Auge zu behalten. In vielen Standard-Microsoft-Dokumentationen wird immer noch der alte MMA-Agent referenziert, was aufgrund des Umstiegs auf den AMA-Agenten jedoch nicht mehr empfohlen wird. Die aktuelle Dokumentation ist etwas unübersichtlich, aber die neuesten Anweisungen findest du [hier](#).

## Einrichten des Forwarders

Das Einrichten eines CEF- oder Syslog-basierten Konnektors erfolgt in vier einfachen Schritten:

1. Richte einen Syslog/CEF-Forwarder-Server ein

2. Konfiguriere deinen Forwarder so, dass er Protokolle an Log Analytics (Microsoft Sentinel) sendet 3. Aktiviere die Syslog-Weiterleitung auf deinem Quellprodukt an deinen Forwarder
3. Erstelle eine Datenerfassungsregel.
4. Stelle den Parser bereit

Ein CEF- oder Syslog-basierter Forwarder ist ein Linux-Computer, der Protokolle von deinen verschiedenen (lokalen) Geräten empfängt und an ein Ziel deiner Wahl weiterleitet. Der Linux-Computer kann in einer Azure-VM oder in deinem lokalen Netzwerk gehostet werden – du kannst den Standort frei wählen. Da du den AMA-Agenten verwenden musst, muss dein Forwarder in Azure gehostet werden oder den Azure Arc-Agenten installiert haben. Diese Voraussetzung könnte dich dazu verleiten, deinen Computer in Azure zu hosten, aber ich empfehle Folgendes:

- Wenn du Daten von Cloud-Ressourcen (wie Cloud-basierten Firewalls) sammeln möchtest, empfiehlt es sich, den Forwarder in der Cloud bereitzustellen - insbesondere, wenn du bereits Azure nutzt, da sonst Kosten für die Datenübertragung ins lokale Netzwerk entstehen.
- Wenn du Daten von lokalen Quellen sammelst, ist es am bequemsten, den Forwarder lokal bereitzustellen. So kannst du all deine lokalen Quellen einheitlich konfigurieren und die Verwaltung deiner Firewall -Regeln vereinfachen.

Einige Ausnahmen sind:

- Bei vielen geografisch verteilten Standorten ist es komfortabler, einen zentralen Forwarder in der Cloud zu nutzen.
- Wenn du alle deine Server bereits in der Cloud befinden und nur noch lokale Netzwerkgeräte im Einsatz sind, kann ein Cloud-basierter Forwarder ebenfalls sinnvoll sein – du musst dann keine lokale Infrastruktur mehr pflegen.

**Hinweis:** Wenn du sowohl CEF- als auch Syslog-Quellen hast, kannst du dafür dieselbe Linux-VM verwenden. Eine zusätzliche Konfiguration ist erforderlich, um doppelte Datenübertragung an Microsoft Sentinel zu vermeiden. Anweisungen findest du im [folgenden Artikel](#).

## Konfigurieren des Forwarders

Nachdem der richtige Computer erstellt wurde (entweder lokal oder in der Cloud), kannst du ihn für die Nachrichtenweiterleitung einrichten. Die Einrichtung unterscheidet sich zwischen CEF und Syslog.

Bei der Verwendung von Syslog zur Nachrichtenweiterleitung wird der integrierte Syslog-Daemon genutzt. Log Analytics unterstützt dabei die Daemons **rsyslog** und **syslog-ng**. Falls du ein Linux-Betriebssystem mit einem anderen Daemon verwendest, sollte einer der unterstützten installiert werden. Die genauen Anweisungen zur Einrichtung findest du auf

[Microsoft Learn](#). Diese Schritte lassen sich mit einem Setup-Skript automatisieren, das vom Microsoft Sentinel-Team im [offiziellen GitHub Repository](#) bereitgestellt wird und direkt auf deinen Linux-VMs ausgeführt werden kann.

## Einrichten des Quellprodukts

Nachdem du den Forwarder eingerichtet und konfiguriert hast, ist es an der Zeit, das Quellprodukt einzurichten, für das du Protokolle weiterleiten möchtest. Dadurch kann das Produkt seine Protokolle an den Linux-Computer senden. Diese Protokolle werden anschließend an deinen Microsoft Sentinel-Arbeitsbereich weitergeleitet. Die Einrichtung unterscheidet sich je nach Produkt, und es wird empfohlen, die Dokumentation des jeweiligen Herstellers zur Konfiguration eines Syslog-Ziels zu konsultieren. Beachte, dass in vielen Dokumentationen möglicherweise keine explizite Unterstützung für Microsoft Sentinel erwähnt wird. Solange das Produkt jedoch Syslog- oder CEF-Protokolle irgendwohin senden kann, können diese Protokolle auch in Microsoft Sentinel eingespeist werden.

## Erstellen einer Datenerfassungsregel

Datenerfassungsregeln wurden bereits in einem früheren Abschnitt behandelt, insbesondere im Zusammenhang mit der Transformation bei der Erfassung. Datenerfassungsregeln sind praktische Ressourcen, mit denen du festlegen kannst, welche Daten in Microsoft Sentinel aufgenommen werden und in welche Tabellen sie eingefügt werden. Wenn du eine neue Protokollquelle einrichtest, empfehle ich, zunächst ohne Vorfilterung zu beginnen. Aktiviere die Protokollierung für ein paar Stunden und überprüfe, welche Informationen in Log Analytics eingehen. Danach kannst du die Protokolle mit KQL analysieren und bestimmen, ob es Datenbestandteile gibt, die du entfernen möchtest, um die Erfassungskosten zu reduzieren.

## Bereitstellen des Parsers

Wenn du mit Syslog-Daten arbeitest, landen diese unstrukturiert in der Syslog-Tabelle deines Microsoft Sentinel-Arbeitsbereichs. Zwar ist es möglich, Analyseregeln zu erstellen, um diese Daten innerhalb deiner Abfragen zu parsen, allerdings macht das deine Abfragen schwer lesbar und schwierig zu verwalten. An dieser Stelle kommen Parser ins Spiel. Ein Parser ist eine Log Analytics-Funktion, die Daten abfragt, JSON-Objekte entpackt und bestimmte Spalten umbenennt, damit Abfragen leichter geschrieben und verstanden werden können.

Wenn du einen integrierten Datenkonnektor verwendest, der auf Syslog basiert, findest du im Microsoft Sentinel GitHub-Repository einen passenden Parser. Navigiere dort zum [Parser-Ordner](#) und suche das Produkt, das du gerade konfigurierst.

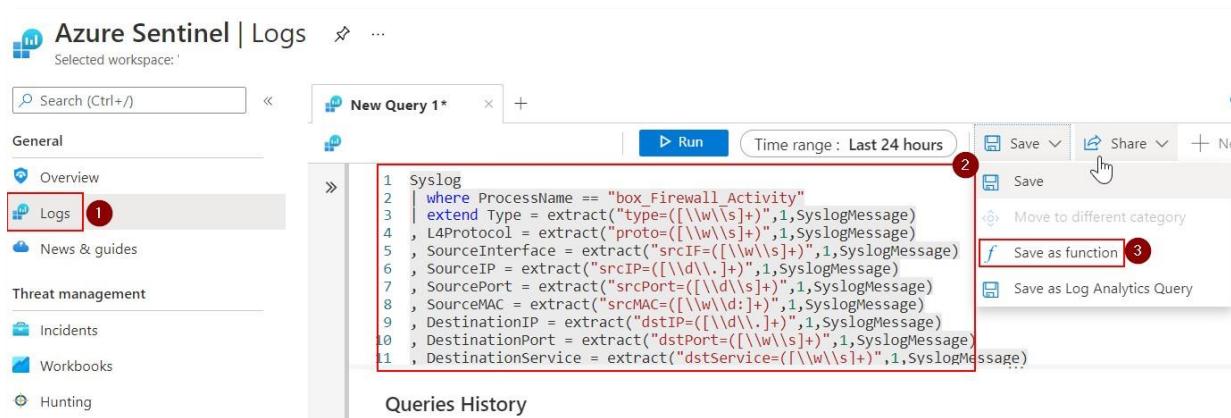
Unten siehst du einen Ausschnitt aus dem Barracuda-Parser, der aus drei Hauptschritten besteht:

1. Er ruft die Syslog-Tabelle auf.
2. Er filtert nur Protokolle mit einem bestimmten ProcessName. Dieser ProcessName ist spezifisch für Barracuda und ermöglicht es dir, die relevanten Protokolle herauszufiltern.
3. Mit dem Operator "extend columns" werden die Rohdaten aus der Spalte "SyslogMessage" extrahiert und in bekannte Spalten überführt, die sich in Microsoft Sentinel besser verwenden lassen.

## Syslog

```
| where ProcessName == "box_Firewall_Activity"
| extend Type = extract("type=(\w+\s+)",1,SyslogMessage)
, L4Protocol = extract("proto=(\w+\s+)",1,SyslogMessage)
, SourceInterface = extract("srcIF=(\w+\s+)",1,SyslogMessage)
```

Nachdem du den Parser aus GitHub abgerufen hast, kopiere die Abfrage in Log Analytics und speichere sie als Funktion – wie in Abbildung 13-13 gezeigt.



The screenshot shows the Azure Sentinel interface under the 'Logs' section. A new query titled 'New Query 1\*' is open. The query code is:

```

1 Syslog
2 | where ProcessName == "box_Firewall_Activity"
3 | extend Type = extract("type=(\w+\s+)",1,SyslogMessage)
4 , L4Protocol = extract("proto=(\w+\s+)",1,SyslogMessage)
5 , SourceInterface = extract("srcIF=(\w+\s+)",1,SyslogMessage)
6 , SourceIP = extract("srcIP=(\d+\.\d+\.\d+\.\d+)",1,SyslogMessage)
7 , SourcePort = extract("srcPort=(\d+)",1,SyslogMessage)
8 , SourceMAC = extract("srcMAC=(\w+\:\w+\:\w+\:\w+\:\w+\:\w+)",1,SyslogMessage)
9 , DestinationIP = extract("dstIP=(\d+\.\d+\.\d+\.\d+)",1,SyslogMessage)
10 , DestinationPort = extract("dstPort=(\w+\s+)",1,SyslogMessage)
11 , DestinationService = extract("dstService=(\w+\s+)",1,SyslogMessage)

```

The top right toolbar has several buttons: 'Save', 'Share', 'Move to different category', 'Save as function' (which is highlighted with a red box and a red number 2), and 'Save as Log Analytics Query' (highlighted with a red box and a red number 3).

Abbildung 13-13: Speichern eines Parsers als Log Analytics-Funktion

Nachdem du auf „Funktion speichern“ geklickt hast, kannst du einen Namen festlegen. Auf der GitHub-Seite wird ein bestimmter Name empfohlen, da die integrierten Abfragen und Arbeitsmappen darauf basieren. Du kannst jedoch auch einen eigenen Namen wählen.

## API-basierte Konnektoren

Auch wenn API-basierte Konnektoren in der SIEM-Welt vielleicht weniger bekannt sind als CEF-/Syslog-Konnektoren, nimmt ihre Nutzung stetig zu. Da immer mehr Anwendungen in die Cloud verlagert werden, unterstützen viele Produkte den Export über CEF oder Syslog nicht mehr. Deshalb müssen alternative Lösungen entwickelt werden.

•

Einige Cloud-Produkte bieten eine native Integration mit Microsoft Sentinel (wie Better MTD), während andere eine API-basierte Integration unterstützen. Es gibt zwei Arten von API-basierten Konnektoren:

- die Verwendung der Codeless Connector Platform (CCP)
- eine Azure-Funktion oder Logik-App, die die Daten einspeist

Die Codeless Connector Platform wurde Anfang 2022 eingeführt und ermöglicht es dir, API-basierte Daten ohne eigenes Hosting einzuspeisen. Du gibst einfach einen API-Endpunkt an, den Microsoft Sentinel regelmäßig abfragt, um Daten zu importieren. Der Hauptvorteil: Du brauchst keine zusätzlichen Ressourcen bereitzustellen.

Zum Zeitpunkt des Schreibens gibt es noch nicht viele integrierte Konnektoren, die CCP verwenden, aber das dürfte sich bald ändern. Beispiele sind JIRA Audit und Slack. Der Autor hat außerdem einen CCP-basierten Konnektor für [\[LastPass Enterprise\]{.underline}](#) veröffentlicht.

Codeless Connectors lassen sich nicht direkt über das Portal erstellen. Du benötigst eine ARM-Vorlage, die alle Konfigurationen definiert – einschließlich Authentifizierung, Abfrage- und Seitenmechanismen. Beim ersten Einrichten wirkt die Vielzahl an Parametern möglicherweise überwältigend. Die Lernkurve ist steil. Meiner Meinung nach richtet sich CCP eher an OEMs, die native Unterstützung für ihre Produkte schaffen, als an SOC-Analysten. So kann eine Organisation Daten einspeisen, ohne die API verstehen oder Infrastruktur bereitzustellen zu müssen.

CCP integriert sich nativ in die Ansicht „Microsoft Sentinel Health“, um Probleme mit dem Abfragemechanismus zu protokollieren. Weitere Informationen dazu findest du ebenfalls in diesem Kapitel.

Die Lösung befindet sich derzeit in der öffentlichen Vorschau und hat ein paar Einschränkungen. Die größte: eingeschränkte Authentifizierungsoptionen und keine grafische Konfigurationsoberfläche. Derzeit werden nur Basic Auth, API-Schlüssel oder Session-Authentifizierung unterstützt. OAuth 2.0 – wie es für Microsoft Graph erforderlich wäre – ist (noch) nicht verfügbar. Die Unterstützung dafür wäre eine wertvolle Ergänzung, etwa zur Erfassung von Microsoft 365-Daten wie Administratorrollen oder benannten Standorten im bedingten Zugriff.

Wenn dein Szenario Funktionen erfordert, die in CCP nicht abgedeckt sind, kannst du weiterhin reguläre API-basierte Konnektoren mit einer Azure-Funktion oder Logik-App nutzen. Meistens wird dabei ein Skript verwendet, das die Daten abruft und in Microsoft Sentinel einspeist. Das Skript kann grundsätzlich überall laufen, wird aber in der Regel in einer Azure-Funktion bereitgestellt – eine einfache, skalierbare, serverlose Lösung.

Ein gutes Beispiel dafür ist der G-Suite-Datenkonnektor. Er ruft Daten aus der Google Reports API ab und überträgt sie in Log Analytics. Beim Bereitstellen eines API-basierten Konnektors hast du zwei Optionen:

1. Bereitstellung über eine vorkonfigurierte ARM-Vorlage
2. Manuelle Einrichtung der Azure-Funktion.

Wenn du gerade erst beginnst, ist die ARM-Vorlage am einfachsten. Die manuelle Einrichtung bietet dir jedoch deutlich mehr Kontrolle, sodass du:

- Azure Key Vault zur sicheren Speicherung von Geheimnissen nutzen kannst
- mehrere Azure-Funktionen in einer App kombinieren kannst (Kosteneinsparungen)
- eine detaillierte Konfiguration der Funktion vornehmen kannst

Je nach Konnektor wird auch hier ein Parser mitgeliefert, den du von GitHub herunterladen und manuell in deinem Log Analytics-Arbeitsbereich speichern musst.

**Geheimnisse und API-Schlüssel:** Standardmäßig verwendet keiner der API-basierten Konnektoren (auf Basis einer Azure-Funktion oder Logik-App) den Azure Key Vault zur Speicherung von Passwörtern und API-Schlüssel. Stattdessen befinden sich diese in der Konfiguration der Funktion. Es wird dringend empfohlen, diese Geheimnisse aus der Konfiguration zu entfernen und im Azure Key Vault zu speichern. Das ermöglicht dir eine fein abgestufte, rollenbasierte Zugriffskontrolle, erleichtert die Schlüsselrotation, erlaubt eine bessere Überwachung - und ist generell eine sicherere im Umgang mit sensiblen Informationen.

## Benutzerdefinierte Konnektoren

Die Anzahl der Datenkonnektoren, die nativ in Microsoft Sentinel verfügbar sind, wächst in rasantem Tempo. Das bedeutet jedoch nicht, dass du keine Daten aus anderen Anwendungen in Microsoft Sentinel einspeisen kannst. Die meisten Anwendungen unterstützen die Datenerfassung in Microsoft Sentinel, wurden nur noch nicht in den offiziellen Katalog aufgenommen.

Es gibt mehrere Möglichkeiten, einen benutzerdefinierten Konnektor zu erstellen. Welche Methode du verwendest, hängt von deinen Anforderungen und der Art der Daten ab, die du einspeisen möchtest:

- CEF/Syslog
- Logstash
- API

## CEF/Syslog

Wenn du CEF- oder Syslog-Quellen einspeisen möchtest, die nicht offiziell unterstützt werden, sind die Schritte meist identisch mit denen für Drittanbieter-Konnektoren. Du kannst denselben Linux-Forwarder für verschiedene Quellen wiederverwenden. Wichtig ist dabei nur, dass du deine eigenen Analyseregeln und Suchabfragen auf Basis der eingespeisten Daten erstellst. Wenn du Syslog-Daten nutzt, empfehle ich, einen eigenen Parser zu erstellen, damit du die Daten effizient verwenden kannst.

## Logstash

Logstash ist ein Open-Source-Produkt von Elastic, mit dem du Daten aus verschiedenen Quellen empfangen, transformieren oder filtern und anschließend an ein SIEM wie Microsoft Sentinel senden kannst. Es ist Teil des [ELK-Stacks](#), der auch Kibana und Elasticsearch umfasst. Während Logstash Vorteile im Zusammenspiel mit Microsoft Sentinel bietet, erlaubt dir der ELK-Stack insgesamt eine erweiterte Nutzung – etwa durch die leistungsstarke Suchmaschine von Elasticsearch oder das Visualisierungstool Kibana. Der Stack ist Open Source, einige Zusatz-Plugins sind jedoch kostenpflichtig. Die Integration mit Microsoft Sentinel ist kostenlos.

**Hinweis:** Im November 2022 wurde eine neue Version des Logstash-Plugins für Microsoft Sentinel veröffentlicht. Sie ersetzt die alte API-Integration und verwendet Datenerfassungsregeln für mehr Granularität. So kannst du Daten direkt in native Tabellen wie die CommonSecurityLog- und SecurityEvent-Tabelen einfügen.

Derzeit gibt es zwei Hauptnachteile bei der Dateneinspeisung in Microsoft Sentinel:

- Nicht alle Konnektoren unterstützen die Normalisierung bei der Erfassung (siehe Abschnitt zur Datennormalisierung).
- Microsoft Sentinel bietet keine generische Funktion zur Normalisierung für der Aufnahme. Dieses Problem kann mit Parsern gelöst werden – diese erhöhen jedoch die Komplexität. Auch die Transformation bei der Erfassung hilft hier nur eingeschränkt, da nicht alle Konnektoren unterstützt werden.

Beide Herausforderungen lassen sich mit Logstash lösen, denn es bietet Funktionen, die in jeder Microsoft Sentinel-Umgebung nützlich sind:

- Du kannst Bedingungen konfigurieren, um Daten ereignisbasiert zu filtern.
- Mithilfe benutzerdefinierter Konfiguration kannst du Spalten anpassen z.B.:
  - Unnötige Spalten entfernen oder Spalten gemäß Namenskonventionen umbenennen
  - Die Daten in aussagekräftige Spalten parsen, um das Parsen zur Abfragezeit zu vermeiden

Ein Vorteil von Logstash gegenüber der Filterung per DCRs ist, dass Logstash die Daten bereits vor dem Versand aus deiner Umgebung filtert. Dadurch vermeidest du große Datenübertragungen, die deine Bandbreite belasten würden. Gerade bei großen Datenmengen ist Logstash empfehlenswert.

Logstash arbeitet eng mit Filebeat zusammen, einem weiteren Elastic-Produkt. Filebeat unterstützt das Einspeisen aus zahlreichen Quellen, insbesondere Cloud-Produkten wie Office 365 und Amazon. Eine Liste der unterstützten Eingänge für [Logstash](#) und [Filebeat](#) findest du in der umfangreichen Elastic-Dokumentation.

Um Logdaten an Microsoft Sentinel zu senden, muss Logstash mit einer Konfigurationsdatei eingerichtet werden, die drei Abschnitte umfasst:

- **Eingänge:** Die Eingabedaten, auf die Logstash achten wird. Das könnten Syslog-Daten oder Daten sein, die über Filebeat importiert wurden.
- **Filter:** Filter sind der interessanteste Teil von Logstash für einen Microsoft Sentinel-Ingenieur. Sie ermöglichen es dir, die Daten zu filtern, die du einspeisen möchtest, oder die Rohdaten (wie Syslog) zu parsen, bevor sie in Microsoft Sentinel eingespeist werden.
- **Ausgänge:** Konfiguriere, wohin Logstash die verarbeiteten Daten senden soll. Logstash hat ein Open-Source-Azure Log Analytics-Modul, das teilweise vom Microsoft Sentinel-Team entwickelt wurde und auch [offiziell unterstützt](#) wird für Azure-Support-Fälle. Das Modul muss auf den Logstash-Servern installiert werden und ermöglicht es dir, Daten an eine Tabelle innerhalb von Log Analytics zu senden.

Auch wenn die Konfiguration anfangs komplex erscheint, bringt sie echten Mehrwert. Logstash schließt Lücken, die Microsoft Sentinel derzeit noch hat. Nutze Logstash in Kombination mit Microsoft Sentinel, um das Beste aus beiden Welten zu verbinden: Open Source trifft auf Microsoft-Integration.

**Syslog/CEF vs. Logstash:** Während die meisten Datenkonnektoren empfehlen, einen Syslog- oder CEF-Forwarder einzurichten, solltest du überlegen, ob Logstash eine Option ist. Es bietet einige Vorteile gegenüber einer Standardintegration und muss von Anfang an in dein Design einbezogen werden. Da immer mehr Datenkonnektoren auf den AMA und die Transformation bei der Erfassung setzen, nimmt der Bedarf an Logstash mittelfristig ab.

## API

Es gibt noch weitere Methoden, um benutzerdefinierte Konnektoren zu erstellen – zum Beispiel über die Codeless Connector Platform oder eigene Azure-Funktionen bzw. Logik-Apps. Die Codeless Connector Platform ist zunächst komplexer in der Einrichtung, bietet aber Vorteile: Du musst keine zusätzlichen Azure-Funktionen oder Logik-Apps verwalten, und die Plattform ist mit Microsoft Sentinel Health integriert, sodass du automatisch benachrichtigt wirst, wenn ein Erfassungsproblem auftritt. Daher empfehle ich diese Methode.

Wenn dein Anwendungsfall die Plattform nicht unterstützt, musst du selbst eine Azure-Funktion oder Logik-App bereitstellen und die APIs dort aufrufen. Alle Varianten basieren auf der „Data Collector API“ (siehe Abschnitt Automatisierung). Diese API funktioniert sowohl lokal als auch in der Cloud. Auf [GitHub von Microsoft Sentinel](#) findest du zahlreiche Beispiele von Microsoft Sentinel.

## Priorisierung von Konnektoren

In jeder Organisation gibt es viele verschiedene Datenquellen, die potenziell für Microsoft Sentinel interessant sein könnten. Es wirkt zunächst sinnvoll, alle diese Quellen sofort einzuspeisen, ich bevorzuge jedoch die Wasserfallmethode: neue Quellen werden nacheinander angebunden, sobald eine neue Microsoft Sentinel-Umgebung eingerichtet wird.

Ich gehe dabei nach folgender Reihenfolge vor:

1. Microsoft-Cloud-Daten
2. Sicherheitsprotokolle
3. SaaS-Produkte
4. Sonstiges

Bei jeder Microsoft Sentinel-Bereitstellung beginne ich damit, alle Microsoft-Cloud-Datenquellen zu aktivieren. Dafür gibt es ein paar Gründe:

- Microsoft Sentinel bietet für diese Quellen eine Menge vorgefertigter Ressourcen - Analyseregeln und Suchabfragen. Wenn du gerade erst mit Microsoft Sentinel anfängst, ist es praktisch, diese Ressourcen zu nutzen und das Produkt und die zugrunde liegenden Daten auf diese Weise kennenzulernen.
- Die Microsoft-Cloud-Datenquellen können mit wenigen Klicks eingerichtet werden und erfordern keine komplexe Einrichtung.
- Viele Microsoft-Cloud-Datenquellen können kostenlos eingespeist werden. Das bedeutet, dass du mit Microsoft Sentinel experimentieren kannst, ohne dir allzu viele Gedanken über die Kosten zu machen.

Der nächste Schritt ist die Einspeisung von Sicherheitsprotokollen. Dabei handelt es sich um Protokolle von Firewalls, Antiviren- oder EDR-Systemen. Diese Protokolle solltest du einspeisen, da sie eine große Menge an Informationen enthalten, die nützlich sind, um deine Umgebung zu überwachen – aber auch, um sie mit Informationen aus einigen der Cloud-Protokolle zu korrelieren. Denk beim Einrichten dieser Art von Konnektoren daran, klein anzufangen. Es kann sinnvoll sein, mit den Administratorprotokollen deiner Switches zu beginnen (da diese zur Erkennung böswilliger Änderungen verwendet werden können), bevor du alle Netflow-Protokolle einspeist – diese enthalten nämlich nicht immer viele verwertbare Daten.

Danach konzentriere ich mich gerne auf wichtige SaaS-Produkte. Ein gutes Beispiel ist LastPass, ein cloudbasierter Passwort-Tresor. Diese Protokolle sind wichtig, da Aktionen in solchen Produkten streng überwacht werden sollten. Ein weiterer Vorteil: SaaS-Produkte unterstützen in den meisten Fällen eine einfache Integration in SIEMs (entweder integriert oder API-basiert), was bedeutet, dass du kein komplexes lokales Setup benötigst, um mit der Datenerfassung zu beginnen.

Im letzten Schritt wähle ich die Einspeisung anderer wichtiger Datenquellen, die nicht in die ersten drei Kategorien passen. Bevor du solche Daten in Microsoft Sentinel eingespeist, denk bitte daran: Das Einspeisen von Daten „zum Spaß“ ist nicht zu empfehlen, da du für jedes Megabyte zahlst, das du speicherst. Stelle sicher, dass du einen triftigen Grund hast, diese Daten einzuspeisen. Zu den Gründen können etwa gehören:

- Einige Protokolle müssen aus rechtlichen oder forensischen Gründen für mehrere Monate **aufbewahrt** werden.
- Du hast spezifische **Erkennungsanforderungen** in bestimmten Ressourcen. Zum Beispiel: Du musst einen Vorfall generieren, wenn Benutzer X eine bestimmte Aktion in einem Produkt ausführt. Oder du möchtest böswillige Aktivitäten erkennen, etwa eine Warnung auslösen, wenn bestimmte (Netzwerk-)Aktivitäten oder Muster auftreten.

## Vorfälle und Analyseregeln

Nachdem du deine ersten Protokolle in Microsoft Sentinel eingespeist hast, kannst du beginnen, Vorfälle zu erstellen, um verdächtige Aktivitäten in deiner Umgebung zu untersuchen. Das Einrichten der richtigen Regeln zur Erstellung von Warnungen und Vorfällen ist ein zentraler Schritt bei der Konfiguration von Microsoft Sentinel.

Wie bei jeder anderen SIEM-Plattform bestimmen die Vorfälle, welche Art von Ereignissen ein SOC-Analyst untersuchen wird. Wenn du zum ersten Mal mit einer SIEM-Plattform arbeitest, kann es herausfordernd sein, die passenden Regeln dafür zu definieren, was ein Vorfall ist – und was nicht. Leider versuchen viele Organisationen, möglichst viele Regeln zu erstellen, um eine große Anzahl von Vorfällen zu generieren. Sie wollen damit sicherstellen, dass keine echten Bedrohungen übersehen werden. Das führt aber häufig zu einem falschen Gefühl von Sicherheit, weil angenommen wird, dass einem SOC-Analysten dadurch nichts entgeht. Das Gegenteil ist der Fall: Wenn du zu viele falsch positive Vorfälle erzeugst, ist ein Analyst gezwungen, eine große Menge an Meldungen zu bearbeiten. Und da kaum eine Sicherheitsabteilung über ausreichende Ressourcen verfügt, wird der Analyst versuchen, die Vorfälle möglichst schnell abzuarbeiten. Mit der Zeit gewöhnt er sich zudem an das hohe Aufkommen an Fehlalarmen. Die Folge: Es ist sehr leicht, echte Bedrohungen zu übersehen. Bei der Vorfalluntersuchung steckt der Teufel im Detail – deshalb ist es wichtig, sich die nötige Zeit zu nehmen, um Warnungen sorgfältig zu prüfen. Es geht darum, relevante Ereignisse, Warnungen und Vorfälle zu analysieren – nicht nur das Rauschen drumherum.

## Vorfälle vs. Warnungen

Bevor du tiefer in die Konfiguration von Microsoft Sentinel einsteigst, solltest du den Unterschied zwischen Ereignissen, Warnungen und Vorfällen verstehen. Diese Begriffe haben bei Microsoft Sentinel klare Definitionen, die sich von anderen Produkten unterscheiden können, mit denen du vielleicht bereits gearbeitet hast.

- **Ereignisse:** Ein Ereignis ist im Wesentlichen eine einzelne Zeile, das Ergebnis einer Abfrage, die später für eine Warnung oder einen Vorfall verantwortlich sein kann. Es beschreibt eine eindeutige Aktion, ist aber nicht zwingend relevant für eine Untersuchung. Im Microsoft Sentinel-Portal wird ein einzelnes Ereignis gar nicht angezeigt.
- **Warnungen:** Eine Warnung basiert auf einem oder mehreren Ereignissen, die das Ergebnis einer bestimmten Abfrage sind. Sie ist bemerkenswert, aber erfordert nicht zwangsläufig sofortiges Eingreifen.
- **Vorfälle:** Ein Vorfall besteht aus einer oder mehreren Warnungen und ist etwas, das von einem Analysten untersucht werden sollte. Vorfälle findet du im Azure-Portal, wo du alle aktuellen und vergangenen Fälle einsehen kannst. Nach der Untersuchung wird ein Vorfall in der Regel als richtig positiv, falsch positiv oder gutartig positiv klassifiziert.

Wie in Abbildung 13-14 dargestellt, gibt es im Vergleich zu den Tausenden von täglichen Ereignissen nur eine kleine Teilmenge richtig positiver Vorfälle.

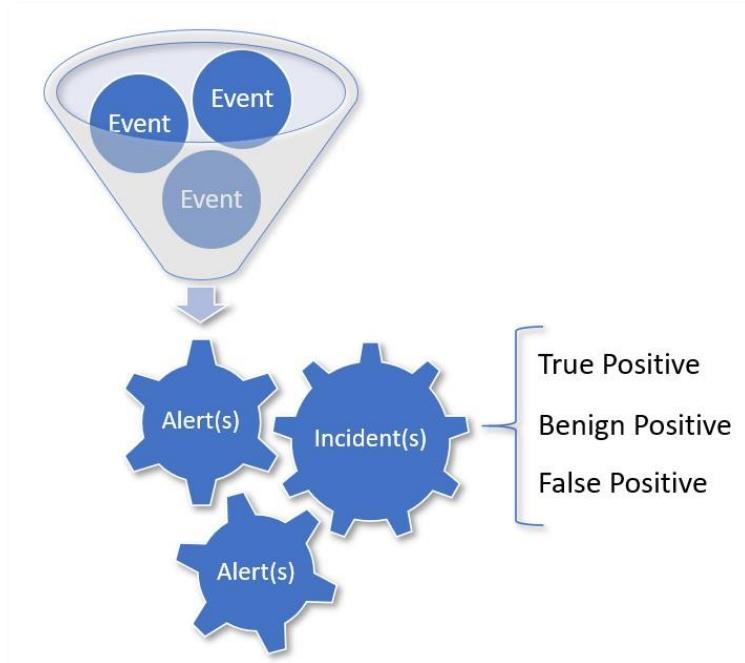


Abbildung 13-14: Visuelle Darstellung, wie Ereignisse, Warnungen und Vorfälle zusammenpassen.

**Hinweis:** Für manche Organisationen mag der Begriff „Vorfall“ in Microsoft Sentinel verwirrend sein. Ein Vorfall wird oft anders verwendet, etwa für tatsächliche Sicherheitsverstöße oder Probleme wie kompromittierte Benutzerkonten.

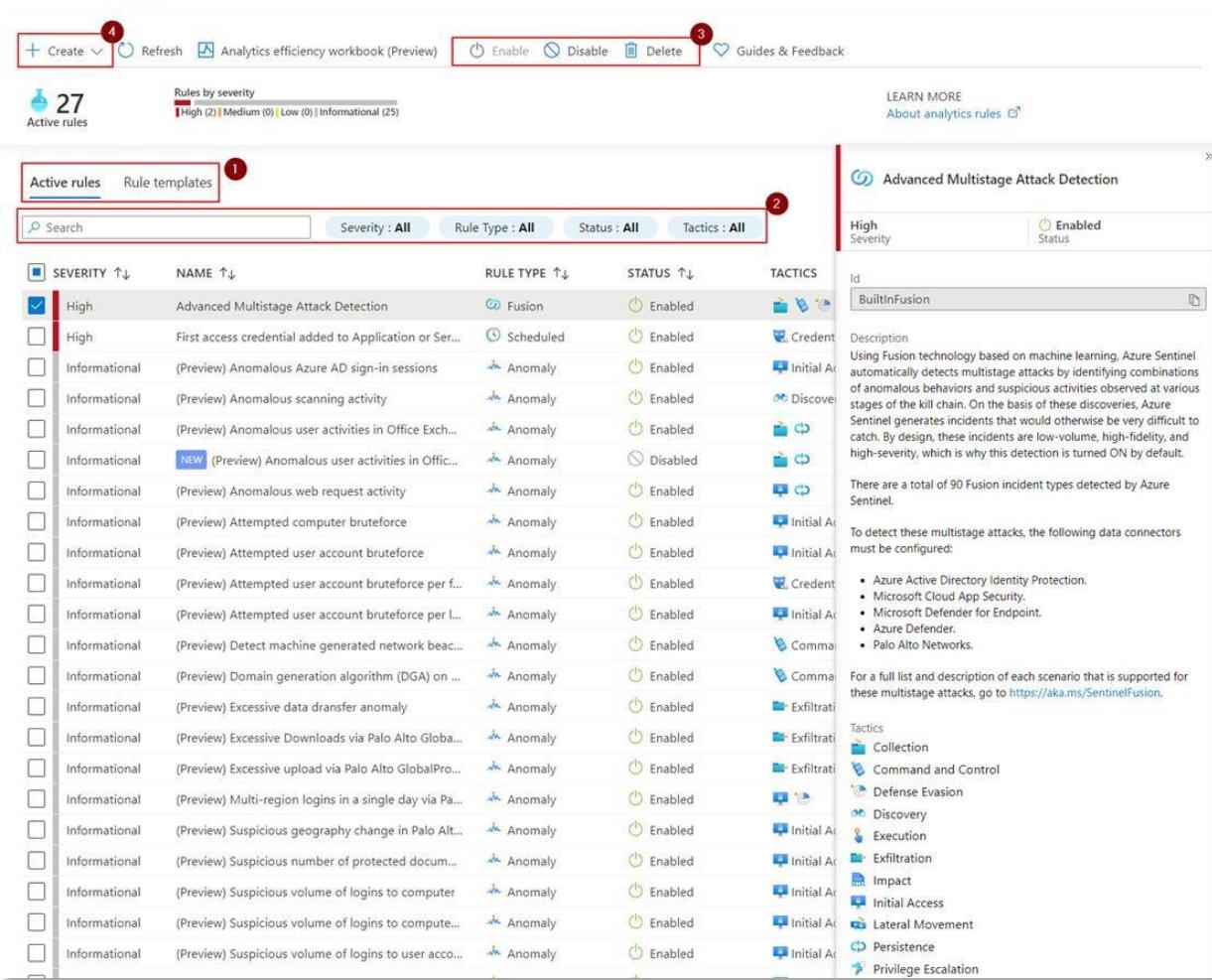
Innerhalb von Microsoft Sentinel haben Warnungen und Vorfälle jeweils eigene Tabellen. Warnungen werden in der Tabelle **SecurityAlert** gespeichert, Vorfälle in der Tabelle **SecurityIncident**. Die Tabelle **SecurityIncident** enthält außerdem ein Protokoll aller Änderungen an einem Vorfall, etwa wer ihn geschlossen hat.

## Einführung in Analyseregeln

Um Warnungen und Vorfälle zu generieren, musst du Analyseregeln erstellen. Mit diesen Regeln kannst du Warnungen und Vorfälle basierend auf bestimmten Ereignissen in deiner Umgebung generieren. In Microsoft Sentinel gibt es sieben verschiedene Arten von Regeln:

- **Geplante** Regeln sind die am häufigsten verwendeten, da sie auf KQL-Abfragen basieren und in regelmäßigen Intervallen ausgeführt werden - zum Beispiel alle paar Minuten, stündlich oder einmal täglich.
- **Near Real Time Detections** oder NRT sind Regeln, die wesentlich häufiger ausgeführt werden als geplante Regeln und dadurch eine schnellere Reaktion ermöglichen.
- **Microsoft Security Incident Creation** Regeln sind integrierte Regeln, mit denen du Vorfälle basierend auf Warnungen generieren kannst, die in anderen Microsoft-Sicherheitsprodukten wie Azure Defender oder Identity Protection erstellt wurden .
- **Fusion ML** ist ein spezifischer Algorithmus für maschinelles Lernen, der mehrere Warnungen oder Ereignisse mit geringer Genauigkeit zu einem Vorfall mit hoher Genauigkeit zusammenfasst. Auch wenn diese Regeln dokumentiert sind, kannst du den zugrunde liegenden Algorithmus nicht anpassen.
- **Anomalieregeln** basieren auf "SOC-ML", einem maschinellen Lernmodell, das ungewöhnliches Verhalten erkennt. Diese Regeln sind nicht unbedingt hochgenau, können aber als Ausgangspunkt für weitere Untersuchungen oder proaktiven Suche dienen. Du kannst sie an deine eigenen Anforderungen anpassen.
- **ML-Verhaltensanalyse**: Verwendet Microsoft eigene ML-Algorithmen zur Erkennung verdächtiger Muster. Diese Regeln analysieren Sicherheitsereignisse/Syslog-Daten und sind nicht konfigurierbar. Aktuell gibt es zwei Regeln dieses Typs: "Anomalous SSH" und "RDP Login Detection".
- **Bedrohungsinformationen**: Ermöglichen es Organisationen, Microsofts Bedrohungsservices mit CEF- oder Syslog-Quellen zu verbinden. Weitere Informationen findest du im Abschnitt "Bedrohungsinformationen".

Um aktuelle Analyseregeln zu finden oder neue zu erstellen, navigiere in deinem Microsoft Sentinel-Portal zur Kategorie **Configuration** und wähle **Analytics**. Dort siehst du eine spezifische Benutzeroberfläche, wie in Abbildung 13-15 dargestellt.



The screenshot shows the Azure Sentinel Analytics Rules page. At the top, there are buttons for Create (4), Refresh, and Analytics efficiency workbook (Preview). Below these are buttons for Enable (3), Disable, Delete, and Guides & Feedback.

On the left, a sidebar shows 27 Active rules. The main area displays a table of rules with columns: SEVERITY ↑↓, NAME ↑↓, RULE TYPE ↑↓, STATUS ↑↓, and TACTICS. The first rule listed is 'Advanced Multistage Attack Detection'.

The right panel provides a detailed view of the selected rule:

- Advanced Multistage Attack Detection**
- High Severity** (Enabled Status)
- Description:** Using Fusion technology based on machine learning, Azure Sentinel automatically detects multistage attacks by identifying combinations of anomalous behaviors and suspicious activities observed at various stages of the kill chain. On the basis of these discoveries, Azure Sentinel generates incidents that would otherwise be very difficult to catch. By design, these incidents are low-volume, high-fidelity, and high-severity, which is why this detection is turned ON by default.
- Tactics:** A list of tactics including Collection, Command and Control, Defense Evasion, Discovery, Execution, Exfiltration, Impact, Initial Access, Lateral Movement, Persistence, and Privilege Escalation.
- For a full list and description of each scenario that is supported for these multistage attacks, go to <https://aka.ms/SentinelFusion>.**

Abbildung 13-15: Übersicht über Analyseregeln

Wenn du dich auf der Seite für die Analysekonfiguration befindest, hast du mehrere Optionen:

1. In jeder Regel findest du zwei Registerkarten. Unter **Aktive Regeln** werden die aktuell konfigurierten Regeln angezeigt, die Warnungen und Vorfälle generieren. **Regelvorlagen** findest du Regeln, die du aus dem Content Hub installiert hast und die du konfigurieren kannst.
2. Du kannst innerhalb der Registerkarten nach bestimmten Regeln filtern. Besonders hilfreich sind die Filter **Regeltyp** und **Datenquelle**. Letzterer hilft dir dabei, passende Regelvorlagen für die Datenquellen zu finden, die du gerade einspeist.
3. Wenn du eine aktive Regel auswählst, kannst du sie direkt aktivieren, deaktivieren oder löschen. Nach der Auswahl öffnet sich rechts ein neues Fenster mit einer Übersicht über die Regeln – inklusive Beschreibung und Abfragezusammenfassung.
4. Wenn du eine neue Abfrage erstellen möchtest, klicke oben auf **Erstellen**, um entweder eine geplante Regel oder eine Microsoft Security-Regel zu definieren.

## Vorlagen für Analyseregeln

Eine besonders hilfreiche Funktion in der Microsoft Sentinel-Benutzeroberfläche ist der Vergleich von Analyseregeln. Wenn du eine Regelvorlage bereitstellst, wird eine aktuelle Version dieser Vorlage in deiner Umgebung erstellt. Wenn Microsoft diese Vorlage aktualisiert, wird deine Regel nicht automatisch überschrieben – das ist gut so, denn du möchtest nicht, dass sich deine Sicherheitsüberwachung ohne dein Zutun verändert. Du solltest aber regelmäßig prüfen, ob neue Versionen nützliche Änderungen enthalten.

Wenn deine implementierte Version von der Vorlage abweicht, erscheint in der Registerkarte **Aktive Regeln** eine Schaltfläche mit der Aufschrift **Mit Vorlage vergleichen**, wie in Abbildung 13-16 zu sehen.

 Correlate Unfamiliar sign-in properti...

High Severity	 Enabled Status
------------------	---

**Id**

**Description**  
The combination of an Unfamiliar sign-in properties alert and an Atypical travel alert about the same user within a +10m or -10m window is considered a high severity incident.

**Tactics**

 Initial Access

**Rule query**

```
let Alert1 =  
SecurityAlert  
| where AlertName ==  
"Unfamiliar sign-in  
properties"  
//sample update
```

**Rule frequency**  
Run query every **1 hour**

**Edit** **Compare with template**

Abbildung 13-16: Vergleichen einer Regel mit einer Analyseregel-Vorlage

Ein Klick auf **Vergleichen** zeigt dir deine aktuelle Regel und die aktualisierte Vorlage nebeneinander. Unterschiede werden dabei deutlich hervorgehoben.

Der Vergleich erfolgt im YAML-Format. Du musst durch die Vorlage scrollen, um die vorgenommenen Änderungen zu erkennen. Alle Änderungen, die bei einer Aktualisierung übernommen würden, sind rot markiert – wie in Abbildung 13-17 dargestellt.

[Compare to latest version \(Preview\)](#) [General](#) [Set rule logic](#) [Incident settings \(Preview\)](#) [Automated response](#) [Review and update](#)

Below is a comparison between the YAML representations of the existing rule and the latest version of the template. [Learn more>](#)

Updating this rule will overwrite your existing rule with the latest version of the template.  
Any automation step or logic that makes reference to the existing rule should be verified, in case the referenced names have changed.

Existing rule (version 1.0.0)	Latest version (1.0.0)
<pre>displayName: Correlate Unfamiliar sign-in properties description: &gt;-   The combination of an Unfamiliar sign-in properties   travel alert about the same user within a +10m or -   high severity incident. tactics:   - InitialAccess severity: High query: &gt;-   let Alert1 =     SecurityAlert       where AlertName == "Unfamiliar sign-in properties"       extend UserPrincipalName = tostring(parse_json(Ex-       Account"))       extend Alert1Time = TimeGenerated       extend Alert1 = AlertName       extend Alert1Severity = AlertSeverity</pre>	<pre>displayName: Correlate Unfamiliar sign-in properties description: &gt;-   The combination of an Unfamiliar sign-in properties   travel alert about the same user within a +10m or -   high severity incident. tactics:   - InitialAccess severity: High query: &gt;-   let Alert1 =     SecurityAlert       where AlertName == "Unfamiliar sign-in properties"       extend UserPrincipalName = tostring(parse_json(Ex-       Account"))       extend Alert1Time = TimeGenerated       extend Alert1 = AlertName       extend Alert1Severity = AlertSeverity</pre>

[Review and update](#) [Next : Custom changes >](#)

Abbildung 13-17: Vergleich einer vorhandenen Regel mit der letzten Version

Wenn du alle Änderungen akzeptieren möchtest, kannst du einfach auf „Überprüfen und aktualisieren“ klicken, um deine bestehende Regel mit der neuen Vorlage zu überschreiben. Wenn du nur bestimmte Änderungen übernehmen möchtest, wird es ein wenig komplizierter. Wie du in der obigen Abbildung sehen kannst, stehen dir verschiedene Registerkarten für die Konfiguration von Analyseregeln zur Verfügung. Diese Registerkarten sind bereits mit den Daten der neuesten Regelvorlage vorausgefüllt. Um einzelne Einstellungen aus der Vorlage mit deiner bisherigen Implementierung zu überschreiben, musst du manuell zur richtigen Registerkarte navigieren und die Regeleigenschaften anpassen. Im gezeigten Beispiel gab es zwei Eigenschaften, die sich von der Vorlage unterschieden: die Abfrage und die Planung. Wenn du deine angepasste Abfrage verwerfen, aber deine benutzerdefinierte Planung beibehalten möchtest, musst du manuell zur Registerkarte „Regellogik“ wechseln, die Planungskonfiguration anpassen (da sie die Konfiguration aus der Vorlage übernommen hat) und auf „Überprüfen und erstellen“ klicken, um die Änderungen zu speichern.

Auch wenn diese Funktion hilfreich ist, um veraltete Regeln zu identifizieren und zu aktualisieren, fehlt meiner Meinung nach noch etwas Wesentliches:

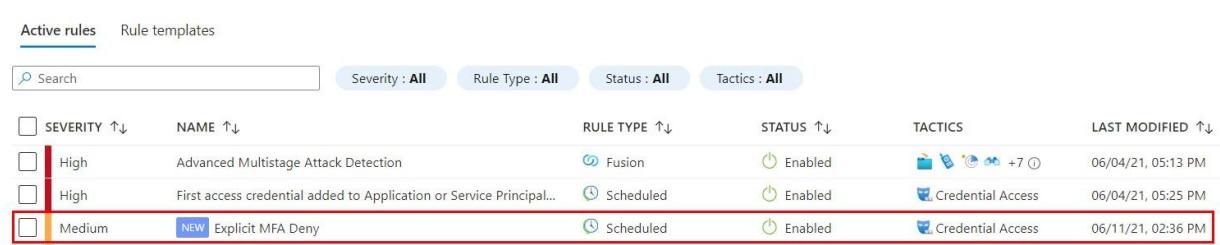
- Es gibt keine Möglichkeit, eine Übersicht über den Status aller deiner Analyseregeln zu erhalten. Um zu prüfen, ob Aktualisierungen verfügbar sind, musst du jede Regel einzeln aufrufen.
- Der Vergleich erfolgt derzeit im YAML-Format. Es wäre deutlich einfacher und übersichtlicher, wenn dieser Vergleich direkt in der Benutzeroberfläche dargestellt würde.

## Geplante Analyseregeln

Geplante Analyseregeln sind mit Abstand die beliebtesten Regeln, da sie es einer Organisation ermöglichen, eigene Regeln nach individuellen Anforderungen zu erstellen. Der wichtigste Teil einer geplanten Regel ist die KQL-Abfrage, die bestimmt, auf welchen Daten eine Warnung oder ein Vorfall basiert. Die Abfrage ruft Informationen aus einer oder mehreren Datentabellen ab und generiert eine Ausgabe. Jede Zeile dieser Ausgabe wird als einzelnes Ereignis betrachtet. Innerhalb der geplanten Regel kannst du festlegen, ab wie vielen Ereignissen eine Warnung generiert werden soll, in welchen Abständen die Abfrage ausgeführt wird und ob eine automatisierte Aktion stattfinden soll.

Wenn du gerade erst anfängst, kannst du die verschiedenen Regelvorlagen durchsuchen, um passende Regeln zu identifizieren und zu aktivieren. Um eine Vorlage zu aktivieren, wähle die Registerkarte „Regelvorlagen“, suche die gewünschte Regel und klicke auf der rechten Seite auf „Regel erstellen“. Dadurch öffnet sich der Assistent „Neue Regel aus Vorlage erstellen“. In den verschiedenen Registerkarten (wie „Allgemein“, „Regellogik“, „Vorfalleinstellung“ usw.) kannst du die Vorlage nach deinen Bedürfnissen anpassen – z. B. Name, Schweregrad oder Abfrage. Wenn du den Assistenten abgeschlossen hast, klicke auf „Erstellen“, um die Regel zu aktivieren.

Nach der Erstellung erscheint die Regel in der Registerkarte „Aktive Regeln“ (wie in Abbildung 13-18 dargestellt) und beginnt, auf Basis der eingehenden Daten in deiner Microsoft Sentinel-Umgebung Warnungen und Vorfälle zu generieren.



The screenshot shows the Microsoft Sentinel Active rules page. At the top, there are tabs for "Active rules" (which is selected) and "Rule templates". Below the tabs are filters for "Severity : All", "Rule Type : All", "Status : All", and "Tactics : All". A search bar is also present. The main area displays a table of rules:

SEVERITY ↑↓	NAME ↑↓	RULE TYPE ↑↓	STATUS ↑↓	TACTICS	LAST MODIFIED ↑↓
<input type="checkbox"/> High	Advanced Multistage Attack Detection	Fusion	Enabled	+7	06/04/21, 05:13 PM
<input type="checkbox"/> High	First access credential added to Application or Service Principal...	Scheduled	Enabled	Credential Access	06/04/21, 05:25 PM
<input type="checkbox"/> Medium	NEW Explicit MFA Deny	Scheduled	Enabled	Credential Access	06/11/21, 02:36 PM

Abbildung 13-18: Eine neu erstellte Regel im Microsoft Sentinel-Portal

Wenn du Regelvorlagen aktivierst, wirst du feststellen, dass es viele Konfigurationsoptionen gibt. Diese sind auf vier verschiedene Registerkarten verteilt: „Allgemein“, „Regellogik“, „Vorfalleinstellungen“ und „Automatisierte Antwort“. Nachdem wir die einzelnen

Konfigurationsmöglichkeiten durchgegangen sind, werden wir gemeinsam unsere ersten Analyseregeln erstellen – dabei zeige ich dir auch einige praktische Beispiele.

## Konfigurationsoptionen

### Allgemein

Die Registerkarte „Allgemein“ enthält generische Einstellungen wie den Namen und die Beschreibung der Regel.

Es gibt fünf Optionen. Die ersten vier kannst du bei Bedarf über die Warnungsanreicherung überschreiben (siehe nächster Abschnitt).

- **Name:** Der Anzeigenamen der Analyseregel. Vorfälle oder Warnungen, die aus dieser Regel resultieren, tragen denselben Namen. Deshalb solltest du einen aussagekräftigen Namen wählen, der den Zweck der Regel gut beschreibt.
- **Beschreibung:** Die Beschreibung sollte das Ziel der Regel erklären und Hinweis geben, wie sie untersucht werden kann. Sie gilt sowohl für die Warnung als auch für den Vorfall.
- **Taktiken und Techniken:** Hier kannst du eine MITRE ATT&CK-Taktik und -Technik zuweisen. Auch wenn dieses Feld optional ist, empfehle ich dir dringend, es zu nutzen, da es in späteren Auswertungen sehr nützlich ist.
- **Schweregrad:** Der Schwerpunkt gibt an, wie dringend ein SOC-Analysten auf eine Warnung oder einem Vorfall reagieren sollte.
- **Status:** Aktiviert oder deaktiviert die Regel. Eine deaktivierte Regel wird nicht ausgeführt und erzeugt keine Warnung oder Vorfälle.

In der Registerkarte „Regellogik“ gibt es mehrere zentrale Einstellungen. Zunächst definierst du hier die eigentliche **Abfrage** – sie ist das Herzstück jeder geplanten Regel. Eine gut durchdachte Abfrage ist entscheidend. Wenn du solche Abfragen entwickelst, denke bitte an Folgendes:

- Versuche, falsch-positive oder gutartige Treffer so weit wie möglich zu reduzieren.
- Nutze so viele relevanten Daten wie möglich, zum Beispiel nicht nur Microsoft Defender-Daten, sondern auch Logs aus Proxy-Servern oder Secure Web Gateways.
- Dokumentiere deine Abfragen. Das klingt banal, ist aber sehr wichtig – verwende Kommentare, damit du (oder jemand anders) später nachvollziehen kannst, was die Regeln bewirken soll.

Zusätzlich zur eigentlichen Regel kannst du auch die **Warnungsanreicherung** einrichten. Damit kannst du die resultierenden Warnungen mit Zusatzinformationen versehen, die einem SOC-Analysten bei der Untersuchung helfen.

Es gibt drei Arten der Warnungsanreicherung:

- **Entitätszuordnung:** Eine Entität ist eine Möglichkeit, Warnungsdaten einem bestimmten vordefinierten Microsoft Sentinel-Attribut zuzuordnen. Microsoft Sentinel bietet eine Reihe von integrierten Entitäten. Es gibt keine Möglichkeit, eigene hinzuzufügen. Einige Beispiele sind Benutzer, IP, E-Mail, Host. Jede Entität hat ein paar Identifikatoren für die korrekte Zuordnung. Zum Beispiel: Benutzer hat Identifikatoren wie SID und vollständiger Name. Je nach den Daten deiner Abfrage musst du den richtigen Identifikator wählen. Die Verfügbarkeit mehrerer Identifikatoren ermöglicht Granularität und Unterstützung für verschiedene Produkte. Lokale Protokolle enthalten die SID, während Cloud-Protokolle die AadUserId enthalten. Entitäten werden im gesamten Produkt verwendet (in UEBA, um maschinelle Lernfähigkeiten hinzuzufügen, in Automatisierungsfunktionen und bei der Untersuchung von Vorfällen). Es ist wichtig, dass du dir die Zeit nimmst, die richtigen Spalten aus deiner Abfrage den Entitäten zuzuordnen. Auch wenn du sie zu diesem Zeitpunkt vielleicht nicht brauchst, werden wir im weiteren Verlauf des Kapitels weiter auf die Verwendung von Entitäten eingehen und wie sie im gesamten Produkt verwendet werden.
- **Benutzerdefinierte Details:** Benutzerdefinierte Details ermöglichen es dir, Schlüssel-Wert-Paare zu erstellen, die der Warnung hinzugefügt werden, um zusätzliche Informationen bereitzustellen. Diese benutzerdefinierten Details sind in den Warnungsmetadaten verfügbar, die während der Untersuchung oder für die Berichterstellung verwendet werden können. Dies sollte verwendet werden, um einer Warnung Details hinzuzufügen, die standardmäßig nicht vorhanden sind. Meistens werden diese verwendet, wenn du Informationen hinzufügen möchtest, für die keine Entität verfügbar ist. Benutzerdefinierte Details werden nicht im Portal angezeigt, sind aber über Playbooks, in der Tabelle SecurityAlert oder in den Protokollen verfügbar.
- **Warnungsdetails:** Mit Warnungsdetails kannst du Eigenschaften einer Warnung überschreiben, die in der allgemeinen Regelkonfiguration konfiguriert sind. Damit kannst du:
  - Erhöhe die Priorität, wenn bestimmte Bedingungen erfüllt sind (vielleicht sollten Warnungen für deine Führungskräfte eine höhere Priorität haben).
  - Gib einen detaillierten Namen und eine aussagekräftige Beschreibung an, die speziell auf die jeweilige Warnung zugeschnitten sind. So haben deine Analysten schnellst möglich viele Informationen zur Hand. Statt eines generischen Namens wie "Eine verdächtige Anmeldung ist aufgetreten" könnte der Name lauten "Bob hatte eine verdächtige Anmeldung auf Laptop5".
  - Aktualisiere Warnungseigenschaften wie den Produktnamen oder den Warnungslink. So kannst du z. B. auf Tools von Drittanbietern verweisen, die Warnungen in deine Microsoft Sentinel-Umgebung übertragen.

Durch die Verwendung von Warnungsdetails kannst du die Informationen auf Basis der Ergebnisse oder der Abfrage aktualisieren.

Unter **Abfrageplanung** legst du fest, wie oft eine Abfrage ausgeführt werden soll und welchen Zeitraum sie abdecken soll. Mit der Konfiguration der Ausführungs frequenz bestimmst du, wie

schnell nach der Datenerfassung eine Warnung generiert werden kann. Bei besonders kritischen Warnungen, etwa einer Anmeldung mit einem Break-Glass-Konto, sollte die Abfrage möglichst häufig ausgeführt werden. Eine Regel, die viele Daten auswertet, etwa zur Erkennung anomaler Anmeldungen, kann seltener laufen, da Microsoft Sentinel die Datenmenge begrenzt, die bei häufigen Abfragen berücksichtigt wird.

Innerhalb der Abfrageplanung hast du drei Optionen zur Verfügung:

- **Häufigkeit:** Legt fest, wie oft eine Abfrage ausgeführt wird. Das Minimum liegt bei 5 Minuten, das Maximum bei 14 Tage.
- **Suchzeitraum:** Bestimmt, wie viele Daten in die Abfrage einbezogen werden. Der Zeitraum muss mindestens so lang sein wie die Abfragehäufigkeit.
- **Start der Ausführung:** Mit dieser Einstellung kannst du definieren, wann die Regel ausgeführt werden soll. Standardmäßig ist sie auf **automatisch** eingestellt, d. h. sie wird bei der Erstellung ausgeführt und folgt danach der Häufigkeit. Wenn du beispielsweise eine Regel um 18 Uhr erstellst, die alle 24 Stunden ausgeführt wird, wird die Regel an den folgenden Tagen jeweils um 18 Uhr ausgeführt. Das ist vielleicht nicht ideal, da dies normalerweise am Ende eines Arbeitstages liegt. Du hast die Option, "**Zu bestimmter Zeit**" zu wählen, was dir erlaubt, die Ausführungszeit des ersten Laufs zu konfigurieren. Hier kannst du wählen, dass deine täglichen Regeln um 13 Uhr laufen, um sicherzustellen, dass du genügend Zeit an deinem Tag hast, um sie zu untersuchen. Das ist am sinnvollsten für Regeln, die alle 6 Stunden oder weniger häufig laufen. Häufigere Regeln laufen mehrmals am Tag, so dass dies weniger wichtig ist.

Standardmäßig verwendet deine Abfrage den Suchzeitraum, um die richtigen Daten zu analysieren. Du kannst dies in der KQL-Abfrage durch eine eigene Zeitdefinition überschreiben, z. B. durch die Angabe:

```
| where TimeGenerated > ago(5d)
```

Ein ausführlicheres Beispiel für eine KQL-Abfrage würde so aussehen. In diesem Beispiel werden die SignInLogs abgefragt, aber nur Ereignisse der letzten 5 Tage abgerufen:

```
SignInLogs
| where TimeGenerated > ago(5d)
```

Da eine Abfrage ein oder mehrere Ereignisse zurückgibt, kannst du mit dem **Warnungsschwellenwert** festlegen, wie viele Ereignisse nötig sind, bevor eine Warnung ausgelöst wird – z. B. bei einem oder fünf Treffern.

Jedes Ergebnis der Regel gilt als Ereignis. Über die **Ereignisgruppierung** kannst du steuern, ob alle Ereignisse zu einer einzigen Warnung zusammengefasst werden oder ob jede Warnung separat erzeugt wird. So kannst du festlegen, ob bei jeder Ausführung nur eine oder mehrere Warnungen generiert werden.

Die letzte Option in der Regellogik ist die **Unterdrückung**. Diese bestimmt, dass nach dem Auslösen einer Warnung für eine bestimmte Zeit keine neue Warnung aus derselben Regel erzeugt wird. Das kann sinnvoll sein, um Warnungsschwämme zu vermeiden.

**Hinweis:** Auch wenn diese Funktion praktisch erscheint, kann sie dazu führen, dass du wichtige Ereignisse/Warnungen verpasst. Verwende sie nur bei Regeln, bei denen zusätzlich Warnungen keinen Mehrwert für die Analyse bieten. Andernfalls empfiehlt sich die Verwendung der **Vorfallsgruppierung** (siehe Abschnitt „Vorfallseinstellungen“).

## Vorfallseinstellungen

Während wir konfigurieren, wann und wie Warnungen generiert werden sollen, erfolgt die Konfiguration speziell für Vorfälle in den Vorfallseinstellungen. Die erste Konfiguration ist vielleicht die einfachste, und sie betrifft die **Vorfallerstellung**. Hier kannst du aktivieren oder deaktivieren, ob Vorfälle aus Warnungen dieser Regel erstellt werden. Standardmäßig wird für jede generierte Warnung ein Vorfall erstellt. Ob du aus Warnungen Vorfälle generieren möchtest, hängt stark von der Art der Regel ab, die du erstellt hast. Wenn es sich um eine Warnung mit geringer Genauigkeit handelt, die keine sofortige Untersuchung erfordert (z. B. wenn sich ein Benutzer von einer unbekannten Anwendung aus anmeldet), ist eine Warnung möglicherweise ausreichend. Diese Warnung kann als Teil anderer geplanter Warnungsregeln (zum Beispiel durch die Kombination der Tatsache, dass sich der Benutzer von einer unbekannten Anwendung aus angemeldet hat, mit der Tatsache, dass er eine neue Posteingangsregel erstellt hat), während einer Untersuchung (über die Zeitachse) oder durch proaktive Suche verwendet werden. Wenn deine Warnung eine sofortige Untersuchung erfordert und eine hohe richtig-positive Rate hat, wird empfohlen, Vorfälle dafür zu erstellen.

Neben den Einstellungen für die Vorfallerstellung hast du auch die Möglichkeit, die **Vorfallsgruppierung** zu konfigurieren. Mit diesen Einstellungen kannst du festlegen, wann jede Warnung zu einem Vorfall werden soll oder ein Vorfall mehrere Warnungen enthalten kann. Das sind extrem leistungsstarke Optionen, die es dir ermöglichen, „Warnungsspam“ zu vermeiden, wenn du mit deinem Security Operations Center (SOC) arbeitest.

Neben der Aktivierung bzw. Deaktivierung der Warnungsgruppierung stehen dir weitere Konfigurationsmöglichkeiten zur Verfügung:

- Begrenzung der Gruppierung von Warnungen auf einen bestimmten Zeitraum (konfigurierbar in Minuten, Stunden und Tage). Wenn du beispielsweise den Zeitraum auf fünf Stunden festlegst und die Warnung Y vier Stunden nach der Warnung X erstellt wird, werden beide Warnungen zu einem einzigen Vorfall zusammengefasst. Das bedeutet, dass der ursprüngliche Vorfall aktualisiert und die neue Warnung hinzugefügt wird. Für die Warnung Z, die sechs Stunden nach der Warnung X erstellt wurde, wird hingegen ein separater Vorfall erzeugt.
- Die Bestimmung, ob zwei Warnungen übereinstimmen, erfolgt durch Entitäten und benutzerdefinierte Details. Dafür gibt es drei Optionen:

- Gruppiere Warnungen, wenn alle Entitäten übereinstimmen.
- Gruppiere alle Warnungen (unabhängig von den Entitäten).
- Gruppiere Warnungen basierend auf bestimmten Entitäten und benutzerdefinierten Details.
- Mit der letzten Einstellung innerhalb der Warnungsgruppierung kannst du festlegen, ob bereits geschlossene Vorfälle erneut geöffnet werden sollen, wenn sie mit neuen Warnungen übereinstimmen. Auch wenn das die letzte Einstellung ist, ist sie keineswegs unwichtig. Wenn du einen Vorfall mit vier Warnungen hattest und ihn als falsch positiv geschlossen hast, ist es möglich, dass dem gleichen Vorfall eine weitere Warnung hinzugefügt wird - sofern der Zeitrahmen und die Entitäten übereinstimmen. In diesem Fall könntest du potenziell wichtige Warnungen verpassen.

**Hinweis:** Die Konfiguration der Warnungsgruppierung unterscheidet sich je nach Art der Analyseregeln. Während die meisten Vorlagen diese Funktion nicht nutzen, empfehle ich sie ausdrücklich. Ich verwende sie häufig, um Warnungen, die sich auf dasselbe Konto oder denselben Host beziehen, in einem einzigen Vorfall zusammenzufassen. Theoretisch könntest du jede Warnung - sogar ohne übereinstimmende Entitäten - in einem einzigen Vorfall zusammenführen. Das kann jedoch die Untersuchung erschweren, da eine Warnung für Benutzer A möglicherweise als falsch positiv geschlossen wird, während eine Warnung für Benutzer B noch untersucht werden muss und mit hoher Wahrscheinlichkeit ein richtig Positiver ist. Ein konkretes Beispiel dafür findest du im Abschnitt "Erstellen deiner ersten geplanten Analyseregel" weiter unten.

## Automatisierte Antwort

Die letzte Registerkarte im Assistenten zum Erstellen von Analyseregeln legt fest, welche automatisierten Antworten ausgeführt werden sollen, wenn eine Warnung oder ein Vorfall für diese geplante Analyseregel erstellt wird. Du kannst konfigurieren, welche Automatisierung ausgeführt werden soll, wenn eine Warnung erstellt wird und wenn ein Vorfall erstellt wurde. Automatisierung wird später im Abschnitt „Automatisierung von Antworten“ ausführlich erläutert.

## Erstellen deiner ersten geplanten Analyseregel

Lass uns ein Beispiel durchgehen und unsere allererste geplante Analyseregel erstellen. Ich werde ein Beispiel verwenden, das auf der [Microsoft Sentinel GitHub-Seite](#) verfügbar ist – „Rare Application Consent“ – und es an unsere Bedürfnisse anpassen. App-Zustimmungen und die damit verbundenen Gefahren wurden bereits in Kapitel 2 besprochen. Auch wenn es einige Maßnahmen gibt, mit denen du die Möglichkeiten des Endbenutzers zur Zustimmung zu einer Anwendung einschränken kannst, wird empfohlen, zu überwachen, welche Art von Anwendungszustimmungen in deiner Umgebung stattfinden. Diese Abfrage sucht nach Anwendungen, die von Microsoft Sentinel in den letzten 14 Tagen nicht gesehen wurden.

## Erstellen der Abfrage

Der erste Schritt für jede neue Regel ist die Erstellung der KQL-Abfrage. Auch wenn manche Microsoft-Sentinel-Abfragen zunächst abschreckend wirken mögen, sind sie – wenn man sie in einzelne Teile zerlegt – relativ leicht zu interpretieren und zu erstellen.

Lass uns zunächst unsere Variablen erstellen. Variablen werden oft am Anfang definiert und dann in verschiedenen Teilen der Abfrage wiederverwendet. In diesem Beispiel verwenden wir drei Variablen:

- **Current.** Für welchen Zeitraum möchtest du die aktuellen Anwendungszustimmungen überwachen? Möchtest du einen Tag, ein Stunde, fünf Stunden zurückblicken?
- **AuditLookBack.** Wie viele Tage soll die Abfrage in die Vergangenheit blicken, um den Audit-Verlauf zu erstellen?
- **Threshold.** Die Anzahl der Zustimmungen zu einer Anwendung, bevor sie in der Abfrage ausgeschlossen wird. Dies ist eine Möglichkeit, das Rauschen zu filtern. Wenn eine Anwendung in den letzten sieben Tagen mehr als drei Mal gesehen wurde, wird sie herausgefiltert und nicht mehr als 'selten' betrachtet.

```
Let current = 1d;  
  
let auditLookback = 7d; let  
  
threshold = 3;
```

Die Konfiguration dieser Zeitvariablen bestimmt, wie die Abfrageplanungseinstellungen deiner Regel aussehen werden.

Der zweite Schritt ist der Aufbau des Audit-Verlaufs. Dieser wird verwendet, um zu sehen, welchen Anwendungen in der Vergangenheit zugestimmt wurde. Durch den Aufbau dieser Daten kannst du später in der Abfrage überprüfen, ob die Zustimmungen, die in den letzten 24 Stunden erfolgt sind, Anwendungen betreffen, die bereits bekannt sind.

Um den Audit-Verlauf zu erstellen, werden in der folgenden Abfrage einige Schritte durchgeführt:

- Wir verwenden die **AuditLogs**-Tabelle, da diese alle Entra ID-Audit-Protokolle enthält.
- Um nach historischen Daten zu suchen, filtern wir nach **TimeGenerated** und nutzen die beiden Variablen **auditLookback** und **current**, um Protokolle aus den letzten sieben Tagen abzurufen (dies ist in der Variable **auditLookback** konfiguriert).
- Durch eine **where**-Klausel für die Spalte **OperationName** filtern wir die Audit-Protokolle so, dass nur Anwendungszustimmungen zurückgegeben werden.

- Der **extend**-Befehl wird verwendet, um den Benutzer zu extrahieren, der die Aktion initiiert hat, indem wir in die JSON-Objekte schauen, die in den Audit-Protokollen gespeichert sind.
- Summarize** wird verwendet, um mehrere Zeilen zu einer zusammenzufassen. Jede Zeile, die denselben **OperationName**, den initiierenden Benutzer und dieselben Anwendungswerte hat, wird zusammengeführt.
- Der letzte Schritt besteht darin sicherzustellen, dass wir nur Protokolle für Vorgänge abrufen, die mehr als dreimal stattgefunden haben (konfiguriert in der Variablen **threshold**).

```
Let AuditTrail = AuditLogs | where TimeGenerated >= ago(auditLookback) and
TimeGenerated < ago(current)
| where OperationName has "Consent to application"
| extend InitiatedBy =
iff(isnotempty(tostring(parse_json(tostring(InitiatedBy.user)).userPrincipalName)), tostring(parse_json(tostring(InitiatedBy.user)).userPrincipalName),
tostring(parse_json(tostring(InitiatedBy.app)).displayName))
| extend TargetResourceName =
tolower(tostring(TargetResources.[0].displayName))
| summarize max(TimeGenerated), OperationCount = count() by OperationName,
InitiatedBy, TargetResourceName
| where OperationCount > threshold;
```

Nachdem wir die Audit-Daten abgerufen haben, ist es an der Zeit, die aktuellen Anwendungszustimmungen abzurufen. Du wirst viele Befehle wiedererkennen, da der Prozess ähnlich ist:

- Rufe die **AuditLogs**-Tabelle auf
- Filtere nur die aktuellsten Protokolle (in unserem Fall vom letzten Tag)
- Erstelle zusätzliche Spalten für die IP-Adresse und die imitierenden Benutzer

```
let RecentConsent = AuditLogs | where TimeGenerated >= ago(current)
| where OperationName has "Consent to application"
| extend IpAddress =
iff(isnotempty(tostring(parse_json(tostring(InitiatedBy.user)).ipAddress)),
tostring(parse_json(tostring(InitiatedBy.user)).ipAddress),
tostring(parse_json(tostring(InitiatedBy.app)).ipAddress))
| extend InitiatedBy =
iff(isnotempty(tostring(parse_json(tostring(InitiatedBy.user)).userPrincipalName)), tostring(parse_json(tostring(InitiatedBy.user)).userPrincipalName),
tostring(parse_json(tostring(InitiatedBy.app)).displayName))
| extend TargetResourceName =
tolower(tostring(TargetResources.[0].displayName))
| parse TargetResources.[0].modifiedProperties with * "ConsentType: "
ConsentType "]" *
```

```
| project TimeGenerated, InitiatedBy, IPAddress, TargetResourceName,
Category, OperationName, ConsentType , CorrelationId, Type;
```

Jetzt ist es an der Zeit, die historischen und aktuellen Daten zu kombinieren. Durch Verwendung einer Join-Operation kannst du zwei Tabellen zu einer zusammenführen. In diesem Beispiel wird ein Left-Anti-Join verwendet. Das bedeutet, dass alle Datensätze aus der linken Tabelle abgerufen werden, die keine Übereinstimmung mit einem Datensatz auf der rechten Seite haben. Anders gesagt: Du gibst nur aktuelle Zustimmungen zurück, für die es in den letzten sieben Tagen keine ähnlichen Aktivitäten gab (die also nicht im Audit-Verlauf enthalten sind).

Der letzte Schritt besteht darin, die Daten zu bereinigen, indem du eine zusätzliche Spalte erstellst und andere entfernst. Das ist nicht zwingend erforderlich, stellt aber sicher, dass der SOC-Analyst, der diese Warnung untersucht, nur die wichtigsten Daten sieht.

```
Let RareConsentApp = RecentConsent | join kind= leftanti AuditTrail on
OperationName,
TargetResourceName
| extend Reason = "Previously unseen app granted consent";
| summarize Reason = makeset(Reason) by TimeGenerated, InitiatedBy,
IPAddress,
TargetResourceName, Category, OperationName, ConsentType, CorrelationId, Type
```

## Erstellen der Regel

Mit der fertigen Abfrage ist der nächste Schritt die Erstellung der Regel. Navigiere im Microsoft Sentinel-Portal zu **Analytics > Create > Scheduled Rule**. Dadurch wird der Assistent für Analyseregeln gestartet. Der erste Schritt besteht darin, die Regeldetails zu konfigurieren, wie in Tabelle 13-5 dargestellt:

Eigenschaft	Wert	Begründung
Name	Rare Application Consent	Der Name ist Geschmackssache, sollte aber beachtet werden, da Vorfälle ebenfalls mit diesem Namen erstellt werden.
Beschreibung	Diese Regel erzeugt Vorfälle, wenn ein Benutzer/Administrator einer Anwendung zustimmt, der in den letzten 7 Tagen nicht zugestimmt wurde. Es ist dann eine Untersuchung der spezifischen	Ihre Beschreibung sollte erklären, was die Regel bewirkt, und einen Hinweis zur Untersuchung liefern.

	Anwendung erforderlich, um die Auswirkungen zu bewerten.	
Taktiken	Persistence Lateral Movement Collection	Bei der Auswahl der Taktiken sollten Sie überlegen: „Wenn in meiner Organisation ein Angriff stattfände, in welcher Phase wäre dieser Vorfall, und welche Auswirkungen hätte er?“
Schweregrad	Niedrig	Die Einstufung des Schweregrads ist wichtig für Konsistenz über alle Regeln hinweg. Für diese Regel wähle ich „Niedrig“, da sie nicht zwingend auf einen tatsächlichen Angriff hinweist.
Status	Aktiviert	Nur wenn die Regel aktiviert ist, werden Alerts und Vorfälle generiert.

*Tabelle 13-5: Regeldetails der Analyseregel*

## Regellogik

Auf der Registerkarte „**Regellogik**“ fügst du die Abfrage hinzu. Nimm die Abfrage, die wir oben erstellt haben, und füge sie in das Feld „Regelabfrage“ ein. Jetzt ist ein guter Zeitpunkt, um die Abfrage zu validieren und zu prüfen, welche Auswirkungen sie in deiner Organisation haben könnte. Das kannst du im Bereich **Ergebnissimulation** auf der rechten Seite tun (wie in Abbildung 13-19 gezeigt). Klicke nach dem Einfügen der Abfrage auf **Mit aktuellen Daten testen**. Wenn du eine hohe Anzahl von Treffern siehst, ist es möglicherweise sinnvoll, die Abfrage weniger „laut“ zu machen oder die Vorfallsgruppierung für diese Regel anzupassen.

#### Results simulation

This chart shows the results of the last 50 evaluations of the defined analytics rule. Click a point on the chart to display the raw events for that point in time.

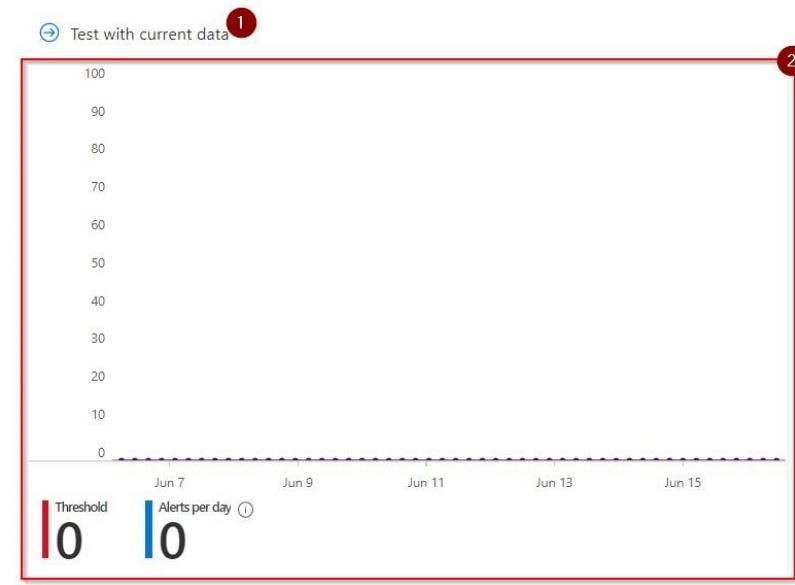


Abbildung 13-19: Simulation der Ergebnisse einer Analyseregel

Als Nächstes kommt die **Warnungsanreicherung**, die drei verschiedene Kategorien umfasst.

Die **Entitätszuordnung** ist die wichtigste Kategorie und sollte für jede Abfrage konfiguriert werden, da sie für die Vorfallsgruppierung, UEBA, Untersuchungen und Automatisierung verwendet wird. Die Liste der möglichen Entitäten wächst ständig und die Konfiguration hängt stark von deiner spezifischen Abfrage und deinen Anwendungsfällen ab. Für die Abfrage zur Anwendungszustimmung würde ich die folgende Konfiguration empfehlen, wie sie in Tabelle 13-6 dargestellt ist:

Entitätstyp	Bezeichner	Wert	Begründung
Konto	FullName	InitiatedBy	Der Benutzer, der der Anwendung zugestimmt hat.
Cloud-Anwendung	Name	TargetResourceName	Die Anwendung, der der Benutzer zugestimmt hat.
IP	Address	IpAddress	Von welcher Adresse die Zustimmung erfolgte.

Tabelle 13-6: Konfiguration der Entitätszuordnung

**Benutzerdefinierte Details** sind meiner Meinung nach nicht zwingend notwendig, aber eine gute Möglichkeit, einer Warnung zusätzliche Informationen hinzuzufügen, die nicht in eine Entität passen. In unserem Fall würde ich zum Beispiel den Zustimmungstyp hinzufügen – also die Information, ob es sich um eine Benutzer- oder Administratorzustimmung handelt.

**Warnungsdetails** werden verwendet, um die Standardregelkonfiguration zu überschreiben. Das kann in unserem Szenario in folgenden Fällen nützlich sein:

1. Füge die Cloud-Anwendung in die Beschreibung der Warnung ein, damit sie für den SOC-Analysten sofort ersichtlich ist. Auch wenn die App als Entität hinzugefügt wird, kann eine zusätzliche Beschreibung helfen, die richtigen Informationen schneller zu finden.
2. Ändere den Schweregrad entsprechend dem Benutzer, der die Zustimmung ausgeführt hat. Wenn etwa der CEO einer Anwendung zugestimmt hat, könnte das eine mittlere oder hohe Warnung rechtfertigen. Um auf solche benutzerspezifischen Werten zuzugreifen, empfiehlt sich der Einsatz einer Watchlist. Watchlists ermöglichen es dir, Werte zentral zu verwalten und an verschiedenen Stellen wiederzuverwenden. Sie werden später in diesem Kapitel behandelt.

Für jede geplante Regel musst du die Häufigkeit und den Suchzeitraum konfigurieren. Diese hängen immer von deiner Abfrage und dem zugrunde liegenden Anwendungsfall ab. Wenn du etwa eine Abfrage geschrieben hast, die nach kritischen Ereignissen sucht, sollte die Regel idealerweise alle paar Minuten ausgeführt werden. Der Suchzeitraum definiert, wie weit in die Vergangenheit die Abfrage schaut. In unserem Fall sollte der Suchzeitraum dem Wert der Variable **AuditLookBack** in der Abfrage entsprechen. Tabelle 13-7 zeigt die spezifische Konfiguration.

Eigenschaft	Wert	Begründung
Häufigkeit	1 Tag	Da diese Ereignisse nicht kritisch sind, muss die Regel nicht alle 5 Minuten ausgeführt werden. Weil die Regel die Daten des vorherigen Tages auswertet, ist eine tägliche Ausführung sinnvoll.
Suchzeitraum	7 Tage	Da die Regel Prüfprotokolle der letzten 7 Tage nutzt, benötigt die Abfrage Zugriff auf die Daten dieses Zeitraums.

Tabelle 13-7: Zeitkonfiguration einer Analyseregel

**Hinweis:** Häufigkeit und Suchzeitraum sind zwar zwei separate Einstellungen, hängen aber voneinander ab. Je häufiger die Regel ausgeführt wird, desto kürzer muss der Suchzeitraum sein. Eine Regel, die alle fünf Minuten ausgeführt wird, kann nur einen Suchzeitraum von zwei

Tagen haben. Das kann zu Problemen führen, wenn du versuchst, eine Regel zu erstellen, die historische Daten verwendet. Wenn du auf diese Einschränkung stößt, wird empfohlen, diese historischen Daten stattdessen in einer Watchlist zu speichern, anstatt sie in einer Tabelle abzufragen.

Durch Konfiguration eines **Warnungsschwellenwerts** kannst du festlegen, wie viele Ergebnisse die Abfrage zurückgeben soll, bevor eine Warnung erstellt wird. Da die Abfrage selbst bereits einen eingebauten Schwellenwert enthält, bleibt es bei der Standardkonfiguration „Ist größer als 0“. Ich neige dazu, diese Einstellung nicht zu ändern und Schwellenwerte stattdessen direkt in der Analyseregel zu definieren, da das eine größere Granularität ermöglicht.

Die **Ereignisgruppierung** wird oft übersehen, ist aber sehr wichtig bei der Konfiguration von Warnungsregeln. Es gibt zwei Optionen:

- Gruppiere alle Ereignisse in einer einzigen Warnung (Standard)
- Löse eine Warnung für jedes Ereignis aus

Für diese Regel solltest du „Löse eine Warnung für jedes Ereignis aus“ wählen. Denn bei der Untersuchung möchtest du nur die Aktionen eines einzelnen Benutzers in einer Warnung sehen. Wenn Benutzer A der Anwendung X zustimmt und Benutzer B der Anwendung Y, würden diese beiden Ereignisse bei der Standardkonfiguration in derselben Warnung landen. Zwei separate Ereignisse in einer einzigen Warnung können die Untersuchung und Berichterstellung erschweren. Durch Deaktivieren der Ereignisgruppierung werden zwar mehr Warnungen generiert, aber durch die Verwendung der Vorfallsgruppierung kannst du die Anzahl der erstellten Vorfälle begrenzen.

Die **Unterdrückung** ist nützlich, wenn du eine Abfrage hast, die häufig ausgeführt wird und bei der jede Warnung die gleiche Bedeutung hat. Ein gutes Beispiel ist der Login eines Break-Glass-Kontos. In unserem Fall können sich zwei Warnungen jedoch auf unterschiedliche Benutzer oder Anwendungen beziehen. Daher möchten wir die Warnungsunterdrückung nicht aktivieren.

## Vorfallseinstellungen

Nachdem du die verschiedenen Warnungseigenschaften konfiguriert hast, kannst du zur Registerkarte „**Vorfallseinstellungen**“ wechseln. Hier bestimmst du, ob Vorfälle erstellt werden und wie Warnungen gruppiert werden sollen.

Die erste Einstellung lautet „**Vorfälle aus Warnungen erstellen, die durch diese Analyseregel ausgelöst wurden**“. Sie legt fest, ob für jede durch diese Regel erzeugte Warnung ein Vorfall erstellt wird. In den meisten Fällen ist diese Option aktiviert – so auch in unserem Beispiel. Wenn du jedoch Warnungen mit geringer Genauigkeit hast, die für sich allein genommen keine Bedeutung haben, solltest du diese Einstellung deaktivieren und eine zusätzliche Analyseregel erstellen, die solche Warnungen mit anderen kombiniert, die eine höhere Genauigkeit aufweisen.

Als Nächstes kommt die **Warnungsgruppierung**. Diese ist standardmäßig deaktiviert, was bedeutet, dass für jede Warnung ein separater Vorfall erstellt wird. Durch Aktivierung der Gruppierung kannst du sicherstellen, dass mehrere Warnungen zu einem einzigen Vorfall zusammengefasst werden. Nehmen wir die Regel „Seltene Anwendungszustimmung“ als Beispiel: Wenn ein Benutzer zwei verschiedenen Anwendungen von zwei unterschiedlichen IP-Adressen aus zustimmt – wie viele Vorfälle möchtest du dann erzeugen? Bei dieser Entscheidung solltest du an die spätere Untersuchung und Berichterstattung denken:

- Die Kombination von Warnungen verschiedener Benutzern in einem Vorfall ist schwer zu untersuchen. Eine Untersuchung könnte bereits abgeschlossen sein, während die andere noch offen bleibt. Zwei separate Vorfälle ermöglichen eine bessere Nachverfolgung.
- Die Berichterstellung über richtig/falsch Positive erfolgt auf Vorfallsebene. Wenn du zwei Warnungen in einem Vorfall hast und nur eine davon richtig Positiver, müsstest du eine Entscheidung treffen – und riskierst dabei, eine Warnung falsch zu klassifizieren.

Basierend auf diesen Überlegungen empfehle ich, die Warnungsgruppierung zu aktivieren – entweder, wenn alle Entitäten übereinstimmen oder wenn bestimmte Entitäten übereinstimmen. Nur in sehr speziellen Anwendungsfällen würde ich empfehlen, alle Warnungen unabhängig von ihren Entitäten zu gruppieren. Weitere Details dazu findest du in Tabelle 13-8.

Eigenschaft	Wert	Begründung
<b>Alert-Gruppierung</b>	Aktiviert	Durch Aktivieren der Gruppierung werden „Incident-Spam“ vermieden, bei dem für dasselbe Problem mehrere Vorfälle erstellt werden.
<b>Zeitraum</b>	1 Stunde	Der Zeitraum sollte zur Häufigkeit der Regel passen. Da unsere Regel einmal täglich läuft, sollen nicht Alerts, die 24 Stunden auseinanderliegen, zusammengefasst werden. Daher 1 Stunde.
<b>Gruppieren nach</b>	Cloud-Anwendung & Benutzer-Entität	Durch Auswahl dieser beiden Entitäten wird für jede eindeutige Kombination aus Benutzer und Anwendung ein eigener Vorfall erstellt. Stimmt ein Benutzer zwei verschiedene Anwendungen zu, entstehen zwei separate Vorfälle. So wird verhindert, dass legitime und unlegitime Zustimmungen vermischt werden.
<b>Wiedereröffnung geschlossener Vorfälle</b>	Aktiviert	Wird einem bereits geschlossenen Vorfall ein neuer Alert hinzugefügt, kann dies auf verpasste Entwicklungen hinweisen. Durch Aktivierung wird der Vorfall wieder in der Übersicht angezeigt.

Tabelle 13-8: Konfiguration der Warnungsgruppierung

**Hinweis:** Ob du die Warnungsgruppierung aktivieren solltest, hängt stark von deinen Anforderungen und dem Aufbau deines SOC ab. Ich konfiguriere die Gruppierung oft auf Basis des Benutzerkontos. Das bedeutet, dass ein Vorfall mehrere Warnungen enthalten kann, die sich aber alle auf denselben Benutzer beziehen. Auch wenn die diese Konfiguration nicht für jede Organisation oder Abfrage geeignet ist, ist es wichtig, solltest du dir genau überlegen, wann du lieber mehrere Vorfälle statt einer Gruppierung erzeugen möchtest.

Indem wir die Schritte zur Erstellung einer Analyse Regel durchgegangen sind, solltest du nun eine Einführung in die möglichen Konfigurationen erhalten haben – inklusive meiner Empfehlungen und Begründungen. Diese Einstellungen funktionieren für mich, müssen aber sorgfältig geprüft werden, wenn du Microsoft Sentinel implementierst, um sicherzustellen, dass sie zu deinen Anforderungen passen.

## Empfehlungen zur Optimierung

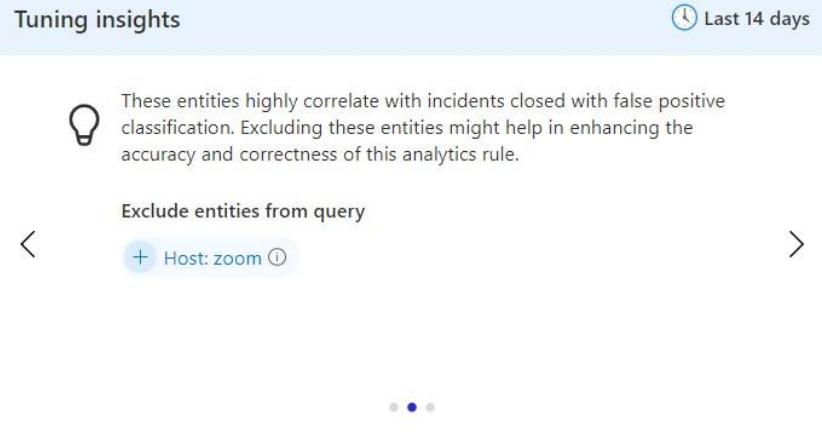
Während die Erstellung von Microsoft Sentinel-Regeln ein Prozess ist, den jede Organisation umsetzen und lernen muss, hilft Microsoft Sentinel dabei, Abfragen zu identifizieren, die optimiert werden können. Über eine Funktion namens „Tuning Recommendations“ werden dir bestimmte Änderungen vorgeschlagen, wenn Microsoft erkennt, dass eine Regel effizienter ausgeführt werden kann.

Wenn es eine Optimierungsempfehlung für eine bestimmte Regel gibt, erkennst du das an einer Glühbirne neben der geplanten Analyse Regel in der Registerkarte „Analytics“, wie in Abbildung 13-20 zu sehen ist.

<input type="checkbox"/> Severity ↑↓	↑↓ Name ↑↓	Rule type ↑↓	Status ↑↓
<input type="checkbox"/> Medium  Rare application consent		 Scheduled	 Enabled

Abbildung 13-20: Identifizierung von Regeln mit aktiven Optimierungsempfehlungen

Innerhalb der Registerkarte „Set Rule Logic“ findest du auf der rechten Seite des Bildschirms einen neuen Abschnitt mit dem Namen „Tuning Insights“, wie in Abbildung 13-21 dargestellt.



The screenshot shows a 'Tuning insights' card for the last 14 days. It features a lightbulb icon with the text: 'These entities highly correlate with incidents closed with false positive classification. Excluding these entities might help in enhancing the accuracy and correctness of this analytics rule.' Below this is a button labeled 'Exclude entities from query'. Navigation arrows and a zoom option are also present.

Abbildung 13-21: Tuning-Einblicke für eine bestimmte Regel

Diese Tuning-Einblicke basieren auf den Abschlussgründen früherer Vorfälle, die durch diese Regel generiert wurden. Wenn eine bestimmte Entität häufig falsch-positive Ergebnisse verursacht, empfiehlt dir Microsoft Sentinel, sie auszuschließen. Klickst du auf das Pluszeichen neben der Empfehlung, wird der Ausschluss automatisch implementiert – durch eine zusätzliche Zeile am Ende der Abfrage, die den betreffenden Wert ausschließt.

Auch wenn dies eine sinnvolle Methode ist, um offensichtlich falsch Positive zu identifizieren, sehe ich darin nur einen ersten Schritt. Solche einfachen Ausschlüsse sind SOC-Analysten oft ohnehin bekannt und sollten auch in deinen Berichten auffallen. Es gibt viele Möglichkeiten, mit falsch Positiven umzugehen – und elegantere Lösungen als harte Ausschlüsse direkt in der Abfrage. In einem späteren Abschnitt dieses Kapitels („Umgang mit falsch Positiven“) gehe ich ausführlich darauf ein und zeige, wie du Watchlists nutzen kannst, um Regeln strukturiert und flexibel anzupassen.

## Near Real Time (NRT) Regeln

Geplante Analyseregeln sind für die meisten Anwendungsfälle gut geeignet, aber manche Organisationen empfinden die Mindesthäufigkeit von fünf Minuten als Einschränkung. Für besonders zeitkritische Fälle bietet Microsoft Sentinel Near Real Time (NRT)-Erkennungen. Diese Regeln basieren ebenfalls auf KQL-Abfragen, werden aber jede Minute ausgeführt.

Das Erstellen einer NRT-Regel funktioniert ähnlich wie bei einer geplanten Analyseregel: Du wählst in der Registerkarte „Analytics“ die Schaltfläche „Erstellen“ und anschließend „**NRT-Abfrageregel**“.

Die Benutzeroberfläche ist fast identisch, allerdings wurden einige Konfigurationen entfernt:

- Die **Abfrageplanung** kannst du nicht anpassen, da die Regel ohnehin minütlich läuft und nur Protokolle der letzten Minute verarbeitet. NRT-Regeln passen sich automatisch an die Erfassungsverzögerung an.
- Ein **Warnungsschwellenwert** ist nicht verfügbar - eine Warnung wird sofort generiert, sobald ein Treffer gefunden wurde.
- Die **Ereignisgruppierung** ist nicht konfigurierbar - alle Ereignisse werden automatisch in einer einzigen Warnung zusammengefasst.

Neben diesen Einschränkungen gibt es auch Einschränkungen bei der Abfrage selbst:

- Der 'join'-Operator ist nicht verfügbar - du kannst also nur Daten aus einer einzigen Tabelle abrufen.
- Die Ausgabe ist begrenzt – verwende den 'project'-Operator, um nur die benötigten Spalten zurückzugeben.
- Die 'search'-Anweisung wird nicht unterstützt.

Mit diesen Einschränkungen solltest du sorgfältig prüfen, ob du eine Regel als reguläre geplante Analyseregel oder als NRT-Regel erstellst. Ich empfehle, geplante Regeln als Standard zu verwenden und NRT-Regeln nur in sehr spezifischen Fällen einzusetzen – zum Beispiel zur Überwachung eines Break-Glass-Kontos.

## Microsoft Incident Creation Rules

Neben geplanten Regeln kannst du auch Microsoft Incident Creation Rules konfigurieren. Diese Regeln erzeugen Vorfälle auf Basis von Warnungen aus anderen Microsoft-Sicherheitsprodukten. Unterstützte Produkte sind:

- Microsoft Defender for Cloud Apps
- Microsoft Defender for Cloud
- Entra ID Identity Protection
- Microsoft Defender for IoT
- Microsoft Defender for Office 365
- Microsoft Defender for Endpoint
- Microsoft Defender for Identity
- Microsoft 365 Insider Risk Management

Es gibt zwei Möglichkeiten, eine solche Regel zu erstellen. Befindest du dich auf der Seite eines dieser Datenkonnektoren, kannst du die Vorfallerstellung für dieses Produkt aktivieren. Sobald du auf „**Aktivieren**“ klickst (siehe Abbildung 13-22), wird die entsprechende Incident Creation Rule erstellt und erscheint in deinen aktiven Analyseregeln.



### Create incidents - Recommended!

Create incidents automatically from all alerts generated in this connected service.

**Enable**

Abbildung 13-22: Incident Creation Rules

**Hinweis:** Bevor Microsoft Incident Creation Rules Auswirkungen haben, musst du die richtigen Warnungen einspeisen. Durch das Verbinden von Microsoft-Sicherheitsprodukten werden Warnungen automatisch zur Tabelle SecurityAlert hinzugefügt. Eine Incident Creation Rule schaut in diese Tabelle und erstellt Vorfälle basierend auf den Warnungen.

Die andere Möglichkeit ist die manuelle Erstellung der Regeln. Das geschieht, indem du zu **Analytics** gehst und **Erstellen > Microsoft Incident Creation Rule** auswählst, wie in Abbildung 13-23 gezeigt.

## Azure Sentinel | Analytics

Selected workspace:

Search (Ctrl+ /)
[Create](#)
Refresh

General
Scheduled query rule

Overview
Microsoft incident creation rule

Logs
3

News & guides

Threat management
Active rules Rule templates

Incidents

Workbooks

Hunting

Notebooks

Entity behavior

Threat intelligence

Content management

Content hub (Preview)

Community

Configuration

Data connectors

Analytics
1

Severity ↑↓
Name

Severity ↑↓	Name
High	Advan
High	Create
High	Known

Abbildung 13-23: Erstellen von Microsoft Incident Creation Rules

Dadurch wird ein Assistent geöffnet, in dem du das Sicherheitsprodukt auswählst, auf dem du deine Regel basieren möchtest. Außerdem bietet er zusätzliche Filtermöglichkeiten. In diesem Bildschirm kannst du drei separate Filter konfigurieren:

- **Schweregrad:** Damit kannst du nur Warnungen einspeisen, die einen bestimmten Schwerpunkt im Quellprodukt haben. Das ist eine gute Möglichkeit, nur Vorfälle für "wichtige" Warnungen zu generieren.
- **Bestimmte Warnungen einbeziehen:** Wenn du nur für bestimmte Warnungen Vorfälle erstellen möchtest (z. B. mit Malware verknüpfte IP).
- **Bestimmte Warnungen ausschließen:** Wenn du einige (lärmende) Warnungen ausschließen möchtest (z. B. Ungewöhnliche Anmeldeeigenschaften).

**Vorfallerstellung vs. geplante Regeln:** Auch wenn die Microsoft Incident Creation Rules auf den ersten Blick sehr vielversprechend wirken, bringen sie doch einige Einschränkungen mit sich. Viele der Funktionen von geplanten Analyseregeln - wie zum Beispiel die Vorfallsgruppierung - sind nicht verfügbar. Mein persönlicher Vorschlag ist, die Incident Creation Rules zu verwerfen und stattdessen eigene geplante Analyseregeln zu erstellen, die die Tabelle SecurityAlert nach neuen Warnungen durchsuchen. Dadurch kannst du die volle Funktionalität der geplanten Regeln nutzen und die Daten aus der Tabelle SecurityAlert mit anderen Tabellen zur Anreicherung verknüpfen.

## Fusion

Fusion ist Microsofts System für maschinelles Lernen in Microsoft Sentinel. Fusion-Regeln kombinieren Ereignisse aus mehreren Quellen zu Vorfällen mit hoher Genauigkeit. Diese Art von Vorfällen wird produktübergreifend erkannt und als „mehrstufige Angriffe“ bezeichnet, da sie die verschiedenen Stufen des MITRE-ATT&CK-Frameworks zur Erkennung betrachten. Fusion ist eine Kombination verschiedener Erkennungen, die Microsoft ständig erweitert. Es gibt derzeit mehrere Instanzen von Fusion:

- **Szenariobasierte Fusion-Erkennungen** sind eine Liste aktueller Erkennungen, nach denen Fusion ständig sucht. Eine aktuelle Liste aller Erkennungen findest du in der [Microsoft-Dokumentation](#).
- **Fusion für neue Bedrohungen** erstellt Vorfälle basierend auf aktuellen Bedrohungen und Malware-Kampagnen, da es deine Daten anhand von Indikatoren aus diesen neuen Bedrohungen analysiert. Eine aktuelle Liste der Erkennungen ist nicht verfügbar, da sie sich ständig weiterentwickelt.
- **Fusion für Ransomware** kombiniert mehrere Warnungen aus verschiedenen Microsoft-Sicherheitsprodukten und identifiziert Ransomware-Angriffe mit hoher Zuverlässigkeit.

Die Fusion-Analyse „Advanced Multistage Attack Detection“ ist in jeder Microsoft Sentinel-Umgebung standardmäßig aktiviert. Sie kann in keiner Weise konfiguriert werden und muss so verwendet werden, wie sie ist. Es gibt keine Möglichkeit, die Priorität zu aktualisieren oder die Warnungsgruppierung zu konfigurieren. Auch wenn die Konfigurationsmöglichkeiten begrenzt sind, heißt das nicht, dass sie nicht nützlich sein kann. Da Fusion mit verschiedenen Datenquellen und Produkten arbeitet, analysiert es eine riesige Menge an Daten, die manuell nur sehr schwer zu untersuchen wären.

Auch wenn es nur eine Fusion-Analyse gibt, ist sie für die Erkennung aller unterschiedlichen Arten von Vorfällen verantwortlich. Fusion hat zwei Möglichkeiten, Erkennungen durchzuführen:

1. Die Erste besteht darin, die Rohdaten in deiner Umgebung zu untersuchen und nach spezifischen Erkennungen zu suchen, wie z. B. „unmögliche Reise an einen atypischen

Ort, die zu mehreren VM-Löschaktivitäten führt". Dabei werden vorhandene Warnungen über unmögliche Reisen mit Azure Activity-Daten kombiniert.

2. Die zweite Möglichkeit ist die Verwendung bestimmter bereitgestellter Analyseregel-Vorlagen, um Ereignisse aus andere Quellen zu kombinieren. Die Liste der verwendeten Regelvorlagen findest du [hier](#). Um bestimmte Fusion-Erkennungen zu verwenden, müssen diese Regeln zuerst aktiviert werden.

**Tipp:** Auch wenn die Analyseregel-Vorlagen für die Verwendung bestimmter Fusion-Erkennungen aktiviert sein müssen, heißt das nicht, dass du die Vorfällerstellung für sie aktivieren solltest. Einige dieser Analyseregeln generieren sehr viele Warnungen, so dass Vorfälle möglicherweise nicht erwünscht sind.

Für beide Fälle analysiert Fusion nur eine Teilmenge deiner Daten. Dafür müssen bestimmte Datenkonnektoren aktiviert sein. In der Microsoft-Dokumentation sind die spezifischen Datenquellen aufgeführt, die für die verschiedenen Arten von Erkennungen erforderlich sind.

Nach einem Microsoft Sentinel-Update im November 2021 hat Microsoft eine kleine Konfigurationsoberfläche bereitgestellt, mit der du die integrierten Fusion-Regeln anpassen kannst. Sie unterstützt zwei Konfigurationsmöglichkeiten:

- Du kannst auswählen, auf welche Datenquellen sich Fusion stützen darf, um Erkennungen zu erstellen und Vorfälle zu generieren. In der Benutzeroberfläche kannst du eine bestimmte Quelle ausschließen oder einbeziehen (standardmäßig sind alle Quellen aktiviert) oder Warnungsanbieter nach dem Schweregrad der von ihnen generierten Warnungen filtern.
- Wenn du mit einer großen Anzahl von falsch-positiven Ergebnissen von Fusion zu kämpfen hast, kannst du bestimmte Erkennungsmuster ausschließen. Das Hinzufügen von Erkennungen ist über diese Benutzeroberfläche nicht möglich und kann nur über die Vorfallsübersichtsseite eines Fusion-Vorfalls erfolgen. Nach dem Hinzufügen eines Ausschlusses wird dieser in der Benutzeroberfläche angezeigt. Diese Ausschlüsse werden derzeit auf eine durch Komma getrennte Weise gespeichert und können schwer zu lesen und zu interpretieren sein.

Da die Erkennungen vollständig von Microsoft verwaltet werden, gibt es hier nicht viel zu konfigurieren. Das kann sowohl ein Vor- als auch ein Nachteil sein. Der Vorteil ist, dass du keine zusätzliche Zeit investieren musst, um solche Produkte zu konfigurieren. Der Nachteil ist, dass du von Microsoft für diese Erkennungen abhängig bist. Es gibt keine Möglichkeit, die Erkennungen einzusehen oder den Schwellenwert oder bestimmte Signale zu aktualisieren. Du bist auch auf die Datenquellen und Analyseregel-Vorlagen beschränkt, die Microsoft bereitstellt. Ich empfehle, die Fusion-Regel aktiviert zu lassen, da sie einige starke Erkennungen liefern kann, die sonst möglicherweise unbemerkt geblieben wären. Auch wenn die Konfigurationsoberfläche für Fusion eine nette Ergänzung ist, solltest du sie nur ändern, wenn es wirklich notwendig ist. Ich habe selten falsch-positive Ergebnisse gesehen, die von Fusion

generiert wurden, und empfehle, nur bei Problemen mit vielen falsch-positiven Vorfällen Anpassungen vorzunehmen.

## SOC-ML

SOC-ML ist der zweite Typ von „Maschinelles Lernen“-basierter Regelart, die Anomalien in deiner Umgebung erkennt. Die Erkennungen werden von Microsoft erstellt, können aber angepasst werden. SOC-ML unterscheidet sich von Fusion, da es eine Vielzahl an Anomalien generieren kann, die nicht unbedingt eine hohe Genauigkeit aufweisen. Fusion hingegen generiert nur Vorfälle mit hoher Genauigkeit. SOC-ML ermöglicht es dir, vorhandene Anomalieregeln zu aktualisieren, um den Prozess des maschinellen Lernens zu beeinflussen.

SOC-ML generiert keine Vorfälle oder Warnungen, sondern protokolliert Anomalien in der Anomalietabelle. So kannst du entweder neue Analyseregeln auf Basis dieser Anomalien erstellen oder bestehende Warnungen mit zusätzlichen Anomaliedaten anreichern.

Anomalieregeln sind in jeder Microsoft-Sentinel-Umgebung standardmäßig aktiviert und werden kontinuierlich ausgeführt, um neue Anomalien in verschiedenen Datenquellen zu identifizieren. Die verwendeten Datenquellen variieren je nach Regel. Eine gute Übersicht erhältst du, wenn du zur Registerkarte *Analytics* navigierst und dort den Bereich *Anomalien* auswählst.

Auch wenn jede Regel standardmäßig aktiv ist, kannst du sie bei Bedarf deaktivieren oder löschen. Jede Regel bietet dir einige Konfigurationsoptionen, mit denen du sie an deine Anforderungen anpassen kannst. Das erfolgt über die Aktualisierung der jeweiligen Regelparameter. Eine SOC-ML-Anomalieregel kann sich in zwei Betriebsmodi befinden: *Production* und *Flighting*. Diese Zustände ermöglichen es dir, Parameteränderungen zu testen, ohne die produktiven Regeln zu beeinflussen. Die Parameter der integrierten Regeln lassen sich jedoch nicht direkt ändern. Um Anpassungen vorzunehmen, musst du eine eigene Instanz der Regel erstellen. Das machst du, indem du auf die drei Punkte neben einer aktiven Regel klickst und *Duplizieren* auswählst, wie in Abbildung 13-24 dargestellt. Dadurch wird eine neue Regel erstellt, die den Zusatz *Customized* im Namen trägt und standardmäßig deaktiviert ist, wie in der nächsten Abbildung zu sehen ist:



Abbildung 13-24: Duplizieren einer Anomalieregel

Wenn du die Regel bearbeitest, findest du zwei Registerkarten: **Allgemein** und **Konfiguration**. In der Registerkarte **Allgemein** kannst du zwei Einstellungen vornehmen:

- Aktualisiere den **Status** einer Regel. Wenn die Regel deaktiviert ist, werden keine Anomalien erstellt.
- Ändere den **Modus**. Du kannst die Regel von Flighting auf Produktion umstellen. Im Flighting-Modus werden zwar weiterhin Anomalien protokolliert, jedoch mit einem speziellen Status, sodass du testdaten gezielt herausfiltern kannst. Ist, protokolliert sie immer noch Anomalien, aber mit einem anderen Status. Eine Teilausgabe der Anomalietabelle ist in Abbildung 13-25 dargestellt. Dort siehst du unter anderem ein Regelmodus (Spalte RuleStatus), die Version und die Regel-ID.

AnomalyTemplateVersion	RuleId	RuleStatus
1.0.1	16d55bbb-8c54-	Production
1.0.2	2d3e33c6-d8e6-	Production

Abbildung 13-25: Beispieldaten der Anomalies-Tabelle

In der Registerkarte **Konfiguration** kannst du die Parameter und Schwellenwerte für die jeweilige Anomalieart anpassen. Abbildung 13-26 zeigt zum Beispiel die Konfiguration für „Anomale Benutzeraktivitäten in Office Exchange“. Dort findest du den Parameter **Top-Grund für hohen Anomalie-Score**, mit dem du festlegst, welche Ereignisse für die Regel berücksichtigt werden sollen.

Customize the definition of anomalous activity by configuring the following parameters.

### Parameters

Top reason for high anomaly score ⓘ

11 selected ⌂

- Distinct number of operations
- Distinct number of ClientInfoString
- Distinct number of mailboxes
- Number of operations
- Distinct number of IP addresses
- Distinct number of /24 subnets
- Distinct number of /16 subnets
- Number of MessageBind operations
- Number of FolderBind operations
- Number of UpdateInboxRules operations
- Number of MailItemsAccessed operations

### Thresholds

Score ⓘ



Abbildung 13-26: Konfiguration einer Anomalieregel

Neben den Parametern kannst du auch den Schwellenwert konfigurieren. Jede Anomalie hat einen Score, der angibt, wie wahrscheinlich es ist, dass es sich um eine tatsächliche Anomalie handelt. Wenn du den Schwellenwert erhöhst, werden nur noch Anomalien mit einem entsprechend hohen Score erfasst – das reduziert die Anzahl der erkannten Ereignisse.

Das gezielte Anpassen des Schwellenwerts ist eine gute Möglichkeit, um deine Anomalieregeln zu optimieren und dich vor einer Überflutung mit unwichtigen Ergebnissen zu schützen. Die unten gezeigte KQL-Beispielabfrage ermöglicht es dir, alle Anomalien abzurufen, die sich derzeit im *Flighting*-Modus befinden – inklusive ihrer Scores.

### Anomalies

```
| where RuleStatus == "Flighting"
| project-reorder RuleName, Score
```

Auch wenn SOC-ML anfangs etwas komplex wirkt, bietet es dir eine effektive Möglichkeit, die Rechenleistung der Cloud zu nutzen, um dein Security Operations Center (SOC) zu verbessern. Es wird empfohlen, zunächst mit den Standardkonfigurationen zu arbeiten und die Menge der

eingehenden Anomalien zu beobachten. Anschließend kannst du eigene Regeln im *Flighting*-Modus testen und gezielt Parameter oder Schwellenwerte anpassen. Sobald du mit deiner benutzerdefinierten Konfiguration zufrieden bist, kannst du sie in die Produktion übernehmen und die Standardregel deaktivieren.

**Hinweis:** Microsoft sendet deine Daten durch seine Machine-Learning-Pipeline, um Anomalien zu erzeugen. Das bedeutet, dass deine Daten den Log Analytics-Arbeitsbereich verlassen und verschlüsselt in die Pipeline eingespeist werden. Wenn du einen kundenverwalteten Schlüssel für deinen Log Analytics-Arbeitsbereich verwendest, hat das keine Auswirkungen - die Pipeline arbeitet mit einem von Microsoft verwalteten Schlüssel. Wenn du das nicht möchtest, kannst du SOC-ML vollständig deaktivieren, indem du unter **Einstellungen > Anomalien > Aus** navigierst.

## ML-Verhaltensanalyse

Machine Learning Behavioral Analytics-Regeln sind die dritte und letzte Art von Analyseregeln auf Basis von maschinellem Lernen in Microsoft Sentinel. Du kannst sie als Kombination aus Fusion und SOC-ML verstehen. Derzeit gibt es zwei ML-Regeln, die als Regelvorlagen verfügbar sind, beide basieren auf Sicherheitsereignissen:

1. Anomale SSH-Anmeldeerkennung
2. Anomale RDP-Anmeldeerkennung

Diese Analyseregeln findest du in den Regelvorlagen und kannst sie mit wenigen Klicks aktivieren. Sie verwenden eine von Microsoft entwickelte Erkennungslogik und lassen sich nicht bearbeiten. Beide Regeln haben eine Lernphase von dreißig Tagen und erstellen Vorfälle, wenn ein Treffer auf Basis der ML-Modelle von Microsoft erkannt wird.

**Hinweis:** ML Behavioral Analytic-Regeln wurden erstmals 2020 in der öffentlichen Vorschau eingeführt. Seitdem hat sich wenig verändert, und die Funktion befindet sich weiterhin in der Vorschau. SOC-ML wurde ein Jahr später eingeführt. Es ist zu erwarten, dass ML Behavior-Regeln langfristig zugunsten von SOC-ML-Anomalien eingestellt werden.

## Wie man mit Analyseregeln beginnt

Beim Aufbau deiner ersten Microsoft Sentinel-Umgebung kann es schwerfallen, eigene Analyseregeln zu erstellen. Der Einstieg in komplexe KQL-Abfragen wirkt anfangs vielleicht abschreckend, doch mit etwas Übung wirst du dich schnell zurechtfinden. Wenn du neu bei Microsoft Sentinel bist, empfiehlt es sich, mit den integrierten Analyseregel-Vorlagen zu beginnen. Indem du die für dich relevantesten Vorlagen aktivierst, erhältst du Vorfälle auf Grundlage von Regeln, die von Microsoft und der Community entwickelt wurden. So lernst du, welche Arten von Vorfällen generiert werden und wie du sie untersuchen kannst. Während

dieser Untersuchungen kannst du erste Erfahrungen mit KQL sammeln, indem du die zugrunde liegenden Daten analysierst. Mit der Zeit wirst du feststellen, dass die Vorlagen nicht zu hundert Prozent zu deinen Anforderungen passen. Dann ist der richtige Moment gekommen, um sie schrittweise anzupassen. Durch die Aktualisierung bestehender Regeln vertiefst du dein Verständnis für die Abfragesprache und baust dein Wissen weiter aus. Schon bald wirst du in der Lage sein, eigene einfache Analyseregeln zu erstellen – und bald darauf auch komplexere Abfragen zu schreiben, mit denen du gezielt Warnungen für deine Umgebung generieren kannst.

Zu Beginn solltest du unbedingt die Microsoft Incident Creation Rules, Fusion, SOC-ML und ML Behavioral Analytics aktivieren. Diese erfordern kaum Konfiguration und ermöglichen dir, die Cloud effektiv zu nutzen, um deine Umgebung zu schützen. Sobald du mehr Erfahrung gesammelt hast, kannst du die Microsoft Incident Creation Rules durch benutzerdefinierte Regelvorlagen ersetzen und bestehende SOC-ML-Regeln anpassen, um sie besser auf deine Umgebung zuzuschneiden.

## Hunting mit Microsoft Sentinel

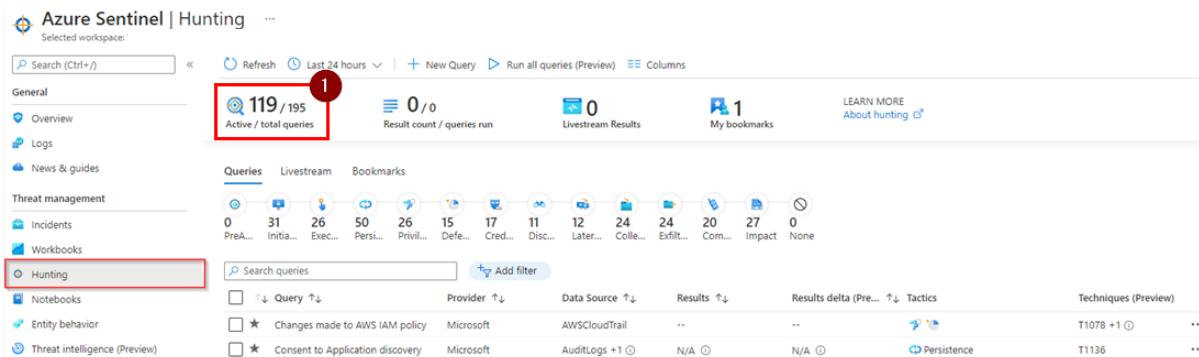
Auch wenn Analyseregeln eine hilfreiche Möglichkeit sind, um potenzielle Sicherheitsverstöße zu identifizieren, reicht das allein nicht aus. Du musst proaktiv nach Anzeichen für Angriffe suchen. Verlass dich nicht ausschließlich auf die integrierten Regeln in Microsoft Sentinel oder anderen Sicherheitslösungen – sie können nicht jede bösartige Aktivität erkennen.

Ein Beispiel ist der Solarwinds-Angriff. Viele frühe Phasen dieses Angriffs – insbesondere die Nutzung bösartiger Solarwinds-DLLs zur Infiltration des Netzwerks – blieben unbemerkt, weil sie auf Techniken beruhten, die vielen Sicherheitsteams nicht bekannt waren. Obwohl der Angriff ungewöhnliche Muster zeigte, wurden diese nicht von den integrierten Analyseregeln erkannt. Selbst wenn es theoretisch möglich wäre, für jede bekannte Angriffstechnik eine Regel zu entwickeln, ist das in der Praxis kaum umsetzbar. Daher ist es entscheidend, dass du deine Daten regelmäßig und aktiv nach Auffälligkeiten durchsuchst. So kannst du laufende Angriffe rechtzeitig erkennen und die Bedrohung schnellstmöglich eindämmen. Diese proaktive Suche – auch „Hunting“ genannt – ist keine exklusive Funktion von Microsoft Sentinel. Sie steht dir auch in anderen Produkten wie Microsoft 365 Defender zur Verfügung, worauf in Kapitel 9 näher eingegangen wird.

In Microsoft Sentinel kannst du die Hunting-Funktionen nutzen, um gezielt mit KQL-Abfragen nach potenziellen Sicherheitsverstößen zu suchen. Diese Funktionen eignen sich nicht nur für die proaktive Suche, sondern unterstützen dich auch bei laufenden Untersuchungen. Mit den richtigen Hunting-Abfragen kannst du deine Analyse deutlich effizienter gestalten.

# Hunting über das Portal

Im Azure-Portal findest du ein eigenes Blade, das dem Hunting gewidmet ist.



The screenshot shows the Microsoft Sentinel Hunting blade. At the top left, there's a search bar and navigation buttons for Refresh, Last 24 hours, New Query, Run all queries (Preview), and Columns. Below that is a summary section with '119 / 195 Active / total queries' (marked with a red box and a circled '1'), '0 / 0 Result count / queries run', '0 Livestream Results', and '1 My bookmarks'. To the right is a 'LEARN MORE About hunting' link. On the left, there's a sidebar with sections like General (Overview, Logs, News & guides), Threat management (Incidents, Workbooks, Hunting - highlighted with a red box), Notebooks, Entity behavior, and Threat intelligence (Preview). The main area has tabs for Queries, Livestream, and Bookmarks. Under Queries, there's a grid of icons representing different attack phases: Pre-, Initia..., Exec..., Persi..., Privil..., Defe..., Cred..., Disc..., Later..., 12, 24, 24, 20, 27, and None. Below this is a table with columns for Query, Provider, Data Source, Results, Results delta, Tactics, and Techniques (Preview). Two rows are visible: one for 'Changes made to AWS IAM policy' (Microsoft, AWSCloudTrail, T1078 +1) and another for 'Consent to Application discovery' (Microsoft, AuditLogs +1).

Abbildung 13-27: Hunting im Microsoft Sentinel-Portal

Microsoft Sentinel bietet dir drei zentrale Hunting-Funktionen: Abfragen, Livestream und Lesezeichen.

## Abfragen

Hunting-Abfragen sind KQL-basierte Abfragen, mit denen du gezielt Informationen in deiner Umgebung identifizieren kannst. Microsoft stellt dir hierfür eine Vielzahl von Abfragen bereit, die du ausführen kannst, um mögliche Treffer zu überprüfen.

Am oberen Rand des Hunting-Blades siehst du eine Übersicht zur Anzahl aktiver und insgesamt verfügbarer Abfragen. Die Gesamtzahl ergibt sich aus allen gespeicherten Hunting-Abfragen in deiner Umgebung. Eine Abfrage gilt als aktiv, wenn die benötigten Datenkonnektoren in deiner Umgebung aktiviert sind.

Ein genauerer Blick auf die Benutzeroberfläche der Abfragen zeigt dir eine strukturierte Übersicht über die MITRE ATT&CK-Taktiken. In der Übersicht (siehe Nummer 1 in Abbildung 13-28) erkennst du, wie viele Hunting-Abfragen für jede Angriffsphase vorhanden sind. Das ist aus mehreren Gründen hilfreich:

- Identifizierte, für welche Angriffsstufen du die wenigsten Regeln hast. Dieses Wissen kannst du nutzen, um deine Bibliothek mit Hunting-Regeln gezielt zu erweitern und solche Regeln zu erstellen, die dir helfen, verschiedene Angriffstechniken zu erkennen.
- Wenn du nach einer bestimmten Aktivität suchst - z. B. Datenexfiltration-, kannst du die passende Hunting-Regeln ganz einfach herausfiltern und für deine aktuelle Untersuchung nutzen.

Abbildung 13-28: Übersicht der Hunting-Regeln

In der unteren Hälfte der Seite findest du eine Übersicht aller aktuellen Hunting-Regeln. Da alle anfänglich von Microsoft bereitgestellt werden (wie in der Spalte „Anbieter“ ersichtlich), lohnt es sich, alle Vorlagen durchzusehen und zu prüfen, welche davon für dich sinnvoll sind. Beachte, dass Vorlagen-Hunting-Regeln weder gelöscht noch bearbeitet werden können – sie bleiben dauerhaft in deiner Umgebung sichtbar.

Auf der Übersichtsseite kannst du die angezeigten Spalten anpassen oder die Ergebnisse mithilfe integrierter Filter wie Datenquelle, Taktik oder Technik filtern. Neben dem Namen der Abfrage und dem Anbieter gibt es noch weitere nützliche Spalten:

- **Datenquelle** zeigt dir, welche Quelle von der Abfrage verwendet wird. Das hilft dir zu erkennen, ob eine Abfrag auf Daten basiert, die in deiner Microsoft Sentinel-Umgebung überhaupt verfügbar sind.
- **Ergebnisse** und **Ergebnisdelta** geben dir nach Ausführung einer Abfrage einen Überblick darüber, wie viele Treffer es gab und ob sich diese Anzahl im Vergleich zum letzten Lauf verändert hat.
- **Taktiken** und **Techniken** helfen dir, die jeweilige Angriffsphase zu identifizieren. Eine Taktik kann mehrere Techniken enthalten, die beschreiben, wie ein Angriff durchgeführt wird. Ein Beispiel: Phising ist eine Technik innerhalb der Taktik "Initialer Zugriff". Eine vollständige Übersicht findest du auf der [offiziellen MITRE-Website](#). Bei Microsoft-Abfragen sind Techniken oft bereits definiert – bei benutzerdefinierten Abfragen kannst du sie jedoch nicht selbst festlegen.
- **Entitäten** zeigen dir, welche Art von Entitäten die Abfrage zurückgibt. Manche Abfragen suchen nach IP-Adressen, andere nach Benutzerkonten oder Hosts.

Jede Regel bietet dir verschiedene Aktionen. Wenn du auf eine bestimmte Regel klickst, öffnet sich auf der rechten Seite des Bildschirms eine Detailansicht. Dort findest du zusätzliche Informationen zur Abfrage, etwa eine Beschreibung und einen Ausschnitt des Regelinhalts. Am unteren Rand kannst du die Abfrage ausführen oder Ergebnisse anzeigen. Nach dem Ausführen wird die Spalte „Ergebnisse“ aktualisiert und zeigt dir die Anzahl der Treffer.

Wenn du auf die drei Punkte ganz rechts neben jeder Regel klickst, stehen dir zusätzliche Aktionen zur Verfügung:

- **Abfrage ausführen:** Die Abfrage wird im Hintergrund ausgeführt, die Spalte "Ergebnisse" wird aktualisiert. Um die Treffer im Detail zu sehen, öffne die Abfrage und klicke auf die Ergebnisse anzuzeigen, klicke auf die Abfrage, um die Detailseite zu öffnen, und klicke auf **Ergebnisse anzeigen**.
- **Zu Favoriten hinzufügen:** Das Hinzufügen einer Regel zu deinen Favoriten hat zwei Hauptvorteile: Sie wird oben in der Liste der Hunting-Regeln angezeigt, so dass du wichtige Regeln leicht erkennen und ausführen kannst. Außerdem werden die Hunting-Regeln, die du zu deinen Favoriten hinzugefügt hast, automatisch ausgeführt, wenn du das Hunting-Blade aufrufst. Das spart dir Zeit, indem du nicht jede Regel manuell ausführen musst.
- **Abfrage bearbeiten:** Hier kannst du die Konfiguration einer Regel ändern, etwa die zugrunde liegende KQL-Abfrage, zugeordnete Taktiken oder Entitäten. Dies ist jedoch nur möglich, wenn die Regel nicht von Microsoft stammt.
- **Abfrage klonen:** Eine praktische Möglichkeit, um eine bestehende Regel – auch eine von Microsoft – als Vorlage zu nutzen. Die kopierten Inhalte werden in den Assistenten zum Erstellen benutzerdefinierter Regeln übernommen. So kannst du schnell neue, angepasste Regeln erstellen.
- **Abfrage löschen:** Damit entfernst du eine Abfrage aus deinem Arbeitsbereich. Diese Aktion ist dauerhaft und steht bei Microsoft Regeln nicht zur Verfügung
- **Zum Livestream hinzufügen:** So fügst du die Regel zur Livestream-Funktion hinzu – diese wird im nächsten Abschnitt erläutert.
- **Anlyseregel erstellen:** Du kannst eine Hunting-Abfrage in eine Analyseregel umzuwandeln, etwa wenn sie regelmäßig sicherheitsrelevant Ereignisse liefert.

Auch wenn bereits viele Abfragevorlagen zur Verfügung stehen, empfiehlt es sich, eigene Abfragen zu erstellen, die speziell auf deine Organisation zugeschnitten sind. Über die Schaltfläche „**Neue Abfrage**“ oben auf der Seite öffnest du den Assistenten zur Erstellung benutzerdefinierter Regeln.

Im Gegensatz zu geplanten Analyseregeln gibt es hier nur wenige Dinge zu konfigurieren, da keine Warnungen oder Vorfälle erzeugt werden. Du gibst lediglich Folgendes an:

- Name
- Beschreibung
- Abfrage
- Entitätszuordnung
- Taktik

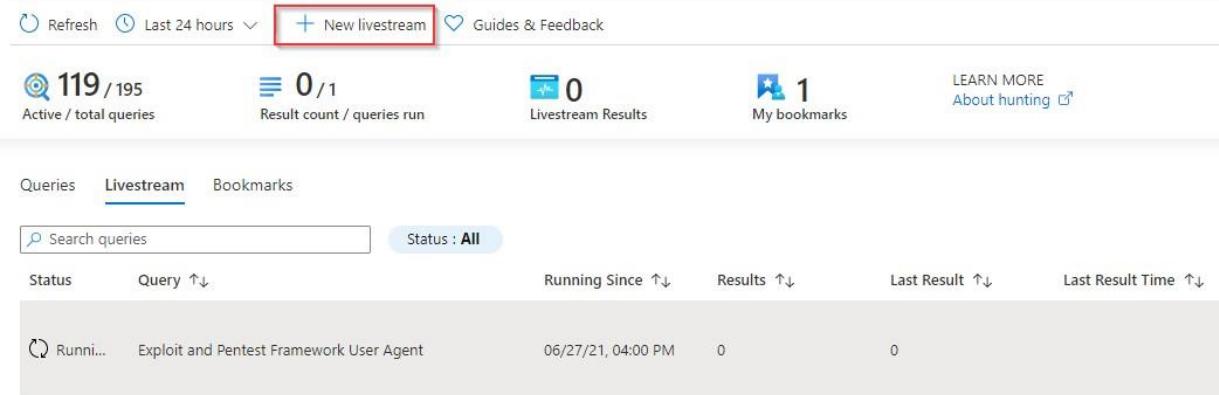
Das Erstellen von Hunting-Abfragen hat keine direkten Auswirkungen auf deine Umgebung. Bei Analyseregeln musst du darauf achten, nicht zu viele Fehlalarme zu erzeugen. Hunting-Abfragen dagegen erzeugen keine Warnungen – deshalb rate ich dir, möglichst viele davon zu

erstellen. So gewinnst du Einblicke in deine Daten und nutzt gleichzeitig die vollen Möglichkeiten von Microsoft Sentinel. Auch wenn du nicht jeden Tag jede Abfrage verwendest, können sie bei bestimmten Untersuchungen später sehr nützlich sein.

## Livestream

Da Hunting-Abfragen manuell ausgeführt werden müssen und keine automatische Benachrichtigung bieten, bietet Microsoft Sentinel zusätzlich die Funktion „Hunting Livestream“. Sie liefert dir nahezu in Echtzeit Rückmeldungen, wenn Treffer für bestimmte Hunting-Abfragen erkannt werden.

Um diese Funktion zu nutzen, musst du eine oder mehrere Abfragen als Livestream-Abfragen aktivieren. Das machst du, indem du eine Hunting-Abfrage öffnest und im Portal auf „**Neuer Livestream**“ klickst – wie in Abbildung 13-29 dargestellt.



The screenshot shows the Microsoft Sentinel interface for managing hunting queries. At the top, there are navigation links for 'Refresh', 'Last 24 hours', and a red-highlighted button '+ New livestream'. To the right are links for 'Guides & Feedback', '119 / 195 Active / total queries', '0 / 1 Result count / queries run', '0 Livestream Results', '1 My bookmarks', and 'LEARN MORE About hunting'. Below this, there are tabs for 'Queries', 'Livestream' (which is selected), and 'Bookmarks'. A search bar 'Search queries' contains the placeholder 'Exploit and Pentest Framework User Agent'. A status filter 'Status : All' is applied. The main table lists one query: 'Exploit and Pentest Framework User Agent' with a status of 'Running Since 06/27/21, 04:00 PM', 0 results, and 0 last results.

Abbildung 13-29: Hinzufügen eines Livestreams

Durch das Hinzufügen von Regeln zum Livestream werden diese kontinuierlich ausgeführt und du kannst alle Treffer überwachen, sobald sie eintreffen. Wenn ein Treffer gefunden wird, erhältst du eine Benachrichtigung im Azure-Portal.

Livestream liegt funktional zwischen Hunting-Abfragen und Analyseregeln. Die Regeln generieren keine Warnungen oder Vorfälle und belasten deine Warteschlange nicht. Du wirst jedoch benachrichtigt, sobald ein Treffer gefunden wird – im Gegensatz zu klassischen Hunting-Abfragen, die keine Benachrichtigung bieten. Das ermöglicht einige interessante Anwendungsfälle:

- Wenn du eine bestimmte Bedrohung in deiner Umgebung untersuchst, kannst du einen Livestream erstellen, um benachrichtigt zu werden, sobald ein Hinweis auf diese Bedrohung oder Aktivität erkannt wird.
- Wenn du proaktiv nach Bedrohungen suchst, kannst du Livestreams nutzen, um einen verdächtigen Host oder eine IP-Adresse in Echtzeit zu überwachen.

Sobald ein Treffer für einen Livestream gefunden wird, erscheint eine Benachrichtigung. Wenn du darauf klickst, wirst du direkt zu den Ergebnissen der Abfrage weitergeleitet. Von dort kannst du das Ereignis bei Bedarf zu einer Warnung eskalieren.

## Lesezeichen

Während deiner Bedrohungssuche kann es vorkommen, dass du auf bestimmte Ereignisse stößt, zu denen du später zurückkehren möchtest. Dafür kannst du spezifische Ereignisse mit einem Lesezeichen versehen – besonders hilfreich, wenn du Protokolle über mehrere Datenquellen hinweg analysierst. Mit Lesezeichen kannst du Ereignisse markieren und später in einer kombinierten Ansicht betrachten.

Ein Ereignis kannst du mit einem Lesezeichen versehen, indem du im Abfragebildschirm das Kontrollkästchen links neben dem betreffenden Protokolleintrag aktivierst und dann auf „Lesezeichen hinzufügen“ klickst, wie in Abbildung 13-30 gezeigt. Anschließend öffnet sich der Assistent „**Lesezeichen hinzufügen**“, in dem du weitere Informationen wie Notizen, Tags und Entitäten eintragen kannst.

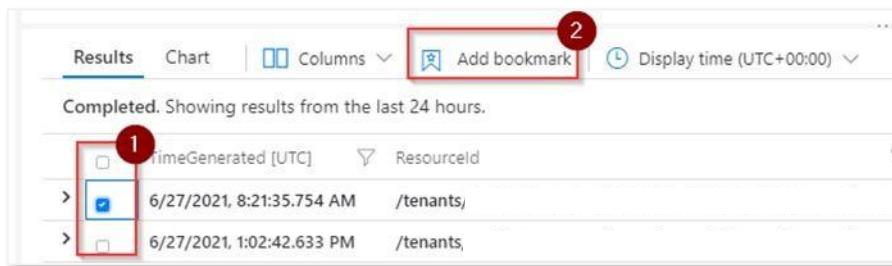
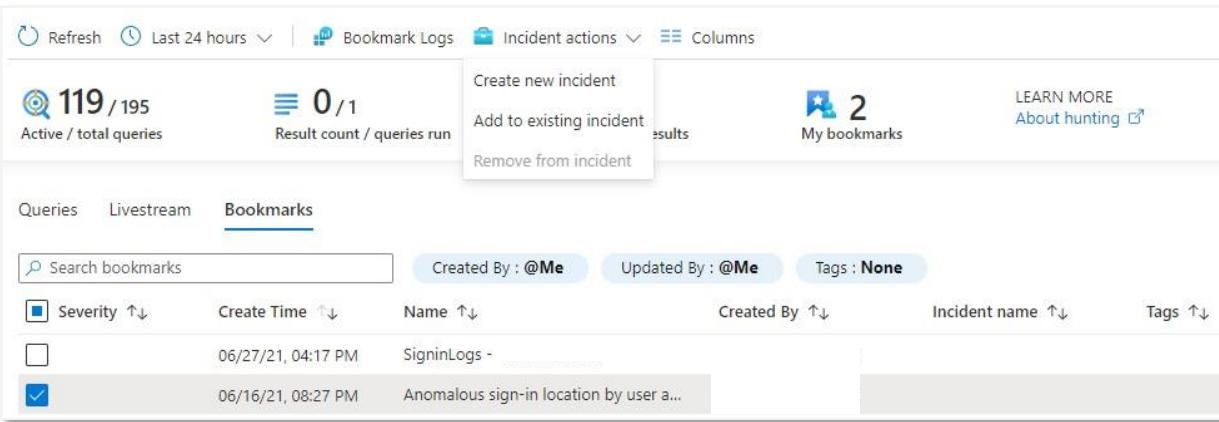


Abbildung 13-30: Hinzufügen eines Lesezeichens

Sobald du ein oder mehrere Ereignisse mit einem Lesezeichen versehen hast, findest du sie auf der Registerkarte „Lesezeichen“ im Hunting-Bereich des Microsoft Sentinel-Portals. Dort stehen dir verschiedene Aktionen zur Verfügung:

- Du kannst ein einzelnes Lesezeichen weiter untersuchen, indem du die Detailansicht öffnest und auf die Schaltfläche **Untersuchen** klickst. Dadurch öffnet sich der Untersuchungsbereich, der im Abschnitt "Auf Vorfälle mit Microsoft Sentinel reagieren" behandelt wird.
- Du kannst Lesezeichen mit neuen oder bestehenden Vorfällen verknüpfen (wie in Abbildung 13-31 unten zu sehen). Das ermöglicht dir, einen neuen Vorfall auf Basis eines gefundenen Ereignisses zu erstellen – hilfreich, wenn du tiefergehende Analysen durchführen möchtest. Ebenso kannst du einem bestehenden Vorfall zusätzliche Informationen hinzufügen, indem du das Lesezeichen mit diesem verknüpft. Das ist besonders nützlich, wenn du weitere Protokolldaten zu einem laufenden Vorfall hinzufügen möchtest.



The screenshot shows the Microsoft Sentinel search interface. At the top, there are navigation links: Refresh, Last 24 hours, Bookmark Logs, Incident actions, and Columns. Below these are two summary cards: "119 / 195 Active / total queries" and "0 / 1 Result count / queries run". To the right of these cards is a context menu with options: Create new incident, Add to existing incident, Remove from incident, and Results. Further to the right are links to "My bookmarks" (2 results) and "LEARN MORE About hunting". Below the summary cards, there are tabs for Queries, Livestream, and Bookmarks, with Bookmarks being the active tab. A search bar labeled "Search bookmarks" is followed by filters: Created By : @Me, Updated By : @Me, and Tags : None. The main table lists bookmarks with columns for Severity, Create Time, Name, Created By, Incident name, and Tags. Two rows are visible: one for "SigninLogs" (severity low, created 06/27/21, updated 06/27/21) and another for "Anomalous sign-in location by user a..." (severity medium, created 06/16/21, updated 06/16/21).

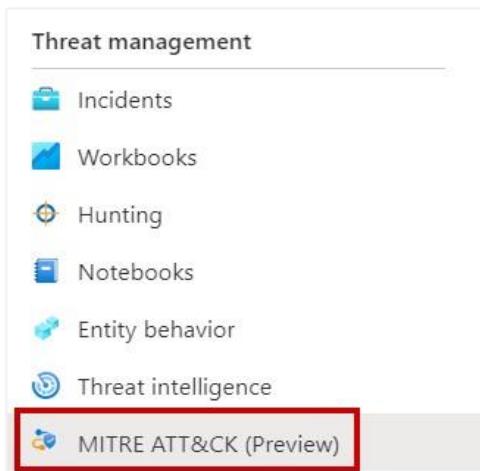
Abbildung 13-31: Interaktion mit Vorfällen während der Suche

Lesezeichen können entfernt werden, indem du auf die drei Punkte auf der rechten Seite klickst und "Lesezeichen entfernen" auswählst.

## MITRE ATT&CK-Übersicht

Jede Analyseregel und jede Hunting-Abfrage hat eine bestimmte Reihe von MITRE ATT&CK-Taktiken und -Techniken konfiguriert. Wenn ein Vorfall auf Grundlage einer Regel erstellt wird, übernimmt auch dieser Vorfall diese Taktiken und Techniken als Eigenschaften. Das gibt dir als SOC-Analyst die Möglichkeit zu erkennen, in welcher Phase eines Angriffs sich der Vorfall befinden könnte.

Innerhalb von Microsoft Sentinel gibt es ein eigenes Blade, das sich vollständig der Abdeckung des MITRE ATT&CK-Frameworks widmet. Du findest es in der Kategorie „Bedrohungsmanagement“ unter „Bedrohungsinformationen“, wie in Abbildung 13-32 gezeigt.



The screenshot shows the Azure portal navigation menu under the "Threat management" category. The menu items are: Incidents, Workbooks, Hunting, Notebooks, Entity behavior, Threat intelligence, and MITRE ATT&CK (Preview). The "MITRE ATT&CK (Preview)" item is highlighted with a red rectangle.

Abbildung 13-32: Das MITRE ATT&CK-Blade im Azure-Portal

Dieses Blade bietet dir eine visuelle Übersicht über die Abdeckung deiner aktuellen Microsoft Sentinel-Bereitstellung. Es betrachtet die aktuell konfigurierten Analyseregeln und Hunting-Abfragen und deren jeweilige MITRE ATT&CK-Zuordnung. Ziel dieses Blades ist es, dir aufzuzeigen, wo du potenzielle Lücken in deinen Erkennungen hast – etwa dann, wenn für eine bestimmte Taktik keine Regeln vorhanden sind. Bevor du diesen Bericht zur Bewertung deiner SIEM-Umgebung nutzt, möchte ich dir zwei wichtige Hinweise geben:

- Dieser Bericht funktioniert nur korrekt, wenn alle relevanten MITRE ATT&CK-Taktiken und -Techniken ausgefüllt wurden. Zwar enthalten die meisten Regelvorlagen diese diese Informationen, allerdings wurden Techniken erst Mitte 2022 eingeführt. Das bedeutet, dass einige deiner Ressourcen unter Umständen noch nicht vollständig konfiguriert sind. Es ist daher sinnvoll, deine Umgebung zu überprüfen und sicherzustellen, dass alles auf dem aktuellen Stand ist.
- Viele Organisationen nutzen EDR- und XDR-Tools, die Erkennungen "out of the box" liefern, um Geräte und Netzwerke zu schützen. Solche Erkennungen werden nicht als Regel in deinem SIEM eingerichtet, sondern als Vorfälle in Microsoft Sentinel synchronisiert. Daher sind sie in der MITRE ATT&CK-Abdeckungsübersicht nicht enthalten, weil sie nicht direkt in Microsoft Sentinel generiert wurden. Das ist besonders wichtig zu Wissen, wenn du den Bericht deinem Management präsentierst: Er zeigt nur die Abdeckung durch Regeln in Microsoft Sentinel und nicht die vollständige SOC-Abdeckung.

Trotz dieser Einschränkungen kann der Bericht sehr nützlich sein, wenn du ihn richtig einsetzt. Er zeigt dir auch, wie deine Abdeckung aussehen würde, wenn du alle verfügbaren Vorlagen für Analyseregeln und Hunting-Abfragen aktivieren würdest. So kannst du simulieren, welche Regeln den größten Einfluss auf eine möglichst vollständige Abdeckung deiner Umgebung hätten.

## User Entity Behavior Analytics (UEBA)

Auch wenn du mit Analyseregeln und Hunting-Abfragen viele potenzielle Sicherheitsverstöße erkennen kannst, haben viele SOC-Teams das Problem, dass ihnen die Zeit fehlt, um anomales Verhalten umfassend zu untersuchen. Hier kommen maschinelles Lernen und künstliche Intelligenz ins Spiel. Ein Computermodell kann darauf trainiert werden, außergewöhnliche Ereignisse zu identifizieren – auch solche, die ein SOC-Team übersehen könnte oder für deren Analyse nicht genug Zeit bleibt.

Als cloudbasiertes SIEM bietet dir Microsoft Sentinel mehrere Vorteile beim Einsatz von ML und KI. Maschinelles Lernen erfordert viel Rechenleistung, um komplexe Algorithmen auszuführen. Dank der Skalierbarkeit der Cloud musst du keine teure Serverinfrastruktur bereitstellen. Azure stellt leistungsstarke ML-Ressourcen bereit, die du bedarfsgerecht nutzen kannst – und du zahlst nur für die Zeit, in der du sie tatsächlich verwendest. So musst du keine hohen Investitionen tätigen, um mit ML-Modellen zu arbeiten.

Microsoft als einer der größten Softwarehersteller der Welt verfügt über zahlreiche Cloud-Produkte, die riesige Datenmengen erfassen – etwa durch Windows, Microsoft Defender oder Azure. Der Zugriff auf diese globalen Datenquellen ermöglicht es Microsoft, weltweit auftretende Bedrohungen zu analysieren und sehr schnell darauf zu reagieren.

Wie bereits erwähnt, bietet Microsoft Sentinel mehrere ML-Funktionen. Einige davon wurden bereits im Abschnitt „Analyseregeln“ behandelt – darunter Fusion, SOC-ML und ML Behavior Analytics. Eine weitere Funktion ist UEBA: User Entity Behavior Analytics. UEBA analysiert Entitäten in deiner Umgebung und erkennt, welche Ereignisse in Bezug auf diese Entitäten ungewöhnlich sind. Außerdem identifiziert es Beziehungen zwischen Entitäten – so kannst du nachvollziehen, welche Interaktionen im Normalfall auftreten und welche davon abweichen.

## Aktivieren von UEBA

UEBA ist standardmäßig nicht aktiv. Um es zu verwenden, musst du es zunächst aktivieren. Navigiere dazu zu den **Einstellungen** und wähle „**UEBA konfigurieren**“, wie in Abbildung 13-33 gezeigt.

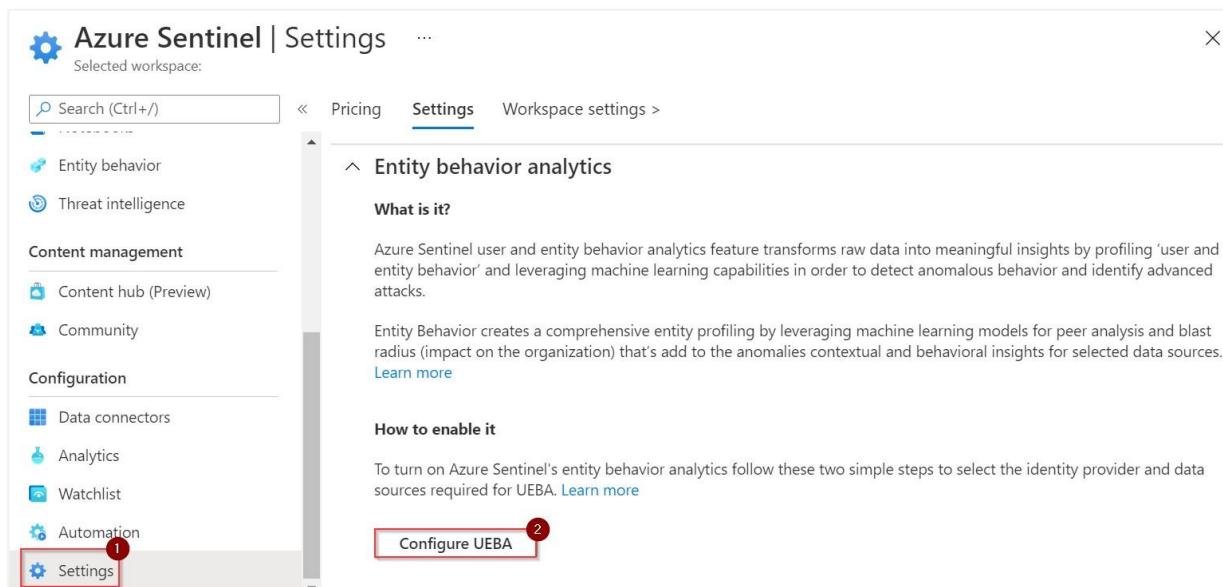


Abbildung 13-33: Aktivieren von UEBA

Durch Klicken auf „**UEBA konfigurieren**“ öffnet sich die UEBA-Konfigurationsseite, auf der du auswählen musst, aus welchem Verzeichnisdienst du Daten synchronisieren möchtest und für welche Datenquellen UEBA aktiviert werden soll. Die Synchronisierung von Daten ist sowohl aus Active Directory als auch aus Entra ID möglich, wobei die Einspeisung von Active-Directory-Daten eine Microsoft Defender for Identity-Bereitstellung erfordert.

Die verschiedenen unterstützten Datenquellen sind derzeit ziemlich begrenzt. Nur Datenquellen, die für UEBA aktiviert sind, werden analysiert. Die Liste der unterstützten Datenkonnektoren wird voraussichtlich wachsen, aber es wird wahrscheinlich eine Weile dauern, bis auch Protokolle von Drittanbietern unterstützt werden.

**Hinweis:** UEBA kann während der Untersuchung einer Warnung oder eines Vorfalls sehr nützlich sein, da es zusätzliche Informationen liefert und Kontext über die beteiligten Entitäten bereitstellt. All dies wird im Abschnitt "Auf Vorfälle mit Microsoft Sentinel reagieren" behandelt.

## UEBA-Daten

Nachdem du UEBA aktiviert hast, werden vier zusätzliche Tabellen in deiner Umgebung erstellt. Diese Tabellen enthalten Informationen aus Entra ID, sowie angereicherte Daten aus der UEBA-Engine. Jede Tabelle umfasst eine bestimmte Art von Informationen. Eine vollständige Übersicht über alle Tabellen und ihre Felder findest du in der [offiziellen Dokumentation](#).

- **BehaviorAnalytics:** Enthält angereicherte Informationen aus den aktivierten Datenquellen und kombiniert Informationen über die Aktion (z. B. Benutzeranmeldung, erstellte VM) mit Angaben zum Benutzer. Dazu gehört, ob die Aktion für diesen Benutzer ungewöhnlich ist, ob die IP-Adresse, die für die Anmeldung verwendet wurde, zuvor vom Benutzer genutzt wurde und ob seine Kollegen die selbe IP ebenfalls verwenden.
- **IdentityInfo:** Kopiert Benutzerdaten aus Entra ID und AD, die zur Anreicherung in Analyseregeln oder bei der Suche verwendet werden können. Sie enthält sehr nützliche Informationen wie die AccountObjectID, den Status (aktiviert/deaktiviert), Gruppenmitgliedschaften sowie allgemeine Entra ID-Daten wie Standort, Abteilung und Berufsbezeichnung.
- **UserAccessAnalytics:** Enthält Informationen über den administrativen Zugriff, den ein Benutzer oder Dienstprinzipal auf bestimmte Ressourcen hat. Derzeit ist dies auf Azure-Ressourcen beschränkt; Entra ID-Rollen sind nicht enthalten. So kannst du erkennen, welche Benutzer Berechtigungen für bestimmte Azure-Ressourcen haben, das kann bei der Anreicherung bestehender Abfragen hilfreich sein.
- **UserPeerAnalytics:** UEBA analysiert Sicherheitsgruppen von Benutzer und verwendet diese Daten, um Beziehungen zwischen den Benutzern innerhalb der Organisation zu identifizieren. Die Tabelle enthält einen Eintrag für jeden Benutzer im Mandanten und dessen Peers. Jeder Eintrag hat einen Rang – je niedriger der Rang, desto enger die Verbindung zwischen den beiden Peers.

Wenn du gerade erst mit Microsoft Sentinel und UEBA beginnst, kann es ziemlich verwirrend sein, durch die verschiedenen Rohdaten und Tabellen zu navigieren. Microsoft stellt ein Workbook namens „**User and Entity Behavior Analytics**“ bereit, das dir eine gute Einführung in die aktuellen Anomalien bietet, die von UEBA identifiziert wurden.

**Hinweis:** UEBA ist allein für die Befüllung von Daten in den genannten Tabellen verantwortlich und du als Administrator kannst keine eigenen Informationen hinzufügen. Das bedeutet, dass du bei der Einspeisung vollständig von Microsoft abhängig bist. Leider werden diese Tabellen nicht häufig aktualisiert. Die Tabelle **IdentityInfo** z. B. kann bis zu 7 Tage für eine Aktualisierung benötigen – das kann unpraktisch sein, wenn du planst, dich auch diese Daten in deinen Analyseregeln zu stützen.

## Entitätsseite

Die UEBA-Hauptseite erreichst du, indem du im Microsoft Sentinel-Portal auf der linken Seite „**Entitätsverhalten**“ auswählst. Von dort aus (ein Beispiel siehst du in Abbildung 13-34 unten) kannst du nach einer bestimmten Entität (z. B. einem Benutzer oder einer IP-Adresse) suchen und jene Entitäten identifizieren, mit denen die meisten Warnungen verbunden sind.

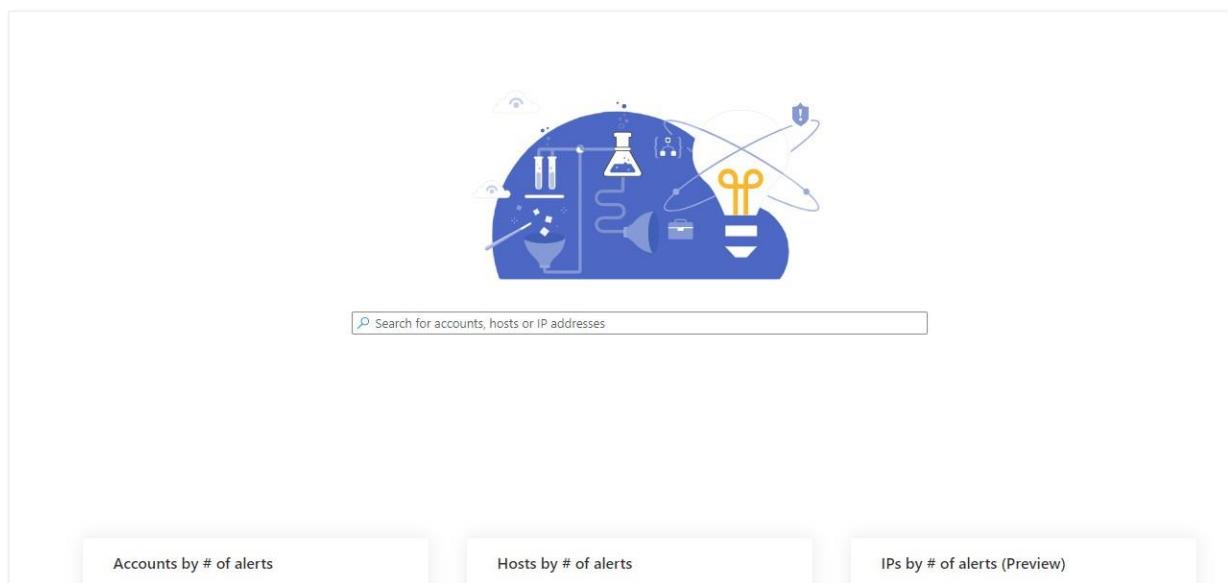


Abbildung 13-34: Beispiel für die Entitätsseite

Von hier aus kannst du zu spezifischen Entitätsseiten wechseln. Diese Seiten enthalten detaillierte Informationen über die gewählte Entität. Je nach Typ der Entität sind unterschiedliche Informationen verfügbar. Einige dieser Informationen stammen aus Microsoft Sentinel-Protokollen, andere werden von der UEBA-Engine eingespeist. Dazu zählen:

- Host
  - Heartbeat-Informationen
  - Informationen von Microsoft Defender for Endpoint (z.B. offene Warnungen und Schwachstellen)
  - Bei Azure-Hosting: Informationen über zugehörige Azure-Ressourcen
- IP

- Geolokalisierungsdaten (Land, ISP usw.)
- Von dieser IP-Adresse verwendete Hosts/Benutzer
- Konto
  - Entra ID-spezifische Informationen (aus der Tabelle IdentityInfo)
  - Informationen zu Benutzer-Peers
  - Relevante Ereignisse, die den Benutzer betreffen (z. B. Hinzufügen zu einer Gruppe oder ob kürzlich das Ereignisprotokoll seines Computers gelöscht wurde).

Diese Entitätsseiten sind während einer Untersuchung sehr nützlich und werden ebenfalls im Abschnitt „Auf Vorfälle mit Microsoft Sentinel reagieren“ behandelt.

## Aktivitäten

**Aktivitäten** sind eine relativ unbekannte, aber leistungsstarke Funktion in Microsoft Sentinel. Dabei handelt es sich um auffällige Ereignisse, die auf bestimmten Entitätsseiten während deiner Untersuchung angezeigt werden. Diese Ereignisse lösen keine Warnung aus, können aber deine Analyse unterstützen. Sie helfen dabei, wichtige Informationen sichtbar zu machen – etwa, wann ein Benutzer zuletzt sein Passwort geändert hat oder wie viele fehlgeschlagene Anmeldeversuche es in der vergangenen Woche gab.

Die Konfiguration dieser Aktivitäten ist etwas versteckt: Öffne das UEBA-Blade über „Entitätsverhalten“ und klicke oben auf „Entitätsseite anpassen“, wie in Abbildung 13-35 dargestellt.

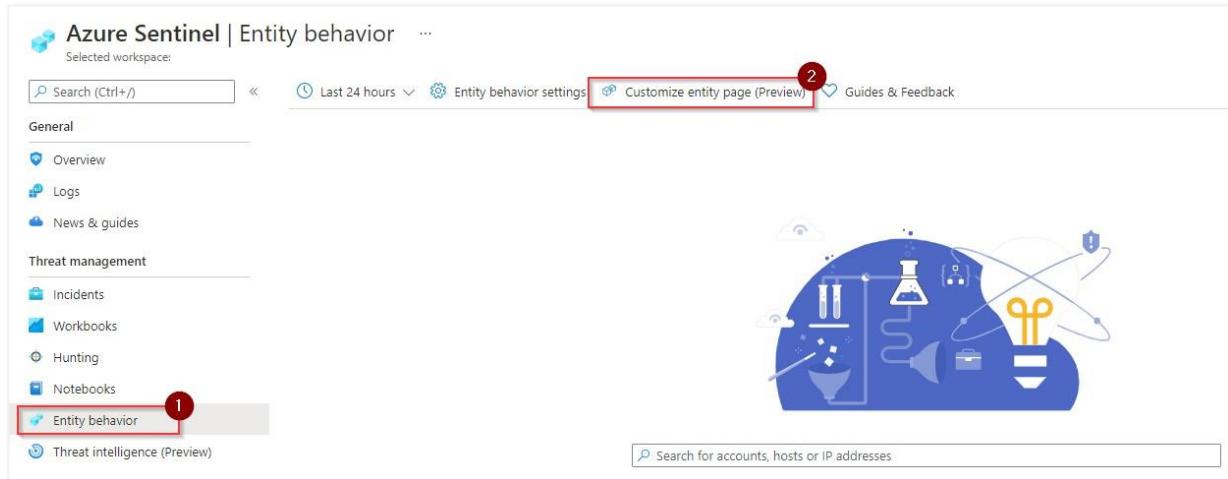


Abbildung 13-35: Navigation zum Konfigurationsbildschirm der Aktivitäten

Dadurch gelangst du zur Übersicht aller aktuellen Aktivitäten. Dort gibt es zwei Registerkarten: „Meine Aktivitäten“ und „Aktivitätsvorlagen“. Die Vorlagen werden von Microsoft

bereitgestellt und können von dir verwendet werden. In der Registerkarte „Meine Aktivitäten“ findest du sowohl Aktivitäten, die du von Grund auf neu erstellt hast, als auch Vorlagen, die du aktiviert hast.

Die Aktivitäten verwenden eine Entität als Eingabe und durchsuchen die Protokolle nach bestimmten Ereignissen. Die als Eingabe verwendete Entität ergibt sich aus der Entitätsseite, die du zu diesem Zeitpunkt geöffnet hast (wenn du zum Beispiel die Entitätsseite für das Konto „John Doe“ geöffnet hast, verwendet die Aktivität John Doe als Eingabe). Das bedeutet, dass die Abfragen für Aktivitäten eine bestimmte Reihe von Variablen enthalten müssen, die zur Laufzeit verwendet werden. Die verfügbaren Variablen hängen von der jeweiligen Entität ab. Setze diese Variablen beim Schreiben der Abfrage in doppelte geschweifte Klammern.

Nachdem du die Abfrage geschrieben hast, kannst du einen Aktivitätstitel definieren. Dieser erscheint später in der Zeitachse, wenn du einen Vorfall untersuchst. Der Titel kann dynamische Inhalte enthalten, die aus der Abfrage stammen, die du in der Aktivität verwendet hast.

Mit der Aktivitätsfunktion kannst du benutzerdefinierte Ereignisse erstellen, die auf einer Abfrage basieren und eine Aktivität darstellen. So lassen sich zusätzliche Details über einen Benutzer (oder eine andere Entität) zur Untersuchungsseite hinzufügen. Dadurch erhältst du mehr Kontext zu Warnungen, Entitäten und relevanten Aktivitäten. Diese Details werden im Aktivitätstitel dargestellt, der mithilfe von Variablen konfiguriert wird.

In Abbildung 13-36 unten findest du ein Beispielergebnis. Es stammt aus einer Vorlage, die zeigt, bei wie vielen verschiedenen Ressourcen sich der Benutzer angemeldet hat.



Abbildung 13-36: Beispiel für eine Aktivität

In diesem Fall durchsucht eine spezifische Abfrage die SigninLogs nach Anmeldungen bei unterschiedlichen Azure-Ressourcen. Der Name der Aktivität lautet „Der Benutzer hat sich bei einer Azure-Ressource angemeldet“, und der Aktivitätstitel wird wie folgt konfiguriert:

**The user signed in to {{shortResourceId}} {{Count}} time(s)**

Durch die Verwendung doppelter geschweifter Klammern kannst du auf Spalten der zurückgegebenen Abfrageergebnisse verweisen. So lässt sich dynamischer Inhalt in den Aktivitätstitel integrieren. In unserem Beispiel ist shortResourceId gleich „Microsoft.aadiam“ und count ist 25 (da die Abfrage 25 Ereignisse zurückgegeben hat).

**Hinweis:** Auch wenn Aktivitäten eine nützliche Funktion sind, solltest du nicht zu viele davon erstellen, da dies dazu führen kann, dass deine Untersuchungszeitachse zu voll wird. Ich empfehle, Aktivitäten zu erstellen, die für jeden erstellten Vorfall, wertvolle Informationen liefern. Überlege, welche Informationen du brauchst, um einen Vorfall zu lösen, und versuche, diese Informationen als Aktivität hinzuzufügen. Einige nützliche Aktivitäten könnten das Zurücksetzen von Passwörtern, die Anzahl der geografischen Länder, aus denen der Benutzer aktiv war.

## Watchlists

Beim Erstellen von Analyseregeln kann es vorkommen, dass du eine bestimmte Liste von Parametern in mehreren Regeln referenzieren musst. Dazu zählen zum Beispiel:

- dein oberes Führungsteam
- hochwertige Server, wie z. B. Verzeichnisserver
- eine Liste von Dienstkonten oder kritischen Konten wie ein Break-Glass-Konto

Während du diese Liste in jeder Analyseregel pflegen könntest, kann es sehr schwierig sein, sie zu warten, wenn du diese Informationen in vielen Regeln wiederverwenden möchtest. Diese Herausforderung können **Watchlists** für dich lösen.

Watchlists sind eine Funktion in Microsoft Sentinel, mit der du zentral verwaltete Informationen hinterlegen kannst, die du bei der Erstellung von Analyse- und Hunting-Abfragen wiederverwenden kannst. Das ist besonders nützlich, wenn du z. B. neue Mitglieder im Führungsteam oder neue Dienstkonten ergänzen musst, ohne jede Regel manuell zu aktualisieren.

Es gibt zwei Möglichkeiten, eine Watchlist zu erstellen:

1. Hochladen einer lokalen CSV-Datei mit den erforderlichen Informationen
2. Verknüpfen einer CSV-Datei, die in einem Azure Storage-Konto gespeichert ist

Das Hochladen einer lokalen CSV-Datei ist am einfachsten. Du bist jedoch auf eine maximale Größe von 3,8 MB pro Watchlist beschränkt. Wenn du größere Watchlists erstellen musst, solltest du deine CSV-Datei in ein Azure Storage-Konto hochladen und sie auf diese Weise einspeisen.

Eine Watchlist kann durch Hochladen einer CSV-Datei mit den erforderlichen Informationen erstellt werden. Bevor du also eine Watchlist erstellen kannst, solltest du eine CSV-Datei erstellen, die die Informationen enthält, die du hochladen möchtest. Microsoft Sentinel bietet ein paar Watchlist-Vorlagen, die du als Ausgangspunkt verwenden kannst. Du findest sie, indem du die Registerkarte **Vorlagen** im Bereich **Watchlists** auswählst. Zum Zeitpunkt des Schreibens sind sechs Vorlagen verfügbar:

- High Value Assets
- Identity Correlation
- Network Addresses
- Service Accounts
- Terminated Employees
- VIP-Benutzer

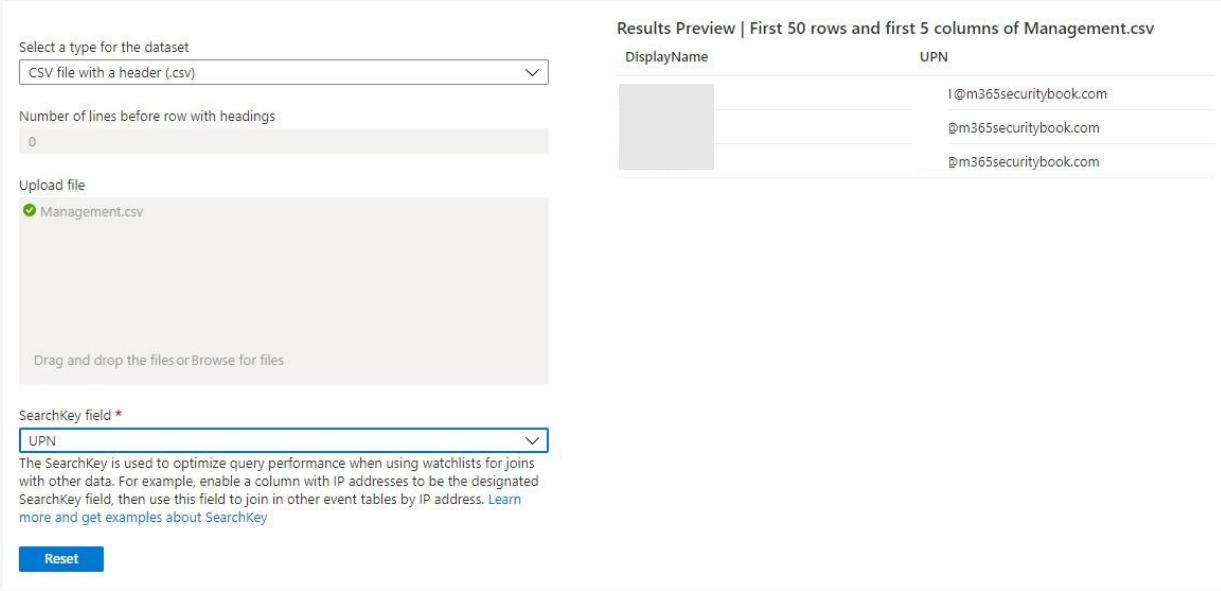
Diese Vorlagen sind besonders hilfreich für Einsteiger, da sie dir die manuelle Erstellung einer CSV-Datei ersparen. Wenn du eine Vorlage auswählst und „**Aus Vorlage erstellen**“ klickst, wirst du durch einen Assistenten geführt. Dort kannst du ein Beispielschema herunterladen und es als Basis für deine eigene Watchlist nutzen.

**Hinweis:** Der große Vorteil dieser Watchlist-Vorlagen ist, dass sie mit UEBA korrelieren (was früher in diesem Kapitel behandelt wurde). Das erscheinen Watchlist-Informationen auch in bestimmten Entitätsansichten, sodass du z.B. sofort erkennst, ob eine Entität ein High Value Asset ist.

Wenn du eine Watchlist von Grund auf neu erstellen möchtest, wähle "**Meine Watchlists**" und dann **Neu hinzufügen** oben auf dem Bildschirm. Jetzt hat sich der Watchlist-Assistent geöffnet. Im Assistenten stehen dir drei Felder zur Verfügung, um Metadaten zur Watchlist hinzuzufügen:

- **Name:** Der Anzeigename der Watchlist, der verwendet wird, um verschiedene Watchlists über das Portal leicht zu identifizieren.
- **Beschreibung:** Ein optionales Feld, welches dir ermöglicht, zusätzliche Informationen über die Watchlist hinzuzufügen.
- **WatchlistAlias:** Der Alias, Welcher verwendet wird, um in Abfragen auf die Watchlist zu verweisen. Wähle etwas Kurzes und Aussagekräftiges.

Der nächste Schritt besteht darin, die CSV-Datei hochzuladen oder den Link zum Azure Blob Storage hinzuzufügen und einen Suchschlüssel zu bestimmen. Ein Suchschlüssel ist die Spalte, die am wahrscheinlichsten verwendet wird, wenn diese Watchlist mit anderen Daten verknüpft wird. Nachdem du die CSV-Datei hochgeladen hast, kannst du die Watchlist-Daten in der **Fehler! Verweisquelle nicht gefunden** in der Vorschau anzeigen.



Select a type for the dataset  
CSV file with a header (.csv)

Number of lines before row with headings  
0

Upload file  
Management.csv

Drag and drop the files or Browse for files

SearchKey field \*  
UPN

The SearchKey is used to optimize query performance when using watchlists for joins with other data. For example, enable a column with IP addresses to be the designated SearchKey field, then use this field to join in other event tables by IP address. [Learn more and get examples about SearchKey](#)

Reset

DisplayName	UPN
	I@m365securitybook.com
	@m365securitybook.com
	@m365securitybook.com

Abbildung 13-37: Erstellung einer Watchlist

Während du die Daten überprüfst, kannst du die Qualität der Watchlist überprüfen und bei Bedarf die Eigenschaft "Anzahl der Zeilen vor der Zeile mit Überschriften" aktualisieren. Wenn du mit dem Inhalt deiner Watchlist zufrieden bist, schließe den Assistenten ab, indem du auf **Überprüfen und erstellen** und dann auf **Erstellen** klickst. Wenn du eine große Watchlist über eine Datei auf einem Azure Storage-Konto eingespeist hast, kann es eine Weile dauern, bis sie eingespeist und in deiner Microsoft Sentinel-Umgebung verfügbar ist.

Alle Watchlists werden in eine große Tabelle namens „Watchlist“ eingefügt. Deine Watchlists sind nicht von der Aufbewahrungsdauer betroffen, die du für deinen Log Analytics-Arbeitsbereich festgelegt hast – sie bleiben dort, bis du sie löscht. Während du die Watchlist-Tabelle selbst abfragen und die richtigen Daten herausfiltern kannst, bietet Microsoft Sentinel zwei einfache Funktionen, um gezielt darauf zuzugreifen:

- **\_GetWatchListAlias** gibt eine Übersicht über alle Watchlists zurück, die in deiner Umgebung verfügbar sind, und liefert dir den Alias.
- Mit **\_GetWatchlist** kannst du den Inhalt einer bestimmten Watchlist abrufen, indem du den Alias als Parameter angibst. Danach kannst du die Daten in deinen Regeln verwenden, um bestimmte Ereignisse zu filtern.

Es gibt zwei Methoden, eine Watchlist in einer Abfrage zu nutzen. Du kannst die Tabelle, die du durchsuchst, und die Watchlist mit einer join-Anweisung verknüpfen. Oder du speicherst die Watchlist-Daten in einer Variablen und verwendest sie als Array. Die join-Methode ist zwar etwas komplexer, wird aber bevorzugt, weil sie mehr Granularität erlaubt – im Gegensatz zur Variablenmethode, die nur eine Spalte aus der Watchlist verwenden kann.

Die folgende Abfrage ist ganz einfach und gibt nur Anmeldeinformationen für Benutzer in der Watchlist des Managementteams zurück, indem sie eine join-Anweisung verwendet.

### SigninLogs

```
| join kind=leftsemi _GetWatchlist('ManagementTeam') on  
$left.UserPrincipalName == $right.SearchKey
```

Wenn du einen Join-Befehl verwendest, musst du definieren, welche Spalten verwendet werden sollen. Dies geschieht durch die Definition der Variablen \$left und \$right. \$left bezieht sich auf deine erste Tabelle, die in unserem Fall die SigninLogs-Tabelle ist. \$right bezieht sich auf die rechte Tabelle in unserer Join-Anweisung, die die Watchlist ist (speziell die Spalte SearchKey).

Das gleiche Ergebnis kann erreicht werden, indem der Inhalt der Watchlist in der Variablen gespeichert und in einer Where-Anweisung verwendet wird, wie unten gezeigt.

```
let managementTeam =_GetWatchlist('ManagementTeam') | project SearchKey;  
SigninLogs  
| where UserPrincipalName !in (managementTeam)
```

**Hinweis:** Auch wenn du zum Erstellen einer Watchlist eine CSV-Datei hochladen musst, kannst du bestehende Watchlists direkt im Portal bearbeiten – einfach die Watchlist auswählen und oben **Watchlist aktualisieren > Watchlist-Elemente bearbeiten** klicken.

## Bedrohungsinformationen

Wie bei jeder anderen SIEM-Plattform musst du Microsoft Sentinel mit Informationen über aktuelle Angriffe versorgen, damit es effektiv arbeiten kann. Das geschieht durch das Hinzufügen sogenannter Bedrohungsinformationen (Threat Intelligence, kurz TI). Ein Indikator für Bedrohungsinformationen ist ein Hinweis auf einen Angriff – etwa ein Element, das während eines Angriffs beobachtet wurde und dir helfen kann, verdächtige Aktivitäten zu erkennen. Ein Indikator kann viele Formen annehmen, z. B. eine IP-Adresse, eine URL oder ein Dateihash.

Oft stellt sich die Frage, warum man überhaupt eigene Indikatoren zu einer SIEM-Plattform wie Microsoft Sentinel hinzufügen sollte – insbesondere, wenn man bereits XDR-Tools wie Microsoft Defender verwendet. Hier sind ein paar Gründe dafür:

1. Durch das Hinzufügen von TIs zu Microsoft Sentinel kannst du Datenquellen auf spezifische Bedrohungen überwachen, die möglicherweise nicht von deiner XDR-Plattform überwacht werden (z. B. Firewall-Protokolle).

2. Es kann Bedrohungen geben, die spezifisch für die Branche deiner Organisation (z. B. Finanzen oder Gesundheitswesen) und sogar länderspezifisch sind. Da diese Informationen so spezifisch sind, sind sie möglicherweise noch nicht in deinem XDR-Produkt enthalten. Durch Hinzufügen spezifischer TIs bist du auch über diese spezifischen Angriffe informiert.
3. Die meisten XDR-Anbieter veröffentlichen nicht die genauen Indikatoren, die sie verwenden, da dies Angreifern eine Möglichkeit bietet, zu identifizieren, welche ihrer Exploits, Systeme und Tools von diesem Antiviren-/EDR-System identifiziert werden. Das bedeutet, dass du nie sicher sein kannst, ob dein Produkt dich vor bestimmten Angriffen schützt.

In Microsoft Sentinel sind derzeit fünf Indikatortypen verfügbar:

- Domain-Namen
- Datei-Hashes
- IPv4-Adressen
- IPv6-Adressen
- URL

Jeder Typ erfordert eine typspezifische Eigenschaft (bei URLs z. B. die tatsächliche URL). Zusätzlich gibt es generische Eigenschaften, die für alle Indikatortypen gelten – diese findest du in Tabelle 13-9 unten.

Eigenschaftsname	Beschreibung
Tags	Jeder Indikator kann mehrere Tags zugewiesen haben. Diese Tags können verwendet werden, um ähnliche Indikatoren zu gruppieren. Ein Beispiel-Tag könnte die Angriffsorganisation oder der spezifische Angriffsname sein.
Bedrohungstypen	Bietet zusätzliche Informationen über die Bedrohung, die dem Indikator zugewiesen ist. Handelt es sich um einen bestätigten bösartigen Indikator oder erfordert er weitere Untersuchungen?
Beschreibung	Eine Beschreibung des Indikators. Hier versuche ich, die Quelle des Indikators hinzuzufügen.
Name	Name des Indikators
Zurückgezogen	Durch das Zurückziehen von Indikatoren konfigurierst du sie als inaktiv, was bedeutet, dass sie keine Auswirkungen mehr haben.

Vertrauen	Das Vertrauen, dass dieser Indikator bösartig ist. Dies ermöglicht es dir, Indikatoren schrittweise zu testen, wenn du dir nicht zu 100 % sicher bist, ob sie Auswirkungen auf deine Umgebung haben.
Kill Chains	Die Lockheed Martin Cyber-Kill-Chain. Die Lockheed Martin Kill Chain ist mit dem MITRE ATT&CK vergleichbar, definiert jedoch eine andere Reihenfolge von Operationen. Eine Übersicht über die verschiedenen Phasen findest du auf deren Webseite.
Gültig ab	Ab wann ist der Indikator gültig? Meistens wird dies das heutige Datum sein.
Gültig bis	Das Datum, bis zu dem ein Indikator abläuft. Dies ist nützlich, wenn du zeitlich begrenzte Indikatoren hast.
Erstellt von	Benutzer, der den Indikator erstellt hat.

Tabelle 13-9: Übersicht über die für Indikatoren verfügbaren Eigenschaften

## Hinzufügen von Indikatoren

Das Hinzufügen von Indikatoren für Bedrohungsinformationen zu Microsoft Sentinel kann auf mehrere Arten erfolgen:

- Microsoft Sentinel-Portal
- Security Graph
- Upload Indicators API
- TAXII

Alle Bedrohungsinformationen, die in deiner Umgebung hinzugefügt werden, werden in der Tabelle **ThreatIntelligence** innerhalb des Log Analytics-Arbeitsbereichs gespeichert, für den Microsoft Sentinel aktiviert ist.

Um deine aktuellen Indikatoren für Bedrohungsinformationen anzuzeigen, kannst du entweder die Tabelle **ThreatIntelligence** abfragen oder die Registerkarte **Threat Intelligence** im Microsoft Sentinel-Portal verwenden. Ein Vorteil der Nutzung des Portals ist, dass die Indikatoren dort mit **WhoIs-** und **Geo-Daten** angereichert werden – das sind dieselben Informationen, die auch auf den Entitätsseiten angezeigt werden.

## Portal

Die einfachste Methode, Indikatoren hinzuzufügen, ist über das Microsoft Sentinel-Portal. Wenn du zum Blatt **Threat Intelligence** navigierst, erhältst du eine Übersicht über alle vorhandenen Indikatoren. Durch Klicken auf **Neu hinzufügen** oben kannst du neue Indikatoren erstellen. Zuerst wählst du den passenden Typ aus und gibst anschließend die erforderlichen Details zum Indikator ein, bevor du ihn speicherst.

## API

Es gibt derzeit zwei verschiedene APIs, um Bedrohungsinformationen programmgesteuert zu Microsoft Sentinel hinzuzufügen. Die **Security Graph API** kann Indikatoren sowohl zu Defender als auch zu Sentinel hinzufügen. Die **Upload Indicators API** ist eine neuere Schnittstelle, die speziell auf Microsoft Sentinel zugeschnitten ist und die Security Graph API langfristig ersetzen soll.

Da es unterschiedliche Indikatortypen gibt, musst du deine API-Aufrufe jeweils so gestalten, dass sie die nötigen Informationen für den jeweiligen Typ enthalten. Auch wenn das auf den ersten Blick komplex wirkt – wegen der Vielzahl an Eigenschaften – ist das Verständnis dieser API eine Fähigkeit, die du als Microsoft-Sicherheitsadministrator unbedingt beherrschen solltest. Eine vollständige Liste aller verfügbaren Eigenschaften findest du in der Microsoft-Dokumentation – sowohl für die [Upload Indicators API](#) als auch für die [Security Graph API](#).

Da die Security Graph API für Microsoft Sentinel eingestellt wird, empfiehlt Microsoft, bei neuen Implementierungen die **Upload Indicators API** zu verwenden. Der einzige Grund, die Security Graph API weiterhin zu nutzen, besteht darin, dass sie Indikatoren sowohl zu Microsoft Sentinel als auch zu Microsoft 365 Defender übertragen kann. Das Defender-Team hat bislang nicht mitgeteilt, ob diese API auch für deren Produkte eingestellt wird. Persönlich würde ich mit einer Umstellung der API warten, bis Microsoft hierzu weitere Informationen veröffentlicht.

### Upload Indicators API

Im Mai 2023 veröffentlichte Microsoft eine neue API namens “Upload Indicators API”. Mit dieser Änderung wurde angekündigt, dass der alte Datenkonnektor **“Threat Intelligence Platforms”**, der die Graph Security API verwendete, eingestellt wird. Während die Graph Security API sowohl für Microsoft Defender als auch für Sentinel funktionierte, ist die Upload Indicators API speziell auf Microsoft Sentinel zugeschnitten.

Um die API zu verwenden, musst du eine App-Registrierung in Entra ID erstellen. Dieser Anwendung muss die Rolle **“Microsoft Sentinel Contributor”** zugewiesen werden. Nach der Erstellung der Anwendung installierst du die Lösung **“Threat Intelligence”** in deinem Microsoft Sentinel-Arbeitsbereich, bevor du sie nutzen kannst.

## Security Graph

Die Microsoft Security Graph API ist eine Schnittstelle, die im gesamten Microsoft-Sicherheitsökosystem verwendet wird. Eine der Funktionen des Security Graph besteht darin, Informationen zwischen verschiedenen Microsoft-Produkten auszutauschen. Sie ermöglicht es dir aber auch, bestimmte Informationen über Warnungen abzufragen und Indikatoren für Bedrohungsinformationen sowohl zu Microsoft Defender for Endpoint als auch zu Microsoft Sentinel hinzuzufügen. Aufgrund ihrer API-Struktur ist sie ideal geeignet, um die Erfassung neuer Indikatoren in diese Produkte zu automatisieren.

Bevor du die Security Graph API in deiner Organisation nutzen kannst, musst du eine App-Registrierung erstellen, die über die Berechtigung "**ThreatIndicators.ReadWrite.OwnedBy**" verfügt. Nachdem du diese Anwendung erstellt hast, kann sie TI-Informationen in jede Microsoft Sentinel-Instanz deiner Umgebung einfügen. Weitere Informationen zum Erstellen von App-Registrierungen findest du in Kapitel 2.

Wenn du die Security Graph API verwendest, um Indikatoren für Bedrohungsinformationen in Microsoft Sentinel einzuspeisen, wird der Datenkonnektor "**Threat Intelligence Platforms**" als verbunden angezeigt, wie in Abbildung 13-38 dargestellt.

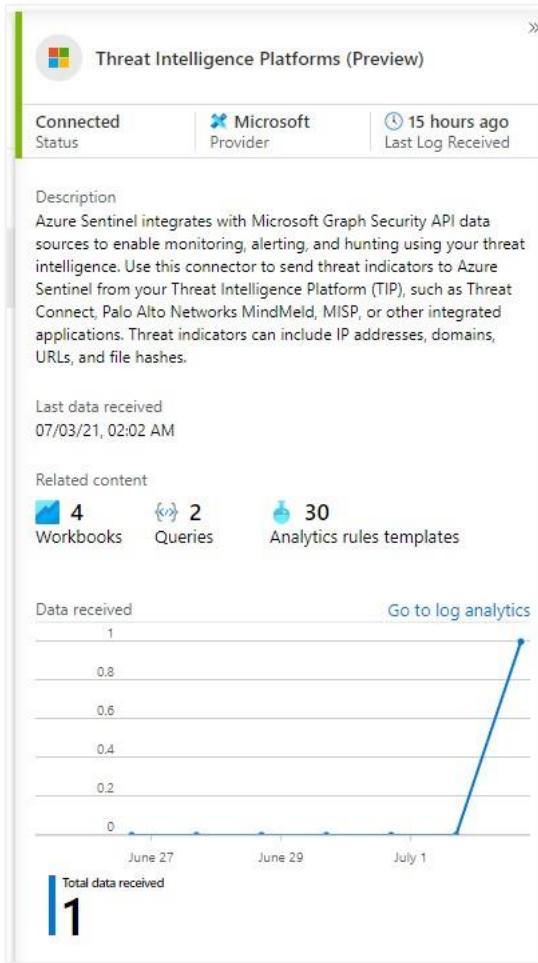


Abbildung 13-38: Threat Intelligence-Konnektor

**Hinweis:** Durch die Verwendung der Security Graph-API zum Einspeisen von Indikatoren in Microsoft Sentinel kannst du problemlos Bedrohungsinformationsquellen von Drittanbietern integrieren. Ein gutes Beispiel ist eine Integration mit **MISP**, der Malware Information Sharing Platform, über die du Indikatoren mit anderen Organisationen abrufen und teilen kannst. Microsoft [stellt Skripte](#) zur Verfügung, mit denen du MISP mit Microsoft Sentinel synchronisieren kannst.

## TAXII

TAXII steht für **Trusted Automated Exchange of Intelligence Information** und ist ein weltweiter Standard für den sicheren Austausch von Indikatoren zwischen verschiedenen Produkten. Damit kannst du bestehende TAXII-Server in Microsoft Sentinel integrieren. Das erfolgt durch Aktivierung des Datenkonnektors **“Threat Intelligence – TAXII”** und Konfiguration von Microsoft Sentinel zum regelmäßigen Abrufen neuer Indikatoren vom TAXII-Server. Der Hauptvorteil der Verwendung von TAXII gegenüber der Security Graph API besteht darin, dass

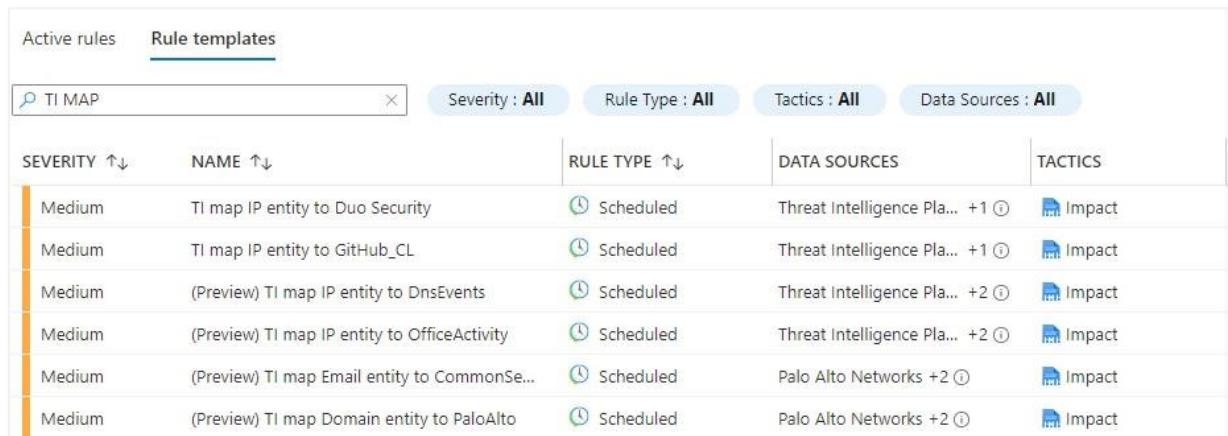
Microsoft Sentinel die API-Abfragen an TAXII und die Konvertierung der verschiedenen Formate automatisch in ein von Microsoft Sentinel verstandenes Modell übernimmt.

## Verwendung von TI in Analyseregeln

Ein häufiges Missverständnis ist, dass Microsoft Sentinel diese Indikatoren sofort nach dem Hinzufügen verwendet. Das ist jedoch nicht der Fall. Nachdem du die Indikatoren eingespeist hast, musst du deine Analyseregeln und Hunting-Abfragen so konfigurieren, dass sie die Daten berücksichtigen, die du in die Tabelle **ThreatIntelligence** eingefügt hast.

Microsoft stellt viele Vorlagen für Analyseregeln bereit, mit denen du die Daten aus der ThreatIntelligence-Tabelle mit anderen Datenquellen in Microsoft Sentinel korrelieren kannst. Nachdem du Indikatoren eingespeist hast, empfehle ich dir, die verfügbaren Vorlagen durchzusehen und alle Regeln zu aktivieren, die zu den von dir eingespeisten Daten passen. Wenn es für deine Daten keine passende Vorlage gibt, musst du manuell eine Analyseregel erstellen, die deine Datenquellen und TI-Indikatoren miteinander verknüpft.

Die verfügbaren Analyseregeln findest du in der Registerkarte **“Regelvorlagen”** im Abschnitt **Analytics**. Sie beginnen alle mit **“TI map”**, wie in Abbildung 13-39 zu sehen ist.



The screenshot shows the Microsoft Sentinel interface with the 'Rule templates' tab selected. A search bar at the top contains 'TI MAP'. Below the search bar are filters for Severity (All), Rule Type (All), Tactics (All), and Data Sources (All). The main area displays a table of rule templates:

SEVERITY ↑↓	NAME ↑↓	RULE TYPE ↑↓	DATA SOURCES	TACTICS
Medium	TI map IP entity to Duo Security	Scheduled	Threat Intelligence Pla... +1 ⓘ	Impact
Medium	TI map IP entity to GitHub_CL	Scheduled	Threat Intelligence Pla... +1 ⓘ	Impact
Medium	(Preview) TI map IP entity to DnsEvents	Scheduled	Threat Intelligence Pla... +2 ⓘ	Impact
Medium	(Preview) TI map IP entity to OfficeActivity	Scheduled	Threat Intelligence Pla... +2 ⓘ	Impact
Medium	(Preview) TI map Email entity to CommonSe...	Scheduled	Palo Alto Networks +2 ⓘ	Impact
Medium	(Preview) TI map Domain entity to PaloAlto	Scheduled	Palo Alto Networks +2 ⓘ	Impact

Abbildung 13-39: Beispiel für TI Analytic Rules.

## Verwendung der Microsoft Threat Intelligence (TI)

Das Finden korrekter und wirkungsvoller Bedrohungssindikatoren kann ziemlich aufwendig sein. Kleine IT-Teams haben oft nicht die Ressourcen, um Bedrohungsinformationsindikatoren zu suchen, zu validieren und einzuspeisen. Dieses Problem kannst du teilweise lösen, indem du einige Bedrohungsinformationsindikatoren von Microsoft verwendest.

Microsoft bietet die Möglichkeit, seine Indikatoren mit deinen Daten zu verbinden. Das geschieht durch die Aktivierung einer bestimmten Analyseregel. Wenn eine Übereinstimmung in deinen Daten gefunden wird, wird ein Vorfall erstellt und der Indikator erscheint im Blatt **“Threat Intelligence”**.

Das bedeutet, dass die Microsoft-TI-Indikatoren nicht standardmäßig für dich verfügbar sind. Sie werden nur dann in deine Umgebung aufgenommen, wenn eine Übereinstimmung mit deinen Daten gefunden wurde. Ich vermute, dass Microsoft damit sein geistiges Eigentum schützen möchte, damit Microsoft Sentinel nicht als kostenlose TI-Quelle genutzt werden kann.

Um diese Funktion zu aktivieren, musst du die Vorlage **“Microsoft Threat Intelligence Analytics”** in deiner Umgebung erstellen. Es gibt keine Konfigurationsoptionen für diese Regel – du solltest sie also so übernehmen, wie sie ist. Derzeit unterstützt diese Regel nur **CEF**, **DNS** und **Syslog** als Quellen. Andere Quellen werden momentan nicht unterstützt, aber ich gehe davon aus, dass künftig zusätzliche Datenquellen folgen werden.

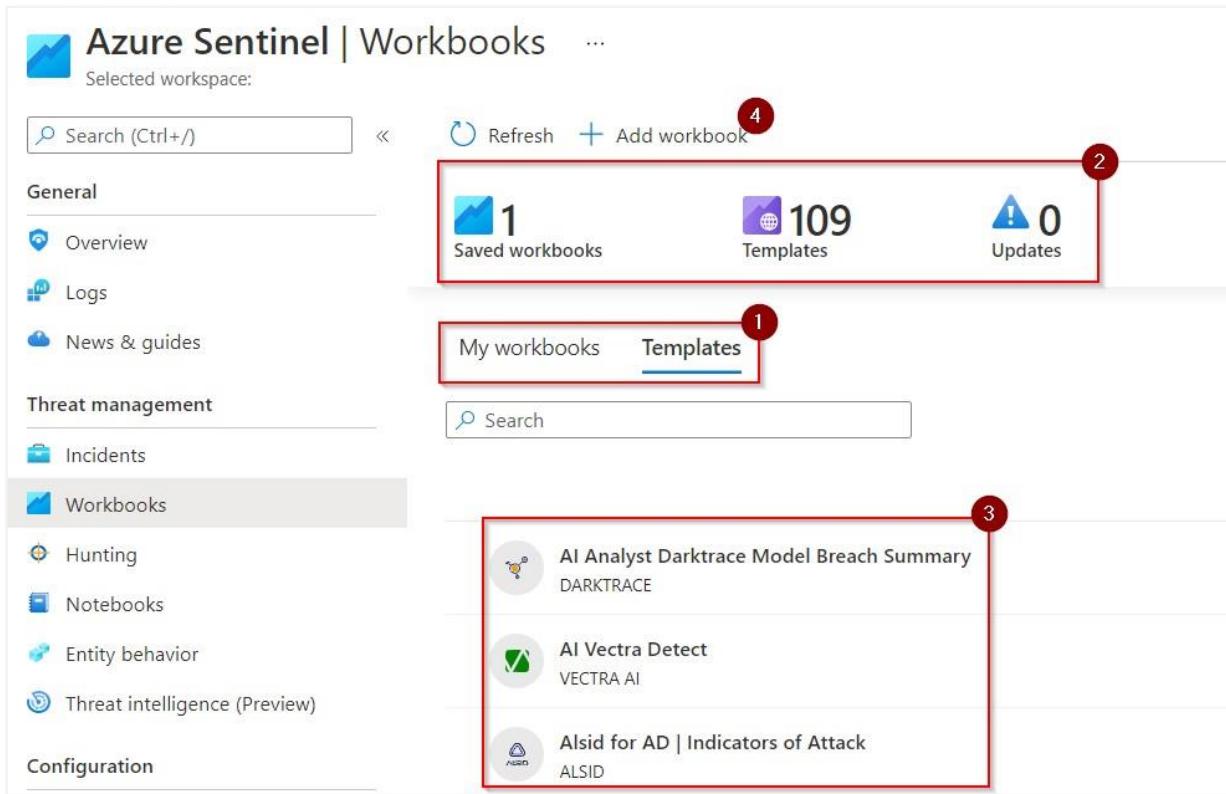
## Daten mit Arbeitsmappen visualisieren

Auch wenn alle Microsoft Sentinel-Daten in den Rohdaten verfügbar sind, ist es oft praktischer, die Daten visuell aufzubereiten. Das kann aus mehreren Gründen sinnvoll sein:

- Um die Untersuchungen zu verbessern, kannst du eine Arbeitsmappe erstellen, die einige der Vorfalldaten visualisiert (z. B. Benutzeraktivitäten). Menschen sind meist besser darin, Bilder zu lesen und zu interpretieren als reinen Text!
- Um Berichte für dein Management zu erstellen und einen allgemeinen Überblick zu geben über:
  - Microsoft Sentinel-Vorfälle und deren Status
  - Datenkonnektor-spezifische Daten wie z.B. Anmeldeprotokolle

Die Visualisierung von Daten in Microsoft Sentinel erfolgt über Arbeitsmappen. Diese sind keine Sentinel-spezifische Funktion, sondern Teil von **Azure Monitor**. Da Microsoft Sentinel auf **Log Analytics** basiert, kannst du Arbeitsmappen auch im Sentinel-Kontext verwenden.

Im Microsoft Sentinel-Portal findest du die Arbeitsmappen im Abschnitt **“Bedrohungsmanagement”**. Wenn du diesen öffnest, gelangst du zur Übersicht, wie in Abbildung 13-40 dargestellt.



Azure Sentinel | Workbooks

Selected workspace:

Search (Ctrl+ /) Refresh Add workbook

General

- Overview
- Logs
- News & guides

Threat management

- Incidents
- Workbooks**
- Hunting
- Notebooks
- Entity behavior
- Threat intelligence (Preview)

Configuration

1 Saved workbooks    109 Templates    0 Updates

My workbooks    **Templates**

Search

AI Analyst Darktrace Model Breach Summary  
DARKTRACE

AI Vectra Detect  
VECTRA AI

Alsid for AD | Indicators of Attack  
ALSID

Abbildung 13-40: Übersicht der Microsoft Sentinel-Arbeitsmappen.

Die Arbeitsmappen-Übersicht enthält Folgendes:

- 1) Zwei Registerkarten, **Vorlagen** und **Meine Arbeitsmappen**. Vorlagen sind von Microsoft bereitgestellte Arbeitsmappen, die du verwenden kannst. Unter „Meine Arbeitsmappen“ findest du eigene erstellte oder gespeicherte Vorlagen.
- 2) Oben findest du eine Übersicht über den Status der Arbeitsmappen:
  - a) **Gespeicherte Arbeitsmappen**: Zeigt an, wie viele Arbeitsmappen unter "Meine Arbeitsmappen" gespeichert sind.
  - b) **Vorlagen**: Die Anzahl der verfügbaren Vorlagen von Microsoft.
  - c) **Updates**: Wenn du eine Arbeitsmappe-Vorlage gespeichert hast, wird sie nicht automatisch aktualisiert, wenn Microsoft eine neue Version veröffentlicht. Die Zahl veranschaulicht, für wie viele deiner gespeicherten Vorlagen ein Update verfügbar ist.

## Einführung in Azure Monitor-Arbeitsmappen

Bevor du dich mit den Sentinel-spezifischen Anwendungsfällen für Arbeitsmappen beschäftigst, ist es sinnvoll, eine allgemeine Einführung in Arbeitsmappen zu bekommen. Eine Sentinel-

Arbeitsmappe unterscheidet sich kaum von einer Azure Monitor-Arbeitsmappe, da beide auf denselben Bausteinen basieren.

Wenn du zum ersten Mal eine Arbeitsmappe erstellst, kannst du verschiedene Objekte hinzufügen. Dazu gehören:

- **Text:** Bietet dir die Möglichkeit, Klartext zu einer Arbeitsmappe hinzuzufügen, was ideal für Dokumentation oder eine Einführung in die Daten ist.
- **Parameter:** Durch die Konfiguration von Parametern kann der Inhalt der Arbeitsmappe entsprechend der Eingabe des Benutzers aktualisiert werden.
- **Abfrage:** Eine Abfrage wird verwendet, um Daten abzurufen, die in der Arbeitsmappe verwendet werden sollen. Es stehen verschiedene Datenquellen zur Verfügung. Wenn die Quelle Log Analytics ist, können Daten über KQL abgefragt werden, was dir ermöglicht, Visualisierungen auf der Grundlage von Microsoft Sentinel-Daten zu erstellen.
- **Metriken:** Metriken sind spezifisch für bestimmte Azure-Ressourcen und ermöglichen dir, Dinge wie den verwendeten Speicher einer VM einzubeziehen. Dies ist beim Erstellen von Microsoft Sentinel-Arbeitsmappen nicht anwendbar.
- **Gruppen:** Mit Gruppen kannst du mehrere Unterelemente in einem Block hinzufügen, was für Layout und Formatierung nützlich ist.

## Datenquellen

Azure Monitor unterstützt nicht nur Log Analytics als Quelle, sondern auch Quellen wie Azure Resource Manager, Data Explorer, JSON und mehr. Diese Datenquellen kannst du in den „Abfrage“-Blöcken der Arbeitsmappe verwenden.

Der Quellname für Log Analytics lautet „**logs**“ und ermöglicht es dir, Protokolle aus einem bestimmten Log Analytics-Arbeitsbereich über eine KQL-Abfrage abzurufen. Da KQL auch in Microsoft Sentinel-Analyseabfragen verwendet wird, sollte es für dich relativ einfach sein, die Abfrage entsprechend anzupassen, um die gewünschten Daten zu erhalten.

**Azure Resource Manager** ist eine weniger bekannte, aber sehr nützliche Quelle. Diese Option erlaubt dir, API-Aufrufe an die Azure Resource Manager REST-API zu senden, um Azure-spezifische Daten einzubeziehen, die nicht in den Protokollen enthalten sind. Ein gutes Beispiel ist der "[Workspace Usage Report](#)", der den Resource Manager nach den aktivierten Analyseregeln in deiner Umgebung abfragt.

## Visualisierungen

Innerhalb einer Abfrage kannst du die gewünschte Datenquelle sowie verschiedene Visualisierungen definieren. Azure Monitor bietet dir eine Vielzahl von [vorgefertigten Visualisierungen](#), die dir bei der Erstellung deiner Microsoft Sentinel-Arbeitsmappen helfen können.

## Parameter

Parameter kannst du in einer Arbeitsmappe verwenden, um den Inhalt entsprechend der Eingabe des Benutzers zu aktualisieren, der den Bericht verwendet. Dazu gehören beispielsweise:

- Auswahl des richtigen Arbeitsbereichs (anstatt ihn in der Arbeitsmappe fest zu codieren)
- Auswahl des richtigen Benutzers für eine Untersuchung
- Änderung des Zeitraums, um den Umfang einer Arbeitsmappe zu begrenzen.

## Verwendung von Arbeitsmappe-Vorlagen

Microsoft stellt dir eine Vielzahl integrierter Arbeitsmappen zur Verfügung. Sie sind eine gute Möglichkeit, dich mit dem Konzept vertraut zu machen, damit du deine eigenen Arbeitsmappen später anpassen oder erstellen kannst. Während du durch die Bibliothek der Arbeitsmappe-Vorlagen navigierst, kannst du eine Arbeitsmappe auswählen, über die du weitere Informationen abrufen möchtest. In Abbildung 13-41 unten siehst du eine Arbeitsmappe-Vorlage, die für einen in Microsoft Sentinel integrierten Drittanbieter-Konnektor erstellt wurde.

Sie liefert dir einige wertvolle Einblicke in die Arbeitsmappe:

- Eine Beschreibung der Arbeitsmappe
- Die erforderlichen Log Analytics-Datenquellen, die von der Arbeitsmappe verwendet werden

Ein paar Beispielbilder, die die Arbeitsmappe veranschaulichen.

Alsid for AD | Indicators of Exposure»  
ALSID

Workbook showcasing the state and evolution of your Alsid for AD Indicators of Exposures alerts.

Required data types: ⓘ  
 AlsidForADLog\_CL

Relevant data connectors: ⓘ  
AlsidForAD



[View template](#)
[Save](#)

Abbildung 13-41: Beispiel für eine Arbeitsmappe-Vorlage

In der Detailansicht der Arbeitsmappe hast du zwei Möglichkeiten (unten in Abbildung 13-41 dargestellt):

- **Vorlage anzeigen:** Hier kannst du die Vorlage mit den Daten deines eigenen Log Analytics-Arbeitsbereichs anzeigen. So kannst du die volle Funktionalität der Arbeitsmappe nutzen, allerdings ohne die Möglichkeit, ihre Komponenten zu bearbeiten.
- **Speichern:** Dadurch wird die Arbeitsmappe in die Registerkarte "Meine Arbeitsmappen" aufgenommen und in derselben Ressourcengruppe wie dein Microsoft Sentinel-Arbeitsbereich gespeichert.

Nachdem du die Arbeitsmappe gespeichert hast, kannst du zur Registerkarte „Meine Arbeitsmappen“ navigieren, um deine eigene Instanz davon zu sehen. Alle Änderungen, die du hier vornimmst, bleiben in deiner eigenen Umgebung und sind nur für dich sichtbar.

Wenn du deine Arbeitsmappe gespeichert hast, stehen dir einige zusätzliche Aktionen zur Verfügung:

- **Gespeicherte Arbeitsmappe anzeigen:** Damit kannst du deine eigene Instanz der Arbeitsmappe öffnen und sie an deine individuellen Bedürfnisse anpassen.
- **Löschen:** Entfernt die Arbeitsmappe als Ressource aus deiner Umgebung. Wenn sie auf einer Vorlage basiert, bleibt die ursprüngliche Vorlage weiterhin verfügbar.

Es kann vorkommen, dass Microsoft später ein Update für eine von dir gespeicherte Arbeitsmappe-Vorlage bereitstellt. Standardmäßig werden Arbeitsmappen, die du gespeichert hast, nicht automatisch aktualisiert. Wenn ein Update verfügbar ist, wird dir das in der Ansicht „Meine Arbeitsmappen“ angezeigt, wie in Abbildung 13-42 dargestellt.

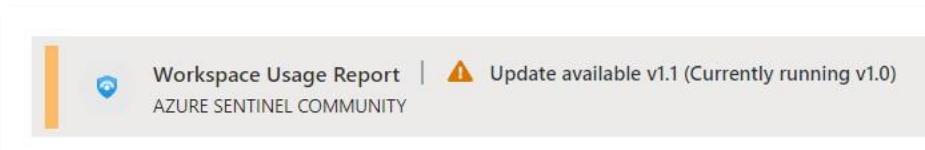


Abbildung 13-42: Arbeitsmappe, die ein Update benötigt

Wenn du auf die Arbeitsmappe klickst, siehst du, dass eine zusätzliche Option verfügbar ist (siehe Abbildung 13-43).

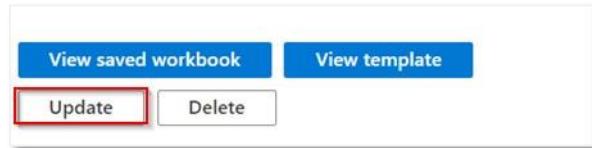


Abbildung 13-43: Aktualisieren einer gespeicherten Vorlage

**Sei vorsichtig:** Wenn du das Update durchführst, wird deine benutzerdefinierte Version der Arbeitsmappe mit der neuesten von Microsoft bereitgestellten Version der Arbeitsmappe überschrieben. Dabei gehen alle von dir vorgenommen Änderungen verloren.

## Erstellen deiner eigenen Arbeitsmappe

Wenn deine Anforderungen die verfügbaren Arbeitsmappen-Vorlagen übersteigen, musst du deine eigenen Arbeitsmappen erstellen. Ein vollständiges Tutorial würde den Rahmen dieses Buchs sprengen, aber ich möchte dir einige Tipps und Tricks zur Erstellung neuer Arbeitsmappen mit auf den Weg geben.

Arbeitsmappen kannst du auf drei verschiedene Arten erstellen:

- Manuell über das Portal
- Importieren einer vorhandenen Arbeitsmappe über JSON
- Durch Verwendung einer ARM-Vorlage

Auch wenn die Erstellung einer eigenen Arbeitsmappe auf den ersten Blick die beste Idee zu sein scheint, kann sie für Einsteiger etwas knifflig sein. Ich empfehle dir als Anfänger, zunächst die verfügbaren Vorlagen durchzusehen und herauszufinden, welche Visualisierungen du benötigst. Diese Visualisierungen kannst du dann in deine eigenen Arbeitsmappen übernehmen. Das Einzige, was du anpassen solltest, sind die Datenquellen hinter den Visualisierungen, da diese mit hoher Wahrscheinlichkeit andere sein werden. In den meisten Fällen handelt es sich dabei um eine KQL-Abfrage. Am besten erstellst du die Abfrage zuerst in den Rohdaten, um sicherzustellen, dass du die richtigen Informationen erhältst, und fügst sie dann in die Arbeitsmappe ein.

Arbeitsmappen werden im Hintergrund als JSON-Objekte gespeichert. Dadurch kannst du einzelne Teile einer Arbeitsmappe ganz einfach kopieren. Um den Quellcode anzuzeigen, beginne mit der Bearbeitung der Arbeitsmappe und wähle dann oben rechts das Symbol „Quellcode“ aus (siehe Abbildung 13-44). Du hast die Wahl zwischen zwei Vorlagentypen:

- **Galerievorlagen**, die nur den Code für die Arbeitsmappe enthalten.
- **ARM-Vorlage**, die alles enthalten, was du benötigst, um die Arbeitsmappe über den Azure Resource Manager bereitzustellen.



Abbildung 13-44: Anzeigen des Quellcodes einer Arbeitsmappe

**Hinweis:** Microsoft hat auf [seinem Blog](#) eine großartige Einführung in Arbeitsmappen bereitgestellt, die sehr detailliert auf die verschiedenen Bausteine eingeht.

## Zugriff auf Arbeitsmappen bereitstellen

Eine häufige Frage zu Arbeitsmappen ist, wie sie mit dem Management geteilt werden können und welche minimalen Berechtigungen erforderlich sind, um sie anzuzeigen. Durch die Zuweisung der Rolle „Workbook Reader“ an ein Mitglied stellst du sicher, dass der Empfänger die Arbeitsmappe lediglich lesen kann, ohne Zugriff auf Microsoft Sentinel als Ganzes zu erhalten.

Eine weitere Möglichkeit besteht darin, Arbeitsmappen direkt aus dem Portal zu drucken und als PDF zu versenden. Klicke dazu auf die drei Punkte neben dem Namen der Arbeitsmappe und wähle „**Inhalt drucken**“, wie in Abbildung 13-45 zu sehen. Es versteht sich von selbst, dass diese Option keine skalierbare Lösung darstellt, da du den Vorgang für jede Arbeitsmappe manuell wiederholen müsstest, sobald jemand eine Kopie anfordert.



Abbildung 13-45: Drucken einer vorhandenen Arbeitsmappe

**Hinweis:** Auch wenn Arbeitsmappen eine hervorragende Möglichkeit zur Visualisierung von Daten sind, eignen sie sich nicht ideal für Personen, die keinen Zugriff auf das Azure-Portal haben. Wenn du umfangreiche Berichte erstellen möchtest, die auch andere Datenquellen einbeziehen, empfehle ich dir die Verwendung von PowerBI. PowerBI bietet einen [integrierten Konnektor](#) für Log Analytics, mit dem du Daten einfach in deine Dashboards einspeisen kannst.

## Automatisierung von Antworten

Wenn du Microsoft Sentinel als Architekt verwaltst oder als SOC-Analyst auf Vorfälle reagierst, wirst du möglicherweise mit vielen sich wiederholenden Aufgaben konfrontiert. Dazu gehören unter anderem:

- Benachrichtigung der zuständigen Analysten, wenn ein neuer Vorfall oder eine neue Warnung generiert wird.
- Anreicherung von Warnungsdaten mit zusätzlichen Informationen aus anderen (externen) Quellen.
- Automatisiertes Auslösen von Aktionen wie das Blockieren von IP-Adressen, das Deaktivieren von Benutern oder das Isolieren von Geräten.

Die Automatisierung in Microsoft Sentinel erfolgt über sogenannte Playbooks, bei denen es sich um Azure Logic Apps handelt. Azure Logic Apps sind die Enterprise-Variante von Power Automate (ehemals Microsoft Flow) und ermöglichen dir die Erstellung von Low-Code-Automatisierungslösungen. Das sorgt dafür, dass SOC-Analysten oder Systemadministratoren – die meist wenig Programmiererfahrung haben – einige der monotonen und wiederkehrenden Aufgaben problemlos automatisieren können.

## Playbooks

Azure Logic Apps bestehen aus Aktionen und Triggern. Ein Trigger ist der erste Schritt in einer Logik-App und definiert, wann die App ausgeführt werden soll. Aktionen sind Schritte, die innerhalb der Logik-App ausgeführt werden, und können vom Aktualisieren eines Microsoft

Sentinel-Vorfalls über das Erstellen einer Variablen bis hin zum Aufrufen einer benutzerdefinierten API reichen.

**Hinweis:** Logic Apps haben zwei Bereitstellungstypen: Verbrauchs- und Standardversion. Microsoft Sentinel-Playbooks unterstützen beide Varianten. Welchen Typ du wählst, hängt ganz von deinem Setup und deiner Organisation ab. Standard-Playbooks bieten Vorteile wie Unterstützung für private Endpunkte und erhöhte Leistung. Ich empfehle die Verwendung von Verbrauchs-Logic Apps, es sei denn, du benötigst spezifische Funktionen der Standard-SKU.

Logic Apps bieten eine Vielzahl integrierter Konnektoren, die eine einfache Integration in Anwendungen von Drittanbietern ermöglichen. Ein Beispiel für einen Konnektor ist „Outlook Office 365“, der eine Aktion „E-Mail senden“ enthält. Durch die Vielzahl verfügbarer Konnektoren wird die Integration in andere Produkte erleichtert, ohne dass du dich um APIs oder die Formatierung deiner Daten kümmern musst.

Innerhalb der verfügbaren Konnektoren ist auch ein Microsoft Sentinel-Konnektor enthalten. Dieser ermöglicht es dir, einfach mit Microsoft Sentinel-Warnungen und -Vorfällen zu interagieren, ohne die API verwenden und verstehen zu müssen. Neben einem Konnektor für Microsoft Sentinel selbst veröffentlicht das Microsoft Sentinel-Team auch zahlreiche Integrationen von Drittanbietern, wie zum Beispiel Konnektoren für CrowdStrike, Palo Alto oder Azure Firewall.

Eine Übersicht über alle verfügbaren Microsoft Sentinel-Trigger und -Aktionen findest du in der Microsoft-Dokumentation. Eine Logik-App mit einem Microsoft Sentinel-Trigger wird als **Playbook** bezeichnet. Aktuell sind zwei Trigger verfügbar:

- Microsoft Sentinel Alert - Wenn eine Reaktion auf eine Microsoft Sentinel-**Warnung** ausgelöst wird (als Warnungs-Trigger bezeichnet).
- Microsoft Sentinel Incident - Wenn eine Microsoft Sentinel-**Vorfall**-Erstellungsregel ausgelöst wurde (als Vorfall-Trigger bezeichnet).

Beide Trigger haben ihre eigenen Anwendungsfälle. Der erste wird jedes Mal ausgelöst, wenn eine Warnung für eine bestimmte Analyseregel generiert wird. Das auf der Vorfallerstellungsregel basierende Playbook wird ausgeführt, wenn ein Vorfall erstellt wird. Um zu bestimmen, welche Playbooks in welchem Szenario ausgeführt werden sollen, müssen Automatisierungsregeln verwendet werden, die später in diesem Abschnitt behandelt werden.

**Hinweis:** Vor Juli 2022 unterstützten Automatisierungsregeln keine Warnungs-Trigger. Für Warnungs-Trigger mustest du die Playbooks zu den entsprechenden Analyseregeln hinzufügen. Das hat sich geändert und Automatisierungsregeln unterstützen jetzt auch die Warnungserstellung.

Die Liste der Aktionen für den Microsoft Sentinel-Konnektor wächst kontinuierlich und ist zu umfangreich, um sie vollständig abzudecken. Zwei Aktionen möchte ich dennoch besonders hervorheben:

- **Vorfall aktualisieren:** Mit dieser Aktion können mehrere Parameter eines Vorfalls (wie Status, Besitzer und Bezeichnungen) gleichzeitig aktualisiert werden.
- **Entitäten -- Get <abc>**: Es gibt einige verschiedene Aktionen wie *Entitäten -- Get Accounts* oder *Entitäten -- Get IPs*, mit denen du die Entitäten abrufen kannst, die einer bestimmten Warnungen zugeordnet sind. Dadurch kannst du die relevanten Entitäten gezielt abrufen und auf ihrer Grundlage Aktionen durchführen. Dieser Prozess unterstreicht die Wichtigkeit, deine Analyseregeln sorgfältig einzurichten und die richtigen Entitäten zu definieren. Mit einer solchen Aktion kannst du zum Beispiel einen bestimmten Host isolieren oder den Zugriff auf ein Konto blockieren.

Bevor ein Konnektor verwendet werden kann, muss er über die richtige Authentifizierungsmethode verfügen. Der Microsoft Sentinel-Konnektor unterstützt derzeit drei verschiedene Authentifizierungsmethoden:

1. Verwendung eines Entra ID-Benutzers
2. Verwendung eines Dienstprinzipals
3. Verwendung einer verwalteten Identität

Die erste Option solltest du nach Möglichkeit vermeiden – sie ist lediglich für Testszenarien geeignet. Wenn du dich als Entra ID-Benutzermeldest, bindest du dieses spezifische Playbook an genau diesen Benutzer. Wird dieser Benutzer deaktiviert oder verliert seine Berechtigungen für Microsoft Sentinel, funktioniert das Playbook nicht mehr. Auch wenn du versuchst, dies mit Dienstkonten zu umgehen, empfehle ich, solche Konten möglichst zu vermeiden. Sobald du MFA oder Bedingten Zugriff für das Konto benötigst, treten Probleme auf – etwa ein ablaufendes Token, das eine erneute Authentifizierung erforderlich macht. In diesem Fall ist die Verbindung nicht mehr gültig und das Playbook schlägt fehl.

Die zweite und dritte Methode sind die bevorzugten Optionen für produktive Umgebungen. Wenn du einen Dienstprinzipal in einem Playbook einsetzen möchtest, musst du zuerst eine App-Registrierung in Entra ID erstellen und dieser App die erforderlichen Berechtigungen für die Microsoft Sentinel-Ressourcengruppe (z. B. Microsoft Sentinel Reader oder Contributor) zuweisen.

Wenn du eine verwaltete Identität zur Authentifizierung nutzen möchtest, erhält die Logik-App automatisch eine eigene Identität – im Grunde ein Dienstprinzipal im Mandanten mit einem Namen, der dem der Logik-App entspricht. Diesem spezifischen Dienstprinzipal musst du dann die notwendigen Berechtigungen für Microsoft Sentinel erteilen. Der Vorteil einer verwalteten Identität gegenüber einem Dienstprinzipal besteht darin, dass du dich nicht um Anmeldeinformationen wie Client-Secrets oder Zertifikate kümmern musst. Diese verwaltet die Ressource automatisch. Der Nachteil: Diese Identität ist an die jeweilige Logik-App gebunden.

Wenn du sie neu bereitstellst oder mehrere Instanzen erzeugst, wird jeweils eine neue verwaltete Identität erstellt.

Wenn du gerade erst mit Microsoft Sentinel anfängst, können Playbooks zunächst ein wenig knifflig erscheinen. Diese Lernkurve lässt sich durch die Verwendung von Playbook-Vorlagen aus dem Content Hub erleichtern. Genau wie bei Analyseregeln stehen dir Vorlagen zur Verfügung, mit denen du auf einfache Weise nützliche Playbooks erstellen kannst. Die Verwendung dieser Vorlagen bringt zwei Hauptvorteile mit sich:

- Sie ermöglicht es Neulingen, einige gängige Automatisierungsaufgaben zu erkunden.
- Der Bereitstellungsmechanismus über die Playbook-Galerie ist deutlich reibungsloser als die manuelle Bereitstellung von ARM-Vorlagen.

Der erste Schritt besteht darin, im Content Hub nach interessanten Automatisierungen zu suchen. Nach der Installation kannst du zur Playbook-Galerie navigieren, dort die Registerkarte „**Automatisierung**“ auswählen und anschließend auf „Playbook-Vorlagen“ klicken, um alle verfügbaren Vorlagen anzuzeigen.

Jede Vorlage enthält bestimmte Merkmale, anhand derer du leicht filtern kannst, welche für deinen Anwendungsfall geeignet sein könnte:

- **Playbook-Trigger:** Zeigt an, welcher Trigger verwendet wird. Zurzeit sind der Warnungs-, Vorfall- und Entitäts-Trigger verfügbar.
- **Verwendete Konnektoren:** Gibt an, welche API-Verbindungen eingesetzt werden und mit welchen Produkten das Playbook interagiert. Ein Playbook mit dem "PAN-OS"-Konnektor interagiert beispielsweise mit einem Palo Alto-Gerät.
- **Entitäten:** Gibt an, welche Entitäten im Playbook erwartet werden. Wenn du etwas URL's blockieren möchtest, muss der Vorfall eine gültige URL-Entität enthalten. Manche Vorlagen basieren jedoch nicht auf Entitäten, sondern auf dem gesamten Vorfall.
- **Tags:** Sie bieten eine einfache Möglichkeit, Playbooks nach Kategorien wie Anreichung, Behebung oder Benachrichtigungen zu sortieren.

Wenn du ein passendes Playbook gefunden hast, kannst du es über die Schaltfläche „**Playbook erstellen**“ direkt in deiner Umgebung bereitstellen. Ein benutzerdefinierter Assistent wird geöffnet, der dich durch mehrere Registerkarten führt:

- **Grundlagen:** Hier legst du Name, Standort und gegebenenfalls die Verbindung zu Log Analytics fest.
- **Parameter:** Ermöglicht dir die Konfiguration von Playbook-spezifischen Einstellungen.
- **Verbindungen:** Bietet eine einfache Möglichkeit, API-Verbindungen zu verwalten oder mit einer verwalteten Identität zu erstellen. Besonders praktisch: Du kannst vorhandene Verbindungen wiederverwenden oder neue definieren.

Auch wenn deine eigenen Anwendungsfälle nicht exakt zu den Vorlagen passen, ist die Playbook-Galerie eine hervorragende Inspirationsquelle. Du bekommst Einblicke, wie andere gängige Automatisierungsaufgaben umgesetzt haben.

Wenn du ein Playbook lieber selbst von Grund auf erstellen möchtest, kannst du dies direkt in der Microsoft Sentinel-Oberfläche tun. Wähle dazu in der Registerkarte „**Automatisierung**“ die Option „**Erstellen**“ und dann eine der folgenden Möglichkeiten:

- Playbook mit Vorfall-Trigger
- Playbook mit Warnungs-Trigger
- Playbook mit Entitäts-Trigger
- Leeres Playbook

Im Abschnitt zur Microsoft Sentinel-Community in diesem Buch verweise ich auch auf die Sentinel GitHub-Seite. Dort findest du deutlich mehr Playbooks als in der Galerie. Ein Blick dorthin lohnt sich, da viele interessante Anwendungsfälle vertreten sind.

Playbooks kannst du sowohl manuell als auch automatisch ausführen. Wie du dabei vorgehst, hängt vom verwendeten Trigger ab:

- Bei einem Warnungs-Trigger öffnest du den Vorfall, gehst zur Registerkarte **Warnungen** und klickst neben der gewünschten Warnung auf **Playbook ausführen** klickst. Das Playbook wird mit den Details dieser Warnung ausgeführt.
- Bei einem vorfallbasierten Playbook öffnest du den Vorfall, klickst auf **Aktionen** und wählt **Playbook ausführen**. Ideal zum Testen während der Entwicklung oder wenn SOC-Analysten manuelle Maßnahmen wie das Isolieren eines Geräts anstoßen sollen.
- Für entitätsbasierte Playbooks öffnest du eine Entität öffnen und klickst auf "**Playbook ausführen**" auf der jeweiligen Entitätsseite.

Die Wahl des richtigen Triggers ist nicht immer eindeutig. Es gibt keine festen Regeln, welcher Trigger wann zum Einsatz kommen sollte. Ich orientiere mich bei der Entscheidung an folgendem Ansatz:

- **Automatisierter vs. manueller Trigger:** Wenn das Playbooks manuell ausgelöst wird, also z.B. von einem SOC-Analyst, empfehle ich in den meisten Fällen den Entitäts-Trigger. Dieser bietet dir die größte Flexibilität, weil du gezielt auf einzelne Entitäten reagieren kannst – unabhängig davon, ob eine Warnung oder rein Vorfall besteht
- **Umfang der Automatisierung:** Ein Vorfall kann mehrere Entitäten enthalten, etwa verschiedene Hosts oder Konten. Ein vorfallbasiertes Playbook könnte also Maßnahmen auf Entitäten ausführen, die du gar nicht ansprechen möchtest. In solchen Fällen sind warnungsbasierte Playbooks nützlicher, obwohl vorfallbasierte Playbooks in der Regel mehr Funktionen bieten. Ich selbst habe allerdings auch schon einige Probleme mit warnungsbasierten Playbooks erlebt. Automatisierungsregeln

Neben Playbooks gibt es in Microsoft Sentinel eine weitere Funktion namens Automatisierungsregeln. Automatisierungsregeln ermöglichen es dir, eine automatisierte Aktion auszuführen, wenn ein Vorfall generiert wird. Dazu gehören das Ausführen eines Playbooks, das Ändern des Schweregrads und das Aktualisieren des Status eines Vorfalls.

## Konfigurieren der Berechtigungen

Während Automatisierungsregeln ohne jegliche Konfiguration eingerichtet werden können, ist eine Aktion erforderlich, wenn du ein Playbook aus einer Automatisierungsregel auslösen möchtest. Du musst Microsoft Sentinel die Rechte erteilen, bestimmte Logik-Apps auszulösen.

Navigiere dazu zu **Einstellungen**, wähle oben auf der Seite **Einstellungen** und dann **Berechtigungen konfigurieren**, wie in Abbildung 13-46 gezeigt.

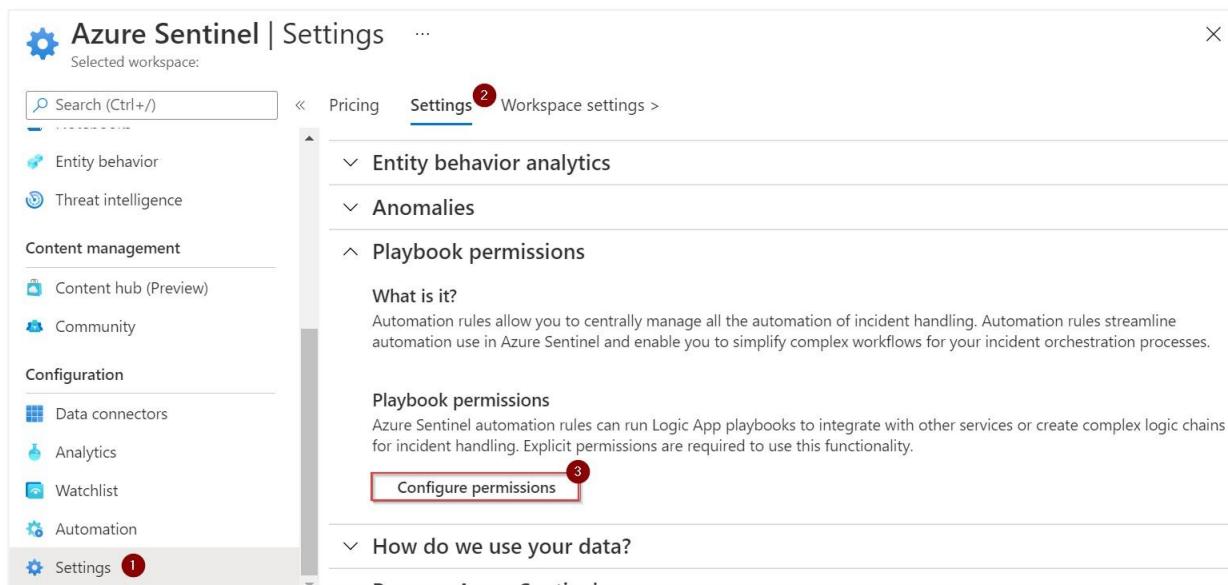


Abbildung 13-46: Hinzufügen der erforderlichen Berechtigungen für Playbooks

Auf der rechten Seite öffnet sich ein neues Blatt und zeigt eine Übersicht über alle deine aktuellen Ressourcengruppen. Du musst jede Ressourcengruppe auswählen, die Playbooks enthält, die Microsoft Sentinel auslösen können muss. Wenn du alle gewünschten Ressourcengruppen ausgewählt hast, klicke auf **Anwenden**, um die notwendigen Berechtigungen hinzuzufügen. Um diese Berechtigungen zu erteilen, sollte der Benutzer, der diese Aktion ausführt, die **Besitzerrolle** für die Ressourcengruppe zugewiesen haben, für die du Berechtigungen hinzufügen möchtest.

Durch die Ausführung dieser Aktionen erhält die Rolle **Azure Security Insights** die Rolle **Microsoft Sentinel Automation Contributor** für die Ressourcengruppe.

**Hinweis:** Wenn du einen mandantenübergreifenden Aufbau über Lighthouse hast, sind einige zusätzliche Konfigurationen erforderlich. Wenn du als Managed Security Services Provider (MSSP) eine Automatisierungsregel konfigurieren möchtest, die ein Playbook im Mandanten des Kunden auslöst, sollte dein "Azure Security Insights"-Dienstprinzipal über Lighthouse die Berechtigung "Microsoft Sentinel Automation Contributor" konfiguriert haben. Eine vollständige Anleitung dazu findest du in der [offiziellen Dokumentation](#).

## Einrichten von Automatisierungsregeln

Um neue Automatisierungsregeln zu erstellen, navigiere zu **Automation** in deinem Microsoft Sentinel-Portal und wähle **Erstellen > Automatisierungsregel**, wie in Abbildung 13-47 zu sehen.

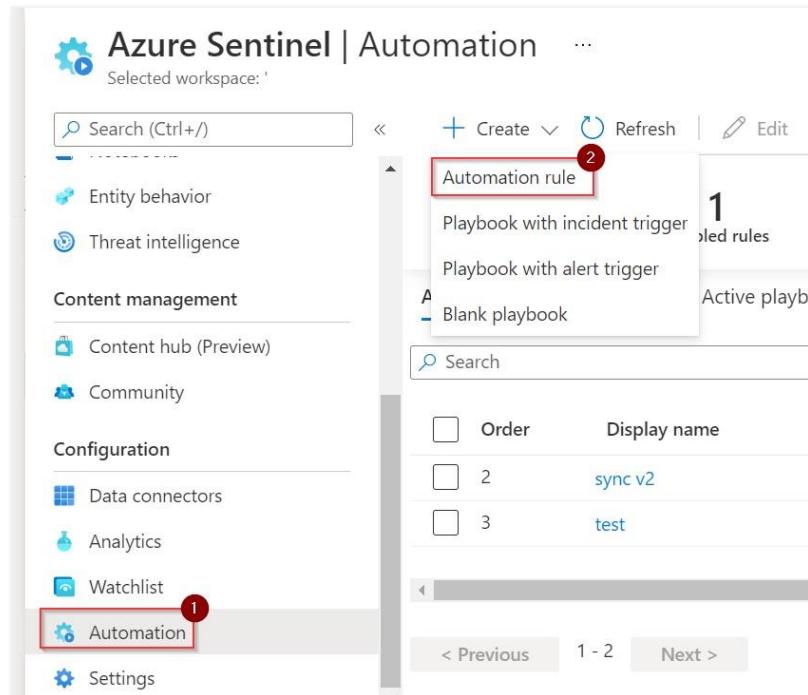


Abbildung 13-47: Erstellen einer neuen Automatisierungsregel

Dadurch wird ein neues Blatt im Portal geöffnet, um die Automatisierungsregel zu erstellen. In diesem Assistenten zur Erstellung von Regeln kann Folgendes konfiguriert werden:

### Name

Der Name der Automatisierungsregel. Dies ist rein eine visuelle Hilfe, um deine verschiedenen Regeln leicht zu identifizieren.

**Namenskonvention:** Wenn du viele verschiedene Automatisierungsregeln verwendest, ist eine Namenskonvention praktisch, um bestimmte Regeln zu identifizieren. Wenn du eine spezifische

Namenskonvention entwickelst, empfehle ich, die Kategorie des Playbooks (Anreicherung/Benachrichtigungen/Auto-Schließen) und die enthaltenen Filter einzubeziehen.

## Trigger

Es stehen verschiedene Trigger zur Verfügung:

- Wenn Vorfall erstellt wird
- Wenn Vorfall aktualisiert wird
- Wenn Warnung erstellt wird

Der erste Trigger wird ausgeführt, wenn der Vorfall zum ersten Mal erstellt wird, während der „Aktualisierungs-Trigger“ jedes Mal ausgeführt wird, wenn ein vorhandener Vorfall aktualisiert wird. Der Erstellungs-Trigger dient dazu, automatisierte Antworten einzurichten und z. B. eine E-Mail an das SOC-Team zu senden, wenn ein neuer Vorfall generiert wird, während der Aktualisierungs-Trigger hervorragend geeignet ist, um Änderungen nachzuverfolgen. Der Aktualisierungs-Trigger wird für jede Aktualisierung eines vorhandenen Vorfalls ausgeführt, außer wenn Entitäten aktualisiert werden.

Es ist wichtig zu beachten, dass der „Warnungs“-Trigger nicht für jede Art von Warnung unterstützt wird. Die einzigen unterstützten Typen sind Warnungen, die von (Sentinel-)Analyseregeln erstellt werden – nicht für Warnungen, die von anderen Sicherheitsprodukten wie Microsoft 365 Defender und Identity Protection erstellt werden.

**Verwendung warnungsbasierter Playbooks:** Manche fragen sich oft, ob die Verwendung warnungsbasierter Playbooks sinnvoll ist, da vorfallsbasierte Playbooks mehr unterstützte Funktionen haben (Aktualisierungen, Unterstützung für alle Vorfallstypen ...). Meine Meinung ist vielleicht untypisch, aber ich glaube, dass es immer noch viel für warnungsbasierte Playbooks spricht. Das hat mehrere Gründe. Erstens sollten Vorfälle nur für Ereignisse erstellt werden, die die Analysten untersuchen sollten. Manchmal möchtest du vielleicht Warnungen erstellen, die zunächst nicht zu einem Vorfall hinzugefügt werden, sondern während einer Untersuchung verwendet werden. Zweitens bevorzuge ich es, die Anreicherung an Warnungen und nicht an Vorfällen durchzuführen. So kann ich Vorfälle sowohl auf der Grundlage der Warnungen als auch der Daten aus der Anreicherung erstellen. Ein Beispiel könnte sein, dass ich den gesamten Datenverkehr zu meiner Web Application Firewall vom imaginären Internet Service Provider (ISP) "M365 Security Book" ignorieren möchte. Da ISP-Daten in Microsoft Sentinel nicht nativ verfügbar sind, müssen diese über eine Drittanbieterquelle angereichert werden. Durch die Verwendung einer warnungsbasierten Automatisierung kann ich diese Daten abrufen und in Log Analytics einfügen. Danach kann eine neue Analyseregel die Warnung und die angereicherten Daten abfragen und einen Vorfall auf der Grundlage dieser beiden Quellen erstellen. Durch die Verwendung warnungsbasierter Automatisierung kann ich vermeiden, Vorfälle zu erstellen, wenn sie nicht anwendbar sind.

## Bedingungen

Durch das Konfigurieren von Bedingungen kannst du definieren, wann die Automatisierungsregel die Aktionen ausführen soll. Du kannst eine oder mehrere Bedingungen konfigurieren. Eine Bedingung ist immer vorhanden, nämlich „Analyseregelnamen enthält“. Damit kannst du aus all deinen aktiven Analyseregeln auswählen. Es gibt keine Möglichkeit, diese Bedingung zu entfernen. Wenn du sie also nicht gezielt verwenden möchtest, solltest du „Alle auswählen“. Dadurch wird sichergestellt, dass die Regel deine weiteren Bedingungen wie z. B.:

- Titel
- Beschreibung
- Schweregrad
- Status
- Taktiken
- Vorfallsanbieter
- Benutzerdefinierte Details
- Warnungsprodukt-Namen
- Verschiedene Entitäten

auswertet.

Durch die Verwendung dieser Bedingungen kannst du genau steuern, wann eine bestimmte Regel ihre konfigurierte Aktion ausführen soll. Die in Automatisierungsregeln verfügbaren Entitäten entsprechen denjenigen, die auch in geplanten Analyseregeln verwendet werden können. So kannst du Automatisierungsregeln z. B. nur dann ausführen, wenn ein Vorfall mit einem bestimmten Benutzer, einer IP-Adresse oder einem Gerät verknüpft ist.

Beachte, dass sich die verfügbaren Bedingungen je nach gewähltem Trigger ändern. Ein Beispiel: Der Erstellungs-Trigger bietet nur die Eigenschaft „Status“, während der Aktualisierungs-Trigger sowohl „geändert von“ als auch „geändert zu“ enthält. So kannst du festlegen, welcher der alte und welcher der neue Wert eines Vorfalls ist.

Auch wenn es viele Bedingungen gibt, stößt du möglicherweise auf Einschränkungen bei der eigenen Konfiguration. Wenn du mehrere Bedingungen hinzufügen möchtest, kannst du zwischen einer „und“- oder einer „oder“-Verknüpfung wählen. Die „oder“-Anweisung eignet sich gut für komplexe Bedingungen, z. B. wenn du bestimmte Entitäten innerhalb eines Vorfalls auf die Whitelist setzen willst. Du kannst zudem mehrere „zulässige Werte“ für eine einzelne Bedingung definieren, indem du neben dem bedingten Wert auf die Schaltfläche „Hinzufügen“ klickst – wie in Abbildung 13-46 neben dem (1)-Symbol zu sehen ist. Dieses Bild zeigt, wie komplexe Automatisierungsregeln aufgebaut sein können. Zum Beispiel:

- Führe die Regel nur aus, wenn der Vorfalltitel „Ungewöhnliche Anmeldung“ **oder** „anonyme IP“ **oder** „Bösartige IP“ lautet
- **und** nur wenn eine der folgenden Bedingungen erfüllt ist (**oder**):
  - o Der Vorfall enthält einen Benutzer namens "Thijs"

- Der Vorfall enthält einen Benutzer namens "Michael" **und** findet von Michaels Gerät aus statt.

#### Conditions

If

Analytic rule name	Contains	All
--------------------	----------	-----

AND

Title	Contains	Unfamiliar Sign-in	
-------	----------	--------------------	--

OR

Anonymous IP	
--------------	--

OR

Malicious IP	
--------------	--

1

AND

Account name	Contains	thijs		
--------------	----------	-------	--	--

+ Add

OR

Account name	Contains			
--------------	----------	--	--	--

AND

Host name	Contains			
-----------	----------	--	--	--

+ Add

2

Abbildung 13-48: Erweiterte Bedingungen in Automatisierungsregeln

Die Bedingung „Analyseregelnamen“ enthält alle aktiven Analyseregeln. Wenn du z. B. den Microsoft 365 Defender-Konnektor verwendest, können Vorfälle mit einem Titel wie „Bitrep-Malware wurde verhindert“ erstellt werden. Um gezielt nach solchen Vorfällen zu filtern, solltest du die Bedingung „Titel“ statt „Analyseregelnamen“ verwenden.

#### Aktionen

Aktionen legen fest, welche Auswirkung die Automatisierungsregel hat, wenn die Bedingungen erfüllt sind. Du kannst folgende Aktionen definieren:

- Playbook ausführen (Mit dieser Aktion kannst du Playbooks mit dem **Vorfall**-Trigger ausführen.)
- Status ändern
- Schweregrad ändern
- Besitzer zuweisen
- Tags hinzufügen
- Aufgabe hinzufügen

**Hinweis:** Auch wenn Logik-Apps sehr leistungsfähig sind, gibt es Szenarien, in denen sie nicht ausreichen. Das kann z.B. bei besonders komplexen Logiken der Fall sein, die sich mit Azure Logic Apps allein nicht umsetzen lassen. In solchen Fällen empfehle ich dir, Azure Functions in Betracht zu ziehen. Auch wenn es keine native Integration für Azure Functions gibt, kannst du eine Logik-App erstellen, die deine Azure-Funktion auslöst. So kannst du weiterhin Automatisierungsregeln mit komplexeren Abläufen nutzen.

## Ablauf der Regel

Standardmäßig ist der Regelablauf auf „unbegrenzt“ gesetzt, d. h. die Regel wird dauerhaft ausgeführt. Du kannst aber auch einen Ablaufzeitpunkt definieren. Das ist besonders nützlich für Tests mit bestimmten Benutzern oder Hosts, bei denen du Vorfälle ausnahmsweise ignorieren möchtest. Ohne definierten Ablauf könnte diese Regel später vergessen werden und weiterhin ungewollt aktiv bleiben – selbst wenn der Test längst abgeschlossen ist.

## Reihenfolge

Die Reihenfolge ist nützlich, wenn du mehrere Automatisierungsregeln konfigurierst. Die Regeln werden in aufsteigender Reihenfolge ausgeführt. Das bedeutet: Regeln mit der niedrigsten Reihenfolge (höchste Priorität) werden zuerst ausgeführt. Danach folgen die Regeln mit einer höheren Reihenfolge (niedrigere Priorität). Du kannst die Reihenfolge konfigurieren, indem du eine bestimmte Zahl im Konfigurationsbildschirm eingibst oder sie im Übersichtsbildschirm der Automatisierungsregel per Drag & Drop verschiebst.

**Tipp:** Auch wenn du für mehrere Regeln dieselbe Reihenfolge verwenden kannst, nutze ich gerne eine Konvention, um meine Automatisierungsregeln zu gruppieren. Eine mögliche Konvention für die Reihenfolge könnte wie folgt aussehen:

- Reihenfolge 0-50: Prioritätsaktualisierungen
- Reihenfolge 51-100: Anreicherung
- Reihenfolge 101-150: Automatisierte Aktionen

Ein solches Beispiel ermöglicht dir einen klaren Überblick und hilft dir zu erkennen, wann eine bestimmte Automatisierungsregel ausgelöst wird.

Standardmäßig werden Regeln sequenziell ausgeführt, eine nach der anderen, unmittelbar nachdem die vorherige Regel abgeschlossen wurde. Die einzige Ausnahme gilt für Aktionen, die ein Playbook ausführen. In diesem Fall hängt der Zeitpunkt der nächsten Regel von der Dauer des Playbooks ab:

- Extrem kurze Playbooks (weniger als eine Sekunde) verursachen keine Verzögerung, und die nächste Automatisierungsregel wird sofort ausgeführt.

- Wenn ein Playbook weniger als zwei Minuten dauert, erfolgt die Ausführung der nächsten Regel mit einer Verzögerung von maximal zehn Minuten nach Abschluss des Playbooks.
- Dauert das Playbook länger als zwei Minuten dauert, wird die nächste Regel nach der zweiminütigen Wartezeit ausgeführt.

**Protokollierung:** Leider ist die Protokollierung für Automatisierungsregeln nur eingeschränkt verfügbar. Zwar gibt es Unterstützung für Automatisierungsregeln innerhalb der Funktion "Sentinel Health", dennoch ist es oft schwierig, den genauen Grund zu ermitteln, warum eine Regel ausgeführt wurde oder eben nicht. Du kannst die Funktion "Sentinel Health" nutzen, um zu überprüfen, ob eine bestimmte Automatisierungsregel eine Aktion ausgeführt hat. Weitere Informationen findest du im nächsten Abschnitt "Microsoft Sentinel Health".

## Einrichten deiner ersten Automatisierung in Microsoft Sentinel

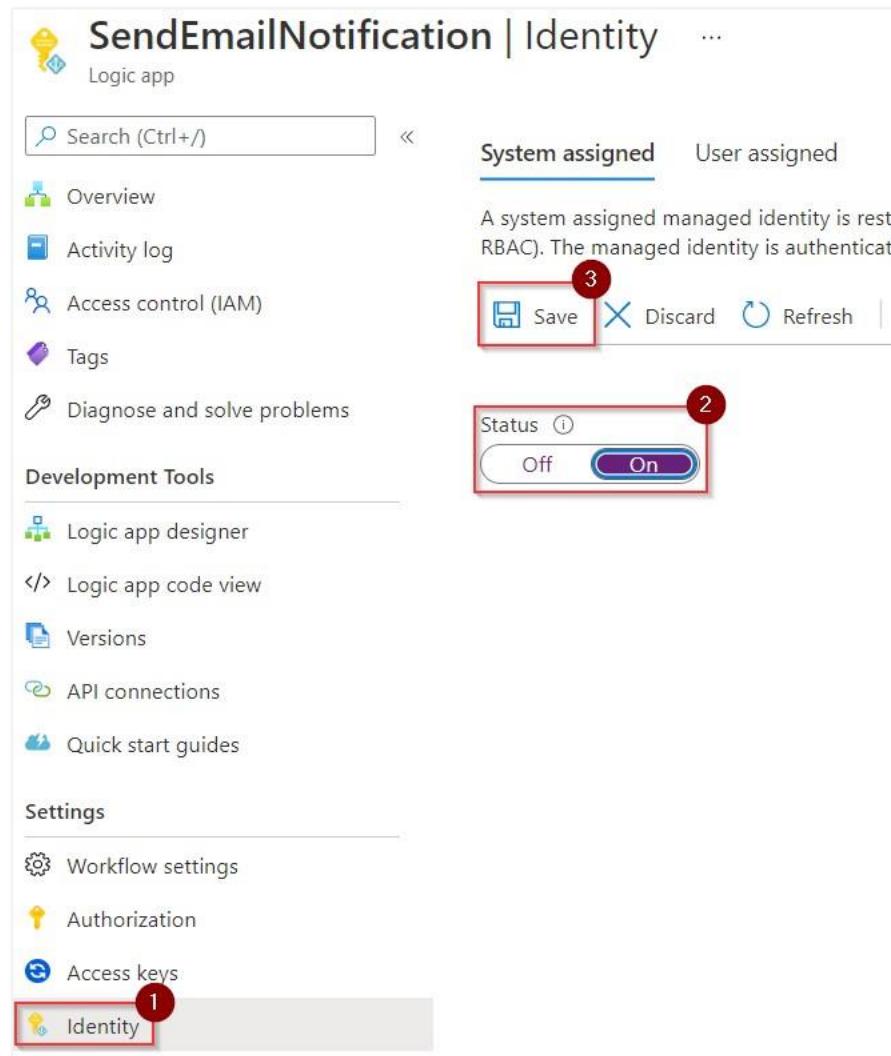
Als Beispiel zeige ich dir, wie du dein erstes Playbook und deine erste Automatisierungsregel erstellst, die in Microsoft Defender for Endpoint ein Untersuchungspaket sammelt, wenn ein Vorfall mit hohem Schweregrad generiert wurde. Der erste Schritt ist das Erstellen eines Playbooks. Navigiere dazu in Microsoft Sentinel zum Blatt „**Automation**“ und wähle „**Erstellen > Neues Playbook hinzufügen**“.

Nun öffnet sich der Assistent zur Erstellung von Logik-Apps. Du musst den Namen, den Standort, das Abonnement und die Ressourcengruppe der Logik-App angeben. Die übrigen Details sind optional und müssen nicht ausgefüllt werden. Wenn du alle erforderlichen Informationen eingetragen hast, klicke auf „**Erstellen**“.

**Hinweis:** Ich empfehle dringend, die Log Analytics-Integration zu konfigurieren. Damit wird deine Logik-App so eingerichtet, dass sie ihre Ausführungsprotokolle an einen Log Analytics-Arbeitsbereich sendet. Mit diesen Protokollen kannst du Fehler in deiner Logik-App überwachen und wirst benachrichtigt, wenn eine bestimmte Logik-App nicht ausgeführt werden kann.

Nachdem die Logik-App erstellt wurde, klicke auf die Schaltfläche „**Zur Ressource**“, um zur neu erstellten Logik-App zu gelangen. Der Logik-App-Designer öffnet sich und zeigt einige Beispielvorlagen. Wähle „**Leere Logik-App**“, um zu beginnen.

Der erste Schritt besteht darin, eine **verwaltete Identität** für die Logik-App zu aktivieren. Dadurch kann sich die Logik-App bei deiner Microsoft Sentinel-Umgebung authentifizieren. Navigiere zum Abschnitt „**Identität**“, ändere den Status auf „**Ein**“ und speichere die Konfiguration, wie in Abbildung 13-49 gezeigt.



The screenshot shows the Azure Logic App 'SendEmailNotification' settings page. The 'Identity' section is highlighted with a red box labeled '1'. The 'Status' switch is set to 'On' with a red box labeled '2'. The 'Save' button is highlighted with a red box labeled '3'.

**SendEmailNotification | Identity**

Logic app

Search (Ctrl+ /) <>

System assigned User assigned

A system assigned managed identity is restricted by RBAC). The managed identity is authenticated by the provider.

Save Discard Refresh

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Development Tools

Logic app designer

Logic app code view

Versions

API connections

Quick start guides

Settings

Workflow settings

Authorization

Access keys

Identity

Abbildung 13-49: Aktivieren der verwalteten Identität für eine Logik-App.

Es wird ein Dienstprinzipal mit dem gleichen Namen wie deine Logik-App erstellt. Navigiere zu dem Log Analytics-Arbeitsbereich, auf dem deine Microsoft Sentinel-Umgebung basiert, wähle „Zugriffssteuerung“ und gewähre der Logik-App die Berechtigung „**Microsoft Sentinel Responder**“, wie in Abbildung 13-50 dargestellt.

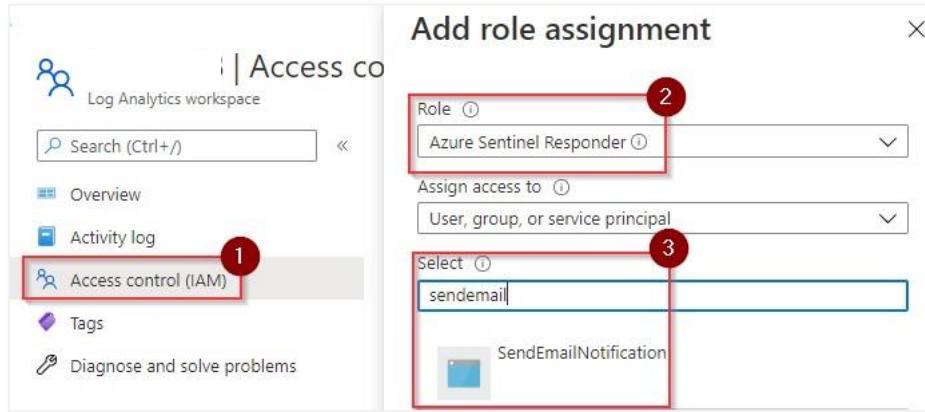
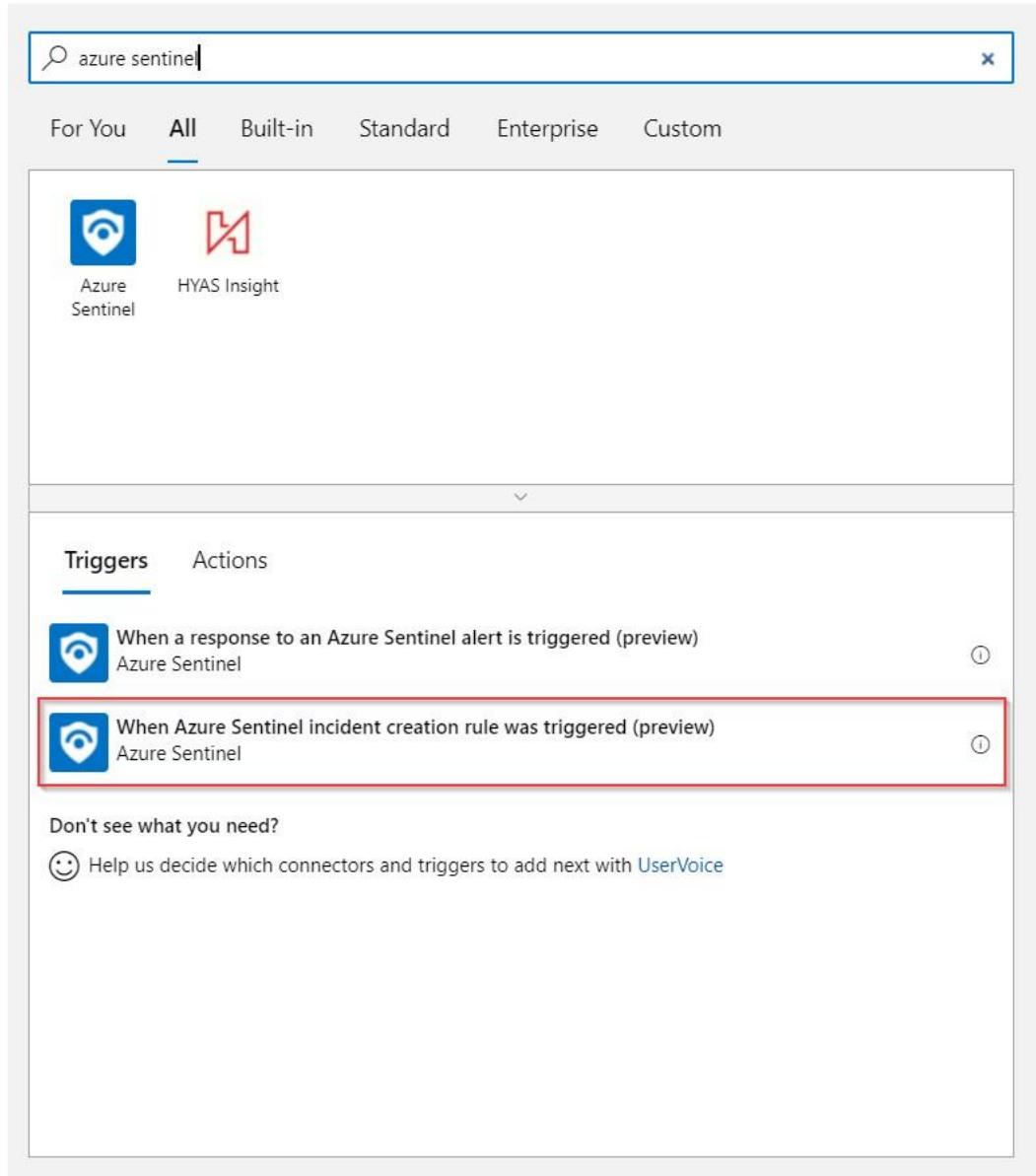


Abbildung 13-50: Hinzufügen der Microsoft Sentinel-Berechtigungen

**Hinweis:** In diesem speziellen Szenario liest und schreibt das Playbook Daten von Microsoft Sentinel. Wenn du nur Daten lesen musst, sollte die Rolle "Microsoft Sentinel Reader" verwendet werden. Um sich erfolgreich bei Microsoft Defender for Endpoint zu authentifizieren, müssen der verwalteten Identität außerdem die notwendigen Berechtigungen zugewiesen werden. Um herauszufinden, welche Berechtigungen für API-Aufrufe erforderlich sind, empfiehlt es sich, die API-Dokumentation zu konsultieren. In unserem Fall findest du diese in den [Microsoft Defender for Endpoint-Docs](#).

Gehe zurück zur Logik-App und wähle „**Logik-App-Designer**“. Jetzt ist es an der Zeit, den Trigger zu konfigurieren. Da wir jedes Mal eine E-Mail erhalten möchten, wenn ein Vorfall erstellt wird, wählen wir den Microsoft Sentinel-Vorfall-Trigger.

Dazu suchst du nach „Microsoft Sentinel“ und wählst den Trigger „*Wenn eine Microsoft Sentinel-Erstellungsregel ausgelöst wurde*“, wie in Abbildung 13-51 gezeigt.



The screenshot shows the Microsoft Logic Apps portal interface. At the top, there is a search bar with the text "azure sentinel". Below the search bar, there are tabs for "For You", "All", "Built-in", "Standard", "Enterprise", and "Custom". The "All" tab is selected.

In the main search results area, there are two items:

- Azure Sentinel (represented by a blue icon with a shield and a key)
- HYAS Insight (represented by a red icon with a stylized letter H)

Below the search results, there are two tabs: "Triggers" and "Actions". The "Triggers" tab is selected. Under the "Triggers" tab, there are two entries:

- When a response to an Azure Sentinel alert is triggered (preview) - Azure Sentinel
- When Azure Sentinel incident creation rule was triggered (preview) - Azure Sentinel

The second entry, "When Azure Sentinel incident creation rule was triggered (preview)", is highlighted with a red border. Below this section, there is a message: "Don't see what you need? 😊 Help us decide which connectors and triggers to add next with UserVoice".

Abbildung 13-51: Auswahl des richtigen Triggers

Als Nächstes musst du die Authentifizierung für die Logik-App konfigurieren. Wenn du auf „Anmelden“ klickst, würde die Authentifizierung bei Microsoft Sentinel auf der Grundlage der Anmeldeinformationen deines aktuellen Benutzerkontos erfolgen. Da wir das vermeiden wollen, wählst du „**Mit verwalteter Identität verbinden**“, wie in Abbildung 13-52 gezeigt.

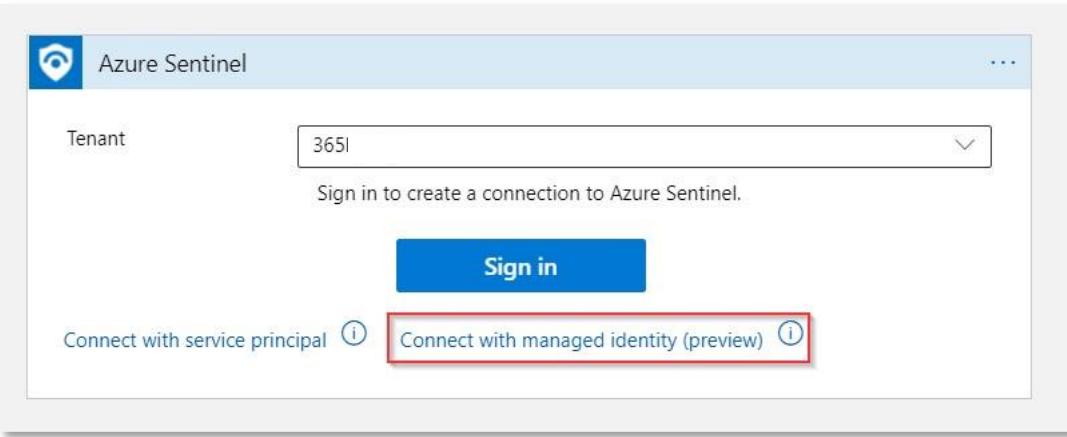


Abbildung 13-52: Verbindung über verwaltete Identität

Da die verwaltete Identität für die Logik-App bereits aktiviert ist, musst du nur noch einen Anzeigenamen für die Verbindung angeben und auf „Erstellen“ klicken. Dieser Name wird für die Azure-Ressource „API-Verbindung“ verwendet und hat keinen Einfluss auf die Authentifizierung.

Nachdem der Trigger definiert ist, kannst du nun Aktionen hinzufügen. Theoretisch kannst du so viele Aktionen integrieren, wie du möchtest. Logic Apps bietet eine Vielzahl integrierter Konnektoren, mit denen du Verbindungen zu nahezu jeder Ressource (in der Cloud oder lokal) herstellen kannst. In diesem Playbook halten wir die Anzahl der Aktionen gering, da wir lediglich ein Untersuchungspaket sammeln und einen Kommentar zum Microsoft Sentinel-Vorfall hinzufügen möchten.

Da wir auf einem von Microsoft Defender verwalteten Gerät agieren wollen, müssen wir zunächst die Microsoft Defender for Endpoint-Maschinen-ID abrufen. Dieser Vorgang erfolgt in zwei Schritten. Zuerst identifizieren wir die betroffene Maschine anhand der Vorfallswarnung. Danach rufen wir die korrekte Maschinen-ID über Defender for Endpoint ab. Um die betroffene Maschine zu identifizieren, sollte die Host-Entität in der Analyseregel konfiguriert sein. So kannst du den Hostnamen des Geräts direkt aus der Warnung oder dem Vorfall abrufen.

Suche nach „Microsoft Sentinel“ und wähle die Aktion „Entitäten – Hosts abrufen“. Klicke auf das leere Textfeld, damit der Bildschirm für dynamische Inhalte eingeblendet wird. Dieser Bildschirm (siehe Abbildung 13-53) ermöglicht es dir, Daten zu verwenden, die in vorherigen Schritten des Playbooks generiert wurden. Wähle „Entitäten“ als dynamischen Inhalt. Dieser Schritt nutzt als Eingabe alle Entitäten, die mit einem Vorfall verbunden sind, und gibt ausschließlich die Entität „Host“ zurück.

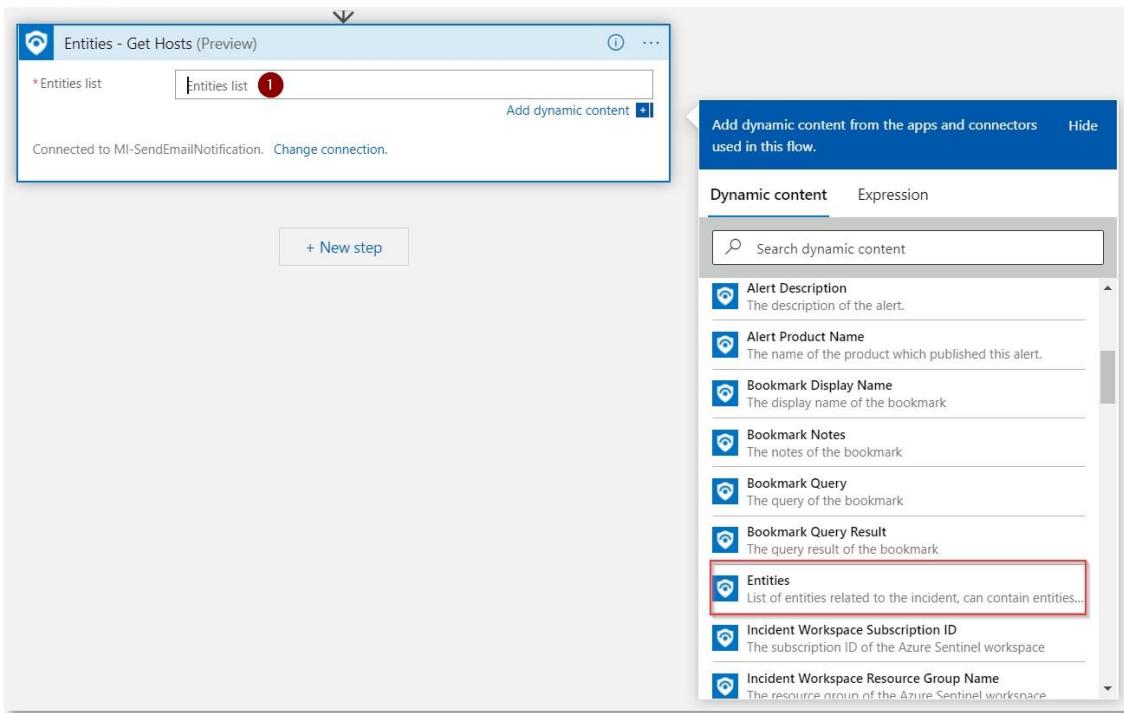


Abbildung 13-53: Auswahl des richtigen dynamischen Inhalts

Nachdem du die betroffene Maschine identifiziert hast, müssen wir in Microsoft Defender for Endpoint eine Aktion ausführen: das Untersuchungspaket sammeln. Wenn du auf einem Gerät in Microsoft Defender for Endpoint agieren willst, benötigst du die Maschinen-ID. Der erste Schritt dazu ist, diese ID mithilfe der Host-Entität abzurufen. Füge einen neuen Schritt hinzu, suche nach „**Microsoft Defender**“, wähle die Aktion „**Maschinen – Liste der Maschinen abrufen**“ und verbinde dich über die verwaltete Identität. Dieser Schritt nutzt einen Filter, um nach Microsoft Defender-Maschinen zu suchen. In unserem Fall verwenden wir den Anzeigenamen der Maschine, um die richtigen Informationen zu finden. Gib im Parameter „Ergebnisse filtern“ den Ausdruck `displayName eq ein` (dies ist der zu verwendende Filter) und füge den dynamischen Inhalt „**Hosts hostname**“ ein.

Wie du in Abbildung 13-54 sehen kannst, wird unsere Aktion durch die Ausgabe des Entitätsschritts automatisch in eine for-each-Schleife eingebettet. Das passiert, weil ein Vorfall eine oder mehrere Host-Entitäten enthalten kann.

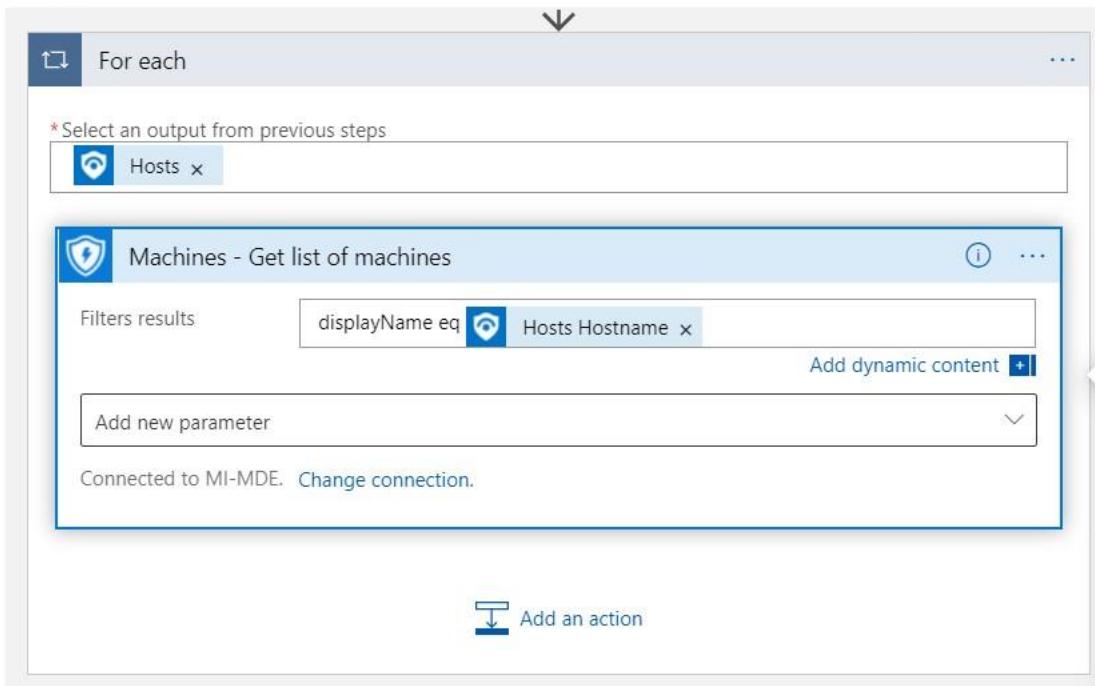


Abbildung 13-54: Konfigurieren der MDE-Aktionen

Der Schritt „Liste der Maschinen abrufen“ gibt die Maschinen-ID der Geräte zurück, die unserem Filter entsprechen. Diese ID kannst du dann im nächsten Schritt verwenden, in dem wir das Untersuchungspaket abrufen. Füge eine Aktion hinzu, suche erneut nach **Microsoft Defender** und wähle die Aktion „**Aktionen – Untersuchungspaket sammeln**“. Diese Aktion erfordert zwei Parameter: die Maschinen-ID und einen Kommentar.

Da der vorherige Schritt eine Liste von Maschinen zurückgibt, würde dieser Schritt automatisch in eine for-each-Schleife eingebettet werden, wenn du den Standard-dynamischen Inhalt des vorherigen Schritts auswählst. Ein weiteres Feature in Azure Logic Apps sind Ausdrücke. Sie ermöglichen dir, Daten zu manipulieren, die von vorherigen Schritten zurückgegeben wurden. Ausdrücke sind spezifisch für Azure Logic Apps und erlauben es dir, gezielte Funktionen auf diese Daten anzuwenden. Beispiele für Funktionen sind concat() und first(). Mit concat kannst du mehrere String-Variablen zu einer kombinieren, während first das erste Element eines Arrays zurückgibt. Eine vollständige Übersicht über Ausdrücke findest du [hier](#). Ausdrücke solltest du nur dann verwenden, wenn du den dynamischen Inhalt tatsächlich manipulieren möchtest. Um einen Ausdruck einzugeben, klicke auf das leere Feld neben „Maschinen-ID“, wähle „Ausdruck“, trage den Ausdruck ein und klicke auf „OK“. Die vollständige Konfiguration siehst du in Abbildung 13-55.

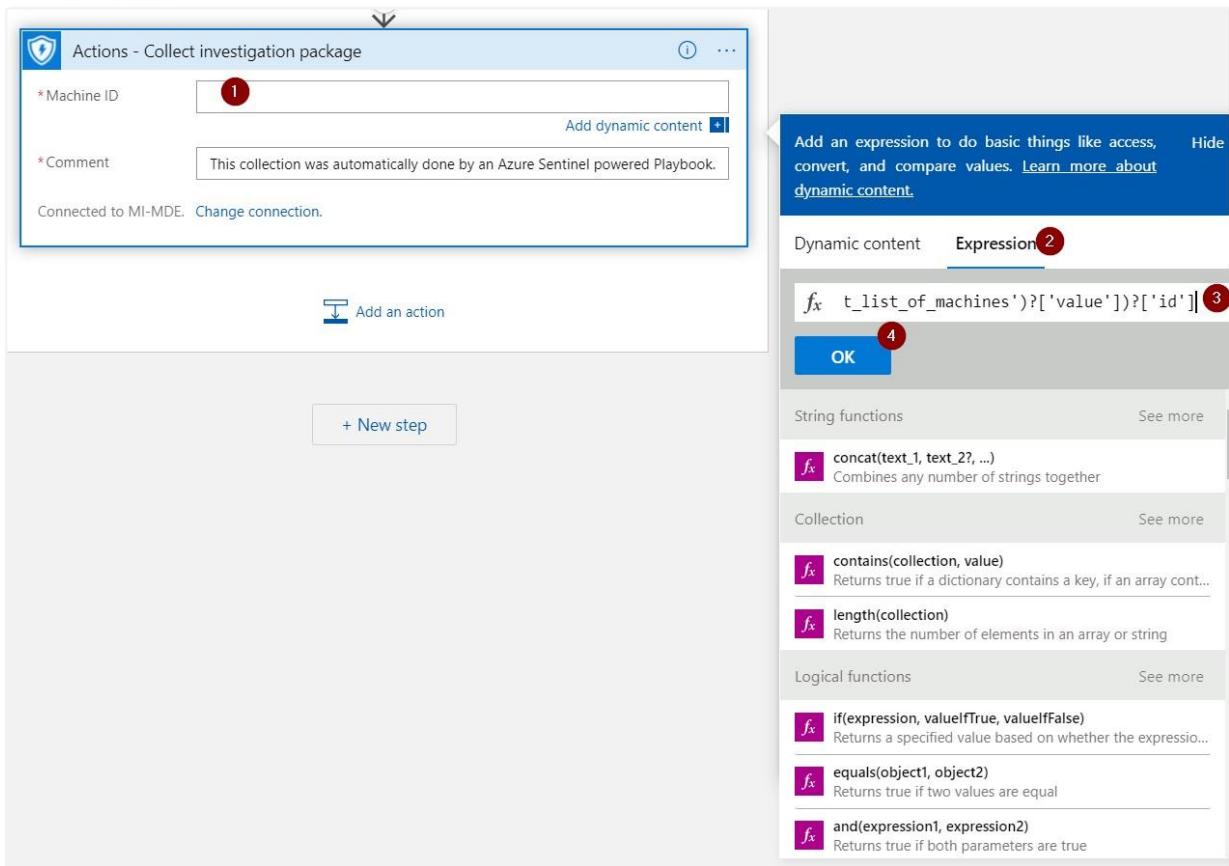


Abbildung 13-55: Konfigurieren von Ausdrücken

Der verwendete Ausdruck ist unten dargestellt. Dieser einfache Ausdruck greift auf das Ergebnis des vorherigen Schritts zu (ein Array) und verwendet die first-Funktion, um nur das erste Element im Array zu verarbeiten. Von diesem ersten Element wird dann die ID-Variable genutzt.

```
first(body('Machines__Get_list_of_machines')?['value'])?['id']
```

Dieser Schritt sammelt das Untersuchungspaket für alle Hosts, die mit dem Vorfall verknüpft sind. Der letzte Schritt besteht darin, das SOC-Team darüber zu informieren, dass diese Aktion ausgeführt wurde. Suche nach der Aktion „**Kommentare zum Vorfall hinzufügen**“, um einen Kommentar zum Vorfall hinzuzufügen. Diese Aktion solltest du außerhalb der for-each-Schleife platzieren, damit der Kommentar nur einmal geschrieben wird. Dafür benötigst du die **Vorfall-ID**, die im dynamischen Inhalt verfügbar ist, da sie im Trigger des Vorfalls enthalten ist. Der finale Schritt ist in Abbildung 13-56 dargestellt.

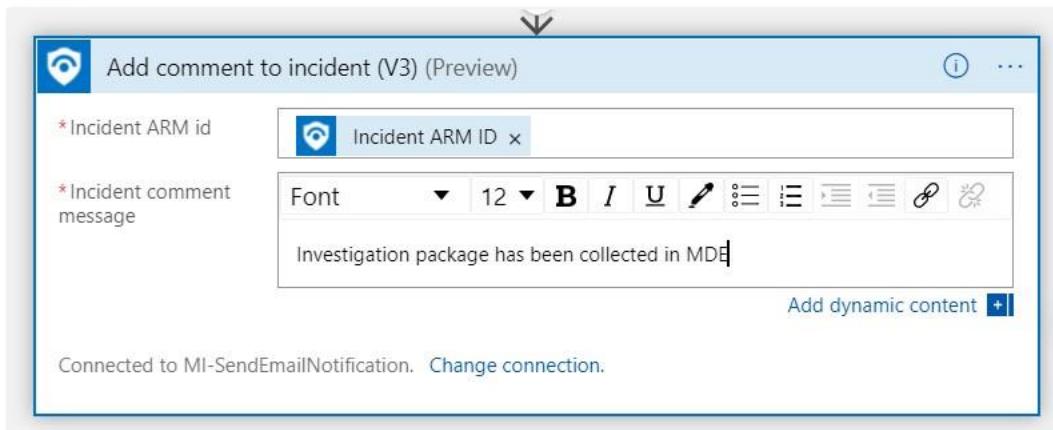


Abbildung 13-56: Hinzufügen eines Kommentars über ein Playbook

Wenn das Playbook fertig ist, klicke auf „**Speichern**“, um die Erstellung abzuschließen.

Jetzt ist es an der Zeit, zu Microsoft Sentinel zurückzukehren und unsere Automatisierungsregel zu erstellen. Navigiere zur Automatisierung und wähle „**Erstellen > Neue Regel**“. Gib der Regel einen aussagekräftigen Namen und konfiguriere die folgenden Bedingungen:

- Wähle **alle** für die Bedingung des Analysenamens, da wir diese Regel unabhängig von der konkreten Analyseregel ausführen wollen.
- Stelle die **Bedingung des Incident-Anbieters** so ein, dass das Playbook nur bei Incidents von Microsoft Defender ausgeführt wird.
- Filtere nach **hohem Schweregrad**, damit das Playbook nur bei schwerwiegenden Incidents ausgeführt wird, bei denen ein Untersuchungspaket sinnvoll ist.

Wähle abschließend „**Playbook ausführen**“ als Aktion und wähle das gerade erstellte Playbook aus. Die vollständige Konfiguration findest du in Abbildung 13-57.

Automation rule name  
Collect Investigation Package ✓

Trigger  
When incident is created

Conditions

If  
Analytic rule name Contains All

And  
Incident provider Equals Microsoft 365 Defen... ⚡

And  
Severity Equals High ⚡

+ Add condition

Actions ⓘ

Run playbook ⚡

SendEmailNotification  
Visual Studio Enterprise Subscription / Automation ⚡

+ Add action

Abbildung 13-57: Erstellen einer Automatisierungsregel.

## Microsoft Sentinel Systemzustand

Nachdem du deine Microsoft Sentinel-Umgebung eingerichtet hast, ist die Arbeit noch nicht abgeschlossen. Auch wenn Microsoft Sentinel ein stabiles Produkt ist, können immer wieder Probleme auftreten. Zum Beispiel kann eine Analyseregel Validierungsfehler enthalten oder ein Datenkonnektor keine Daten mehr erfassen. Es ist essenziell, dass du deine Umgebung kontinuierlich überwachst und Benachrichtigungen einrichtest, sobald Probleme auftreten.

Bevor wir auf die Details zur Systemüberwachung und zu den von Microsoft Sentinel bereitgestellten Funktionen eingehen, möchte ich kurz auf die Verantwortlichkeiten innerhalb des SOC eingehen. In vielen Organisationen sehe ich, dass Probleme als Sentinel-Incidents erstellt werden. Davon bin ich kein Fan, denn Incidents sollten sich vollständig auf (potenzielle) Sicherheitsprobleme konzentrieren. Ein Incident ist etwas, das ein SOC-Analyst untersuchen muss – ein Systemproblem hingegen ist Sache des Sentinel-Architekten oder Azure-Ingenieurs. Deshalb bevorzuge ich alternative Methoden zur Behandlung solcher Probleme:

- Erstellen eines Problems in deinem ITSM-Tool.
- Senden einer E-Mail an das richtige Team.
- Einrichten von [Log Analytics-Warnungen](#).

In einer Microsoft Sentinel-Umgebung habe ich einige Bereiche identifiziert, in denen eine Überwachung erforderlich sein könnte.

- Datenkonnektoren
  - Signifikante Zunahme der Datenerfassung
  - Authentifizierungsfehler des Datenkonnektors
  - Ausfall des Datenkonnektors
- Analyseregeln
  - Analyseregel konnte nicht ausgeführt werden
- Automatisierung
  - Automatisierungsregel konnte nicht ausgeführt werden
  - Playbook ist fehlgeschlagen

Hier kommt die **Microsoft Sentinel Health Monitoring**-Funktion ins Spiel. Wenn du diese Funktion aktivierst, wird eine neue Tabelle in deinem Log Analytics-Arbeitsbereich angezeigt – sie heißt „SentinelHealth“ und dient dazu, mögliche Systemprobleme abzufragen.

Bevor du diese Funktion verwenden kannst, musst du sie in den Microsoft Sentinel-Einstellungen aktivieren. Öffne dazu deinen Microsoft Sentinel-Arbeitsbereich, wähle „Einstellungen“ und klicke oben auf die Registerkarte „Einstellungen“. Wähle anschließend „Health Monitoring“ aus der Liste und klicke auf „Diagnoseeinstellungen konfigurieren“. Es erscheint die typische Benutzeroberfläche für Diagnoseeinstellungen in Azure, mit der du Protokolle in einem Log Analytics-Arbeitsbereich, einem Speicherkonto oder einem Event Hub speichern kannst. Wie in **Abbildung 13-57** zu sehen, gibt es derzeit drei Datenkategorien: DataConnectors, Automation und Analytics. Ich empfehle dir, stattdessen „allLogs“ auszuwählen. So stellst du sicher, dass auch zukünftige, unterstützte Workloads automatisch eingeschlossen werden. Microsoft investiert stark in diese Funktion, daher gehe ich davon aus, dass sie in den kommenden Monaten weiter ausgebaut wird. Wähle anschließend deinen Microsoft Sentinel Log Analytics-Arbeitsbereich als Ziel aus. Dadurch befinden sich alle Daten an einem zentralen Ort.

## Diagnostic setting ...

Save Discard Delete Feedback

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

Diagnostic setting name	Sentinel Health Monitoring
<b>Logs</b>	
Category groups ⓘ	
<input checked="" type="checkbox"/> allLogs	
Categories	
<input checked="" type="checkbox"/> Analytics	
<input checked="" type="checkbox"/> Automation	
<input checked="" type="checkbox"/> Data Collection - Connectors	
<b>Destination details</b>	
<input checked="" type="checkbox"/> Send to Log Analytics workspace	
Subscription	Visual Studio Enterprise Subscription – MPN
Log Analytics workspace	TCTest-Sentinel (westeurope)
<input type="checkbox"/> Archive to a storage account	
<input type="checkbox"/> Stream to an event hub	
<input type="checkbox"/> Send to partner solution	

Abbildung 13-58: Einrichten von Microsoft Sentinel-Diagnoseeinstellungen.

Sobald diese Konfiguration abgeschlossen ist, beginnt die Tabelle „SentinelHealth“, Daten zu erfassen – sowohl bei erfolgreichen als auch bei fehlgeschlagenen Aktionen.

## Datenkonnektoren

Die Überwachung der Datenkonnektoren ist vermutlich eine der wichtigsten Aufgaben. Wenn keine Daten erfasst werden, könnten dir kritische Sicherheitsvorfälle entgehen. Deshalb solltest du sicherstellen, dass alle Datenkonnektoren fehlerfrei arbeiten und Daten an Microsoft Sentinel senden.

Derzeit ist die [Liste der unterstützten Datenkonnektoren](#) für die Health-Funktion noch stark eingeschränkt und umfasst nur einige Microsoft-Konnektoren. Die Daten in dieser Tabelle lassen sich jedoch einfach abfragen. Mit einer einfachen KQL-Abfrage kannst du beispielsweise alle Datenkonnektoren identifizieren, bei denen ein Fehler aufgetreten ist.

```
 SentinelHealth
| where Status != "Success"
```

Leider deckt diese Funktion nur einen extrem kleinen Anwendungsfall ab, denn die Microsoft Sentinel Health-Funktion unterstützt derzeit folgenden Szenarien nicht:

- Überwachung für alle verbundenen Datenkonnektoren
- Erkennung für Erfassungsspitzen

Wenn du andere Datenkonnektoren überwachen möchtest, musst du dir eigene Lösungen überlegen. Dabei stehen dir je nach Typ des Datenkonnektors unterschiedliche Optionen zur Verfügung.

Wenn du beispielsweise einen CEF- oder Syslog-Connector verwendest, solltest du die Integrität deiner Forwarder-VMs überwachen. Wenn kürzlich kein Heartbeat gesendet wurde, ist es möglich, dass die VM aktuell ausgefallen ist oder Verbindungsprobleme bestehen. Die Überwachung dieser Systeme kannst du über die **Heartbeat-Tabelle** in Log Analytics umsetzen. Mit einer einfachen Abfrage lässt sich der Zeitpunkt ermitteln, zu dem ein bestimmter Computer zuletzt eine Verbindung zu Log Analytics hergestellt hat.

### Heartbeat

```
| summarize max(TimeGenerated) by Computer
```

Für andere, nicht CEF/Syslog-basierte Konnektoren stehen dir ein paar weitere, wenn auch nicht perfekte, Möglichkeiten offen:

- Verwenden des integrierten [Datenkonnektor-Systemüberwachungs-Workbooks](#).
- Einrichten einer KQL-Abfrage, um eine Warnung zu erstellen, wenn innerhalb der letzten Stunde keine Daten für eine bestimmte Tabelle eingegangen sind. Dies ist eine schnelle und einfache Lösung zur Überwachung potenzieller Anomalien, aber es ist nicht perfekt. Dadurch werden auch Warnungen erstellt, wenn der Datenkonnektor keine Daten zum Senden hat.
- Wenn du API-basierte Datenkonnektoren wie Functions oder Playbooks verwendest, stehen einige native Azure-Tools zur Verfügung, eines davon ist [Application Insights](#), das ich sehr empfehle.

Neben der Überwachung von Ausfällen bei der Datenübertragung solltest du auch Erfassungsspitzen im Blick behalten. Dabei handelt es sich um eine Situation, in der ein Datenkonnektor plötzlich deutlich mehr Ereignisse in einem kurzen Zeitraum sendet als üblich. Auch wenn das auf den ersten Blick unkritisch erscheint, gibt es zwei gute Gründe, warum du diese Spitzen überwachen solltest:

- Erfassungsspitzen könnten ein Angreifer sein, der versucht, dein SIEM mit Dummy-Daten zu überschwemmen.
- Eine starke Zunahme der Menge der erfassten Daten wird deine Microsoft Sentinel-Kosten erheblich erhöhen.

Microsoft bietet dafür zwar noch keine perfekte Lösung, aber zwei Optionen stehen dir zur Verfügung:

- Verwendung des [offiziellen Playbooks](#) zum Senden von E-Mails bei erkannten Erfassungsspitzen.
- Einrichten einer [täglichen Erfassungsobergrenze in Log Analytics](#).

Auch wenn diese Funktion zunächst vielversprechend klingt, solltest du bei der Aktivierung vorsichtig sein. Wenn du eine Obergrenze für die Datenerfassung festlegst, werden nach Erreichen des Tageslimits keine Daten mehr erfasst. In den meisten Organisationen steigt die Menge der erfassten Daten mit der Zeit langsam an. Wenn du diese Funktion aktiviert hast, empfehle ich dir, sie regelmäßig zu überwachen.

## Automatisierung

Wenn es um die Überwachung deiner Automatisierung mit Microsoft Sentinel geht, gibt es zwei Aspekte, die du im Blick behalten solltest:

- Automatisierungsregeln
- Playbooks

Automatisierungsregeln werden ebenfalls durch die Microsoft Sentinel Health-Funktion unterstützt. Du kannst damit überprüfen, ob eine bestimmte Regel ausgeführt wurde. So kannst du beispielsweise mit einer einfachen Abfrage feststellen, ob eine Automatisierungsregel tatsächlich gelaufen ist.

```
 SentinelHealth
| where OperationName == "Automation rule run"
```

Obwohl das ein Schritt in die richtige Richtung ist, fehlt bei Automatisierungsregeln noch eine entscheidende Funktion: ein “What If”-Modus oder ein gründliches Auditing. Wenn du komplexe Automatisierungsregeln mit mehreren Bedingungen eingerichtet hast, ist es oft schwer zu erkennen, warum eine Regel nicht ausgelöst wurde. Möglicherweise erfüllt ein Incident nicht die nötigen Bedingungen oder die Regel ist falsch konfiguriert. Mithilfe einer gründlichen Überprüfung kannst du herausfinden, welche Bedingungen erfüllt waren und welche nicht.

Für die Überwachung von Playbook-Ausführungen stehen dir mehrere Optionen zur Verfügung:

- Du kannst Warnungen für Logic App-Ausführungsfehler einrichten, indem du die [Diagnoseeinstellungen](#) konfigurierst. So lassen sich Log Analytics-Warnungen erstellen, wenn eine Logic App nicht wie vorgesehen ausgeführt werden konnte.
- Wenn du dir einen Überblick über den allgemeinen Zusatznd deiner Playbooks verschaffen möchtest, empfehle ich dir das Workbook namens **Playbooks Health Monitoring**. Es bietet dir einen guten Einblick und hilft dabei, Playbooks mit einer hohen Fehlerrate zu identifizieren.

Systemprobleme bei Analyseregeln sind möglicherweise nicht sofort ersichtlich, aber du solltest ein paar Dinge beachten. Wenn eine Regel bestimmte Bedingungen nicht erfüllt – zum Beispiel

das Überschreiten des Abfragezeitlimits oder den Versuch, auf nicht vorhandene Tabellen zuzugreifen – kann sie fehlschlagen oder sogar automatisch deaktiviert werden.

Solche Fälle sind zwar selten, sollten aber jedem Microsoft Sentinel-Administrator bekannt sein, damit man entsprechend reagieren kann.

Microsoft stellt eine [vollständige Liste](#) bereit, in der mögliche Fehlerursachen und die Bedingungen für die Deaktivierung einer Regel aufgeführt sind. Auf GitHub findest du auch ein [Playbook](#), das diese Art von Fehlern überwacht und E-Mail-Benachrichtigungen versendet, wenn Probleme auftreten. Leider sind diese Fehlerarten nicht in der Microsoft Sentinel Health Monitoring-Funktion enthalten.

Mit der Microsoft Sentinel Health-Funktion kannst du jedoch auch deine Analyseregeln überwachen. Dabei stehen dir zwei zentrale Funktionen zur Verfügung:

- Du kannst erkennen, ob eine Regel nicht ausgeführt oder deaktiviert wurde.
- Du erhältst Informationen, wenn Änderungen an deinen Analyseregeln vorgenommen wurden. Zwar sind diese Änderungen auch über die Azure Activity-Datenkonnektoren abrufbar, doch die Sentinel Health-Funktion konzentriert sich ausschließlich auf Änderungen innerhalb von Sentinel. Diese Informationen werden in der Tabelle „SentinelAudit“ gespeichert.

## Automatisierung

Das Microsoft Sentinel-Portal ist ein guter Einstiegspunkt, weil es eine intuitive Benutzeroberfläche bietet. Mit zunehmender Erfahrung wirst du jedoch feststellen, dass viele Aufgaben automatisiert werden sollten, um Zeit zu sparen, die du sonst im Portal aufwenden würdest.

Azure bietet dir dafür zahlreiche Automatisierungsfunktionen – darunter APIs, PowerShell-Module und ARM-Vorlagen. Da es viele verschiedene Möglichkeiten gibt, kann es anfangs schwierig sein, zu entscheiden, welche Methode wann sinnvoll ist. Wir werden alle Optionen im Detail durchgehen und sie miteinander vergleichen.

## API

Da Microsoft Sentinel auf Log Analytics basiert, stehen dir auch APIs zur Verfügung, die speziell für Log Analytics und Sentinel gedacht sind. Insgesamt kannst du vier APIs nutzen:

- Microsoft Security Graph API
- Data Collector API
- Log Analytics API

- Microsoft Sentinel Management API

## Microsoft Security Graph API

Wir haben das Security Graph im Abschnitt über Threat Intelligence bereits kurz erwähnt. Die Security Graph API ist nicht spezifisch für Microsoft Sentinel, sondern deckt alle Microsoft-Sicherheitsprodukte ab, wie Microsoft Defender for Endpoint, Microsoft 365 Defender, Entra ID Identity Protection und weitere.

Neben der Interaktion mit Threat-Intelligence-Indikatoren und dem Abrufen von Secure Scores kannst du über diese API auch Warnungen abrufen, einschließlich derjenigen von Microsoft Sentinel. Der Hauptvorteil dieser API besteht darin, dass du Warnungen aus verschiedenen Microsoft-Sicherheitsprodukten mit einer einzigen Abfrage abrufen kannst. Leider unterstützt die Security Graph API nur Warnungen und keine Incidents. Deshalb ist sie im Kontext von Microsoft Sentinel nicht besonders nützlich, abgesehen von der Arbeit mit Threat-Indikatoren.

## Data Collector API

Die Data Collector API ist speziell für Log Analytics konzipiert und ermöglicht es dir, eigene benutzerdefinierte Daten an Log Analytics zu senden. Du kannst damit Daten in benutzerdefinierte Tabellen einfügen – das Einfügen in integrierte Tabellen wird jedoch nicht unterstützt. Sie eignet sich ideal zum Erstellen eigener benutzerdefinierter API-Connectors, die z. B. in einer Azure Logic App ausgeführt werden. Diese API ist die einzige Möglichkeit, Protokolldaten in Microsoft Sentinel zu schreiben. Du wirst sie häufig in Verbindung mit verschiedenen Datenconnectors sehen. Eine Schritt-für-Schritt-Anleitung zur Nutzung findest du in der [Microsoft-Dokumentation](#).

Wichtig zu wissen ist, dass die Data Collector API Log Analytics über die Arbeitsbereichs-ID und den zugehörigen Schlüssel autorisiert. Eine Authentifizierung per Entra ID Service Principal wird nicht unterstützt. Du kannst die API aber auch als [Connector innerhalb einer Logic App](#) verwenden, wodurch du benutzerdefinierte Daten einfach in eigene Playbooks einbinden kannst.

## Log Analytics API

Während die Data Collector API das Schreiben von Daten ermöglicht, dient die Log Analytics API dem Lesen von Daten. Die [einzige Funktionalität](#) dieser API besteht darin, Daten aus Log Analytics abzufragen, und ermöglicht einige schöne Integrationen:

- Du kannst sie verwenden, um zusätzliche Daten zu einem Incident innerhalb von Playbooks abzurufen.

- Es gibt einen nativen Connector, mit dem du Microsoft Sentinel-Berichten in PowerBI erstellen kannst.

Im Gegensatz zur Data Collector API unterstützt diese API die Authentifizierung über einen Entra ID Service Principal. Dafür musst du eine neue App-Registrierung mit der Berechtigung „Data.Read“ erstellen. Anschließend musst du dem zugehörigen Service Principal die Rolle „Log Analytics Reader“ für den betreffenden Microsoft Sentinel-Arbeitsbereich zuweisen.

Wie bei der Data Collector API steht auch hier ein [Azure Logic Apps Connector](#) zur Verfügung, mit dem du einfach Daten aus Log Analytics abfragen kannst.

## Microsoft Sentinel Management API

Die Microsoft Sentinel Management API ist die wichtigste REST-API speziell für Microsoft Sentinel. Mit dieser API kannst du mit Microsoft Sentinel-spezifischen Ressourcen wie Incidents, Datenconnectors, Watchlists und mehr interagieren. Die API ist Teil der generischen Azure Management API, die verschiedene APIs für alle Azure-Ressourcen bereitstellt.

Die Hauptdokumentation dieser API ist etwas versteckt und nicht leicht zu finden. Sie ist in der [Azure Rest API-Dokumentation](#) enthalten und wird dort oft als SecurityInsights-API bezeichnet. Diese Dokumentation zeigt den jeweils aktuellen Stand der stabilen Version der API. Microsoft Sentinel bietet in der Regel mehrere API-Versionen an: Während das stabile Release alle derzeit allgemein verfügbaren Funktionen enthält, bietet das Vorschau-Release zusätzlich Funktionen, die sich noch in der öffentlichen Vorschau befinden. Diese Vorschau-Releases sind nicht in der offiziellen REST-API-Dokumentation enthalten, aber es gibt ein [GitHub-Repository](#) mit dem Vorschau-Branch und einigen grundlegenden Informationen dazu.

Diese API basiert auf der rollenbasierten Zugriffssteuerung von Azure (Azure RBAC) für die Authentifizierung. Das bedeutet, dass du ein gültiges Entra ID-Zugriffstoken bereitstellen musst. Ein solches Token kannst du entweder im Benutzerkontext (delegierte Berechtigungen) oder als Hintergrundaufgabe (Anwendungsberechtigungen) erwerben. Nachdem du die App-Registrierung erstellt hast, musst du die passende Rolle für den Microsoft Sentinel-Arbeitsbereich zuweisen. Welche Rolle du zuweisen solltest, hängt vollständig von den Aktionen ab, die die API ausführen wird. Wenn sie mit Incidents interagieren muss, solltest du die Microsoft Sentinel Responder-Rolle konfigurieren. Wenn du Analyseregeln über die API aktualisierst, ist die Microsoft Sentinel Contributor-Rolle erforderlich.

Nachdem du der App-Registrierung die erforderlichen Berechtigungen zugewiesen hast, kannst du den folgenden PowerShell-Code verwenden, um ein Entra ID-Zugriffstoken zu erhalten. Aktualisiere die Variablen ClientID, ClientSecret und TenantID entsprechend der von dir erstellten App-Registrierung.

```
$body=@{
    client_id=<ClientID>
    client_secret=<ClientSecret>
    resource="https://management.azure.com"
    grant_type="client_credentials"
}

$accesstoken           =             Invoke-WebRequest
"https://login.microsoftonline.com/<TenantID>/oauth2/token"
"application/x-www-form-urlencoded" -Body $body -Method Post

$authHeader = @{
    'Content-Type'='application/json'
    'Authorization'="Bearer " + $accessToken.access_token
    'ExpiresOn'=$accessToken.expires_in
    'Content-Encoding'='gzip' }
```

Nach erfolgreicher Authentifizierung kannst du mit Microsoft Sentinel interagieren und die benötigten API-Aufrufe ausführen. Während du API-Aufrufe machst, gibt es ein paar Dinge, die du beachten solltest:

- Die Eigenschaft **etag** ist typisch für die Azure Management Rest API und stellt die Version einer bestimmten Ressource innerhalb von Azure dar. Wenn du eine vorhandene Ressource (wie Incidents, Analyseregeln usw.) aktualisierst, musst du zuerst die aktuelle etag abrufen und diesen Wert dann im POST-Body angeben. Um die etag-Eigenschaft abzurufen, musst du eine GET-Anfrage gegen dieselbe URL ausführen, die du für die POST-Anfrage verwenden wirst.
- Die **URL** für deinen API-Aufruf muss mit den richtigen Variablen aus deiner eigenen Umgebung aktualisiert werden. Für jeden API-Aufruf musst du die folgenden Informationen aus deiner Microsoft Sentinel-Umgebung angeben: Abonnement-ID, Ressourcengruppe und den Namen deines Microsoft Sentinel-Arbeitsbereichs.
- Jede Ressource in Microsoft Sentinel hat eine eindeutige **GUID**. Es kann nicht zwei Analyseregeln mit derselben GUID geben. Wenn du einen vorhandenen Incident oder eine Regel aktualisierst, solltest du die GUID dieses spezifischen Incidents oder dieser Regel angeben. Wenn du einen neuen Incident erstellst, musst du eine neue GUID generieren, die in deiner Microsoft Sentinel-Umgebung noch nicht vorhanden ist. Dazu kannst du den PowerShell-Befehl New-Guid verwenden.

```
$incidentbody = @{
    "etag": "\"0300bf09-0000-0000-0000-5c37296e0000\"",
    "properties": {
        "lastActivityTimeUtc": "2019-01-01T13:05:30Z",
        "firstActivityTimeUtc": "2019-01-01T13:00:30Z",
```

```
"description": "This is a demo incident",
"title": "My incident",
"severity": "High",
"status": "New"
}
}

Invoke-WebRequest -Uri "
https://management.azure.com/subscriptions/<SubscriptionID>/resourceGroups
/<ResourceG
roup>/providers/Microsoft.OperationalInsights/workspaces//<WorkspaceName>/
providers/Microsoft.SecurityInsights/incidents/bc464691-c74e-4ac4-9762-
bf76b20d512d?apiVersion=2021-04-01" -ContentType "application/x-
www-form-urlencoded" -Body $incidentbody -Method Post -Headers
$authHeader
```

## PowerShell

Viele Systemadministratoren ziehen es vor, ein PowerShell-Modul zu verwenden, anstatt sich mit einer bestimmten API vertraut zu machen. Obwohl ich glaube, dass jeder Microsoft Sentinel-Administrator mit der API vertraut sein sollte, gibt es auch ein PowerShell-Modul. Der Hauptvorteil der Verwendung der API besteht darin, dass Preview-Funktionen in der Regel unterstützt werden, da auch das Azure-Portal auf dieser API basiert. Die PowerShell-Module unterstützen nicht immer die neuesten Funktionen.

Derzeit sind zwei PowerShell-Module verfügbar:

- Das offizielle **Az.SecurityInsights** PowerShell-Modul
- Das von der Community betriebene Modul **AzSentinel**

Im ersten Jahr der Existenz von Microsoft Sentinel war das einzige verfügbare PowerShell-Modul AzSentinel. Es wurde von [Pouyan Khabazi](#), einem in den Niederlanden ansässigen Azure MVP, entwickelt. Das Modul ist auf [GitHub](#) verfügbar und enthält sowohl allgemein verfügbare als auch Vorschaufunktionen. Es bietet ein paar hilfreiche Features, wie das Cmdlet Import-AzSentinelAlertRule, mit dem du mehrere Analyseregeln gleichzeitig importieren kannst. Das Modul unterstützt sowohl JSON- als auch YAML-Dateiformate.

Im Mai 2021 wurde das Az.SecurityInsights-Modul allgemein verfügbar. Es ist das erste offizielle PowerShell-Modul für Microsoft Sentinel. Es basiert auf der stabilen Version der API und enthält eine begrenzte Anzahl von Funktionen. Die vollständige Dokumentation findest du in der [Microsoft-Dokumentation](#), und Beispiele sowie Anwendungsfälle sind auf [GitHub](#) verfügbar. Der Hauptvorteil dieses Moduls ist, dass es offiziell von Microsoft unterstützt wird.

Der Nachteil ist, dass es auf der stabilen Version der API basiert, wodurch einige neue Funktionen unter Umständen noch nicht verfügbar sind.

Ich empfehle dir, mit dem offiziellen Modul zu beginnen und zu prüfen, ob es alle von dir benötigten Funktionen bereitstellt. Wenn du auf Einschränkungen stößt, kannst du jederzeit zum AzSentinel-Modul wechseln. Obwohl es nicht offiziell unterstützt wird, wird es aktiv gepflegt, und alle Probleme, auf die ich gestoßen bin, wurden schnell behoben, nachdem ich ein Issue im GitHub-Repo des Moduls erstellt hatte.

## ARM-Vorlagen

Obwohl Microsoft Sentinel auf Azure Resource Manager basiert, können nicht alle Microsoft Sentinel-Ressourcen über ARM-Vorlagen erstellt werden. Ab Juni 2021 hat Microsoft begonnen, ARM-Unterstützung für verschiedene Microsoft Sentinel-Ressourcen freizugeben. Zum Zeitpunkt der Erstellung dieses Dokuments unterstützen die folgenden Ressourcen ARM-Vorlagen:

- Watchlists
- Analyseregeln
- Hunting-Regeln
- Playbooks
- Arbeitsmappen
- Parser
- Automatisierungsregeln

Der Hauptvorteil der ARM-Unterstützung besteht darin, dass du die Erstellung von Microsoft Sentinel-Ressourcen einfach automatisieren kannst, ohne die verschiedenen APIs kennen zu müssen. ARM-Vorlagen lassen sich zudem viel einfacher importieren und exportieren als manuell zusammengestellte API-Abfragen.

Wie du die ARM-Vorlage erstellst, hängt stark vom jeweiligen Ressourcentyp ab.

- Analyseregeln können über das Portal in eine ARM-Vorlage exportiert werden. Navigiere zu Analyseregeln, wähle die Regel aus, die du exportieren möchtest, und klicke oben auf die Schaltfläche "Exportieren".
- Arbeitsmappen sind Teil von Azure Monitor und können auch einfach über [die Benutzeroberfläche](#) exportiert werden.
- Wenn du die Vorlage einer Logic App nativ exportierst, enthält sie viele Verweise auf das Abonnement, in dem sie bereitgestellt wurde. Auf [GitHub](#) ist ein Skript verfügbar, um die exportierte Vorlage zu normalisieren.
- Automatisierungsregeln und Parser können mit [diesem Skript](#) exportiert werden, das von Shreedar, einem Mitglied des Microsoft Sentinel-Kundenteams, erstellt wurde.

Ich persönlich freue mich sehr auf die Zukunft, da dies viele Möglichkeiten in Bezug auf Automatisierung und CI/CD-Unterstützung eröffnet. ARM-Vorlagen werden dir das Leben deutlich erleichtern!

## Wahl der richtigen Automatisierungsmethode

Wenn du eine Analyseregel programmgesteuert erstellen möchtest, gibt es viele verschiedene Möglichkeiten, dies zu tun:

1. ARM-Vorlagen
2. Az.SecurityInsights PowerShell-Modul
3. AzSentinel PowerShell-Modul
4. Azure Management API

Die oben genannte Reihenfolge entspricht auch meiner persönlichen Präferenz bei der Auswahl der passenden Automatisierungsmethode. Meine erste Wahl sind ARM-Vorlagen, da sie keine speziellen Kenntnisse über Module oder APIs voraussetzen. Sie können einfach über das Azure-Portal, die API, PowerShell oder Azure DevOps bereitgestellt werden. ARM-Vorlagen sind ideal zum Erstellen von Ressourcen, bieten jedoch keine Möglichkeit zur Interaktion mit Incidents.

Für Aufgaben, die nicht durch ARM-Vorlagen abgedeckt werden, verwende ich PowerShell. Hier ziehe ich zunächst das offizielle PowerShell-Modul vor. Sollte dieses bestimmte Aufgaben nicht unterstützen, greife ich auf das AzSentinel-Modul zurück.

Die letzte Option ist die Azure Management API. Sie erfordert zwar den größten Aufwand zur Einrichtung und Konfiguration der passenden API-Aufrufe, bietet aber nahezu vollständige Funktionalität – auch für Features in öffentlicher Vorschau. Daher wirst du sie wahrscheinlich relativ häufig einsetzen.

Natürlich ist dies alles eine Frage der persönlichen Vorliebe. Wähle die Methode, mit der du am besten zurechtkommst.

## Automatisierte Bereitstellung

Wenn du deine Microsoft Sentinel-Umgebung erweiterst, gibt es einige sinnvolle Anwendungsfälle für die Arbeit mit „Microsoft Sentinel as Code“. Dabei werden Sentinel-Ressourcen automatisiert bereitgestellt. Typische Szenarien sind:

- Wenn du **mehrere Arbeitsbereiche** verwaltest, ist es wesentlich effizienter, Ressourcen wie Analyseregeln automatisiert bereitzustellen, anstatt sie manuell mehrfach zu erstellen.

- In größeren Organisationen gelten oft strengere mit dem **Vier-Augen-Prinzip**. Jede Änderung von mindestens zwei Personen freigegeben werden. Eine automatisierten Bereitstellung ermöglicht es dir, Änderung zur Genehmigung einzureichen, bevor sie produktiv geschaltet werden.
- Durch die Integration **mehrerer Stufen** in deinem Workflow kannst du sicherstellen, dass Ressourcen zunächst in einer Testumgebung bereitgestellt und geprüft werden, bevor sie in die Produktion übergehen.

Diese Herausforderungen kannst du durch den Einsatz automatisierter CI/CD-Tools wie Azure DevOps oder GitHub Actions meistern – statt manuelle Änderungen im Portal vorzunehmen. CI/CD steht für Continuous Integration und Continuous Deployment. Dabei werden Änderungen automatisch getestet und validiert (CI), bevor sie automatisiert bereitgestellt werden (CD). Innerhalb eines CI/CD-Tools gibt es zwei zentrale Komponenten:

- **Repositories:** Hier speicherst du alle Quelldateien, die in deiner Umgebung bereitgestellt werden. Diese basieren in der Regel auf Git (Azure DevOps Repositories oder GitHub).
- **Workflows:** Diese enthalten die Logik, die festlegt, wie die Dateien automatisiert bereitgestellt werden (z.B. Azure DevOps Pipelines oder GitHub Actions).

Seit der Einführung von Microsoft Sentinel engagiert sich die Community stark, um die Bereitstellung zu automatisieren. Auf der Ignite 2021 kündigte Microsoft eine neue Funktion namens „Repositories“ an – den offiziellen Ansatz zur automatisierten Bereitstellung in Microsoft Sentinel.

## Repositories

Mit der Funktion „Repositories“ kannst du Cloud-Repositories mit deiner Microsoft Sentinel-Umgebung verbinden. Ein integrierter, von Sentinel verwalteter Workflow erstellt automatisch alle erforderlichen Sentinel-Ressourcen in deiner Umgebung, sobald ein neuer Commit durchgeführt wird.

Aktuell werden GitHub und Azure DevOps als Quellcodeverwaltungssysteme unterstützt. Die Funktion basiert auf ARM-Vorlagen, weshalb nur bestimmte Ressourcentypen automatisiert bereitgestellt werden können. Du musst den Inhalt also im ARM-Vorlagenformat hochladen, damit er von Microsoft Sentinel verarbeitet werden kann. Derzeit sind folgende Ressourcen unterstützt: Analyseregeln, Hunting-Abfragen, Playbooks, Automatisierungsregeln, Parser und Arbeitsmappen.

Bevor du ein Repository integrieren kannst, musst du die Microsoft Sentinel-App installieren. Dabei handelt es sich um ein Plugin für das Quellcodeverwaltungssystem, das die kontinuierliche Bereitstellung deiner Regeln übernimmt. Du installierst das Plugin, indem du im Portal dem Workflow „**Repository hinzufügen**“ folgst, den du im Blatt „**Repositories**“ findest.

Dort kannst du auswählen, in welchen Repositories die App installiert werden soll – wie in Abbildung 13-59 dargestellt.

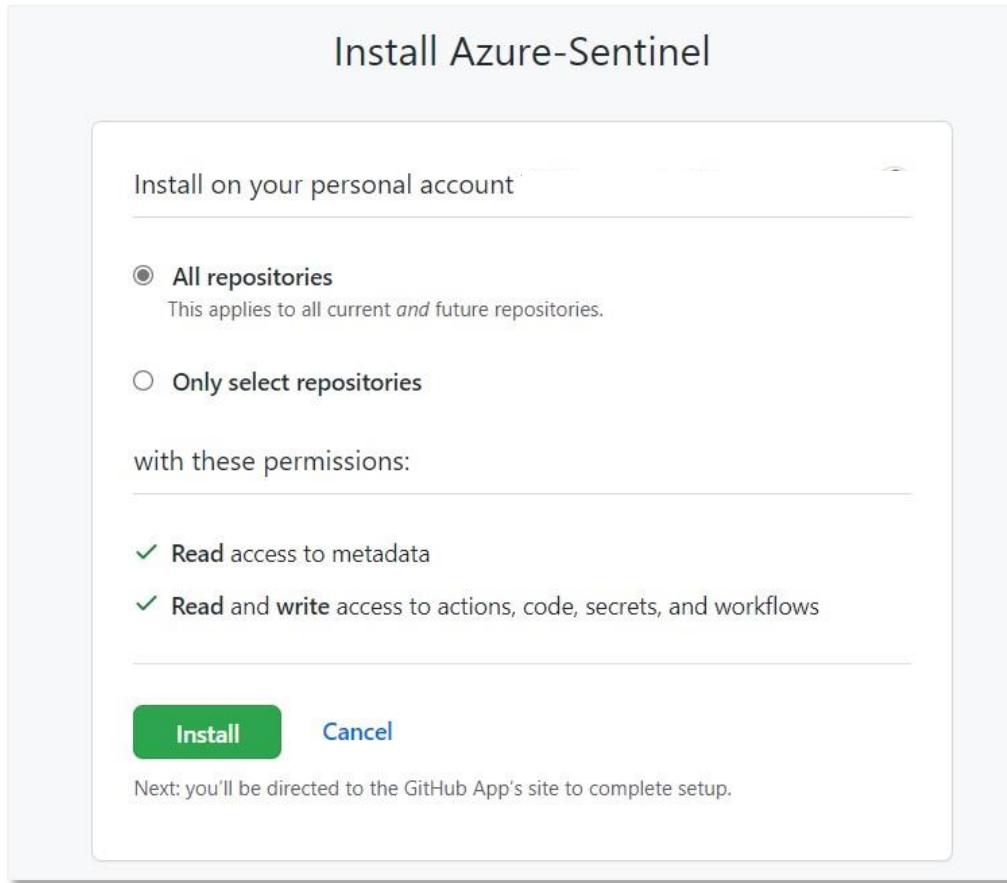


Abbildung 13-59: Installation der Microsoft Sentinel-App auf GitHub

Nach der Installation der Anwendung auf den Repositories kannst du mit der Konfiguration fortfahren:

1. Wähle die gewünschte **Quellcodeverwaltungsanwendung** aus und autorisiere sie. Das erfordert die Anmeldung mit einem Konto für den ausgewählten Dienst und autorisiert Microsoft Sentinel.
2. Wähle das **Repository** aus, das du verwenden möchtest.
3. Konfiguriere den gewünschten **Branch**. Nur Commits im ausgewählten Branch werden an Microsoft Sentinel übertragen.
4. Wähle die gewünschten **Inhaltstypen** aus, die du bereitstellen möchtest. Nur der hier ausgewählte Inhalt wird bereitgestellt, auch wenn ARM-Vorlagen für andere Inhaltstypen im Repository verfügbar sind.

Create new connection ...

Name \*  
Corporate GitHub Connection

Description

Source control \*  
GitHub  
 Authorized

Repository \* ⓘ  
-TC/Azure-Sentinel

Branch \*  
master

Content types \*  
3 selected

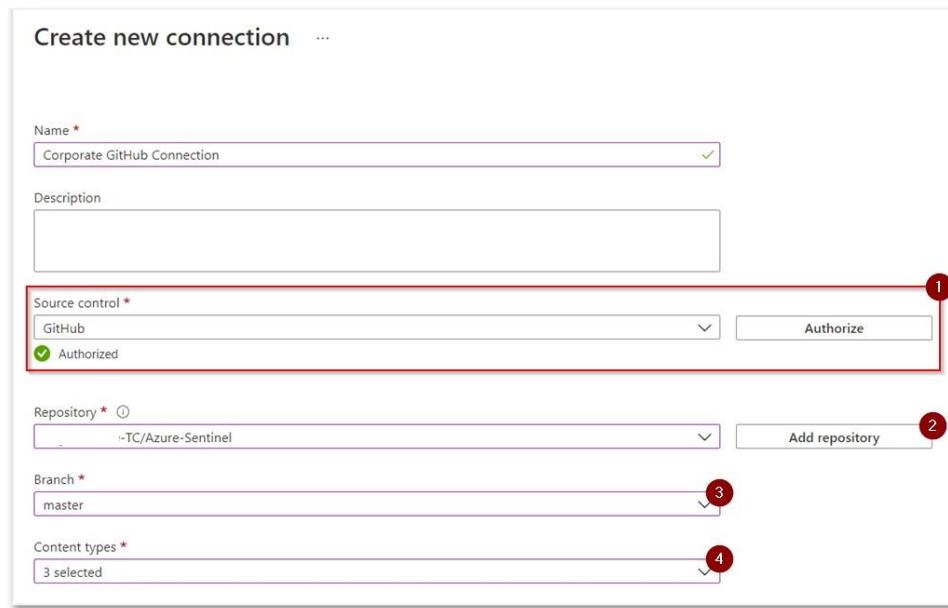


Abbildung 13-60: Erstellen einer Verbindung zu einer Quellcodeverwaltung

Nach der Konfiguration wird die Verbindung im Portal sichtbar. Unmittelbar nach dem ersten Commit wird der Inhalt automatisch in der Microsoft Sentinel-Umgebung deiner Organisation bereitgestellt. Die Unterstützung ist derzeit auf fünf (5) Repositorys pro Arbeitsbereich beschränkt.

Im Backend verwendet diese Funktion Pipelines und Actions in Azure DevOps bzw. GitHub. Das bedeutet, dass du den Erfolg deiner Bereitstellungen überwachst, indem du den Status in den jeweiligen Plattformen prüfst. Es gibt derzeit keine integrierte Möglichkeit, Integritätswarnungen zu erhalten, wenn die Bereitstellung einer bestimmten Vorlage fehlgeschlagen ist.

**Hinweis:** Das Standardverhalten dieser Funktion besteht darin, nur Inhalte bereitzustellen, die geändert wurden. Das bedeutet, dass Änderungen, die direkt im Portal vorgenommen werden, nicht überschrieben werden. Meiner Meinung nach ist das nicht das, was du möchtest, da dein CI/CD-Tool immer die korrekten Werte enthalten sollte. Dieses Verhalten kann mithilfe der [folgenden Anleitung](#) aktualisiert werden, siehe "Intelligente Bereitstellungen deaktivieren".

## Benutzerdefinierte Implementierung

Während die integrierte Funktion „Repositories“ einen großen Mehrwert bietet und es neuen Kunden ermöglicht, CI/CD einfach in ihren Workflow zu integrieren, stößt man möglicherweise an einige Einschränkungen mit den integrierten Funktionen.

Es gibt einige Szenarien, auf die ich gestoßen bin, die eine weitergehende, manuelle Einrichtung erfordern:

- Wenn du ein MSSP bist, möchtest du deine Quellcodeverwaltungs-Repositories nicht wirklich mit ihrer Umgebung verbinden, da dies bedeuten würde, dass ihre Microsoft Sentinel-Umgebung Berechtigungen für deine Repositorys hat.
- Während einige Anpassungen verfügbar sind, wie z. B. das Ändern des Bereitstellungstriggers oder -pfads, werden erweiterte Anpassungen wie das Zusammenführen von Inhalten oder andere (Nicht-ARM-)Ressourcen nicht unterstützt.
- Derzeit wird nur eine begrenzte Überprüfung dieser ARM-Vorlagen durchgeführt. Wenn du deine eigene Umgebung verwaltetst, ist es möglicherweise empfehlenswert, einige Validierungsregeln zu erstellen, die bestimmte Konventionen überprüfen, bevor eine Regel bereitgestellt wird.

Wenn deine Anforderungen über die integrierten Funktionen hinausgehen, hindert dich nichts daran, deinen eigenen CI/CD-Workflow zu erstellen und ihn mit Microsoft Sentinel zu integrieren.

Ein Blick auf die Einrichtung der integrierten Repositories ist ein guter Ausgangspunkt, aber es gibt auch einige großartige Ressourcen in der [Microsoft Tech Community](#), die dir beim Einstieg helfen.

## Multi-Tenant-Unterstützung in Microsoft Sentinel

Während Microsoft Sentinel Funktionen bietet, die für alle Arten von Kunden gelten, gibt es spezifische Features, die besonders für Kunden nützlich sind, die eine Multi-Tenant-Umgebung betreiben. Bevor wir auf diese Funktionen eingehen, möchte ich kurz etwas ansprechen, das Microsoft Sentinel aktuell nicht bietet:

Genauer gesagt: die Multi-Tenant-Unterstützung für bestimmte Datenconnectors. Es gibt einige First-Party-Datenconnectors, die mandantenspezifische Daten erfassen – darunter solche für Entra ID, Office 365, Defender und Azure-Aktivitätsdaten. All diese Connectors unterstützen nur die Verbindung mit dem Mandanten, in dem Microsoft Sentinel bereitgestellt ist. Es gibt keine Möglichkeit, Microsoft Sentinel mit mehreren Mandanten oder einem externen Mandanten zu verbinden.

Wenn du Daten aus verschiedenen Mandanten erfassen möchtest, hast du im Wesentlichen zwei Möglichkeiten:

- Du erstellst einen benutzerdefinierten (API-basierten) Connector, der eine Verbindung zu den verschiedenen Mandanten herstellt und die Daten an Log Analytics überträgt.
- Du rittest Microsoft Sentinel in jedem Mandanten ein und nutzt die Multi-Tenant-Funktionen von Microsoft Sentinel.

Die Multi-Tenant-Unterstützung in Microsoft Sentinel basiert auf **Azure Lighthouse**. Lighthouse ist eine Funktion, mit der Organisationen Abonnements aus mehreren Mandanten zentral verwalten können. Sie wurde speziell für MSSPs und Organisationen entwickelt, die mehrere Mandanten betreiben.

Bevor du eine Microsoft Sentinel-Umgebung in einem anderen Mandanten verwalten kannst, musst du die entsprechenden Abonnements über Azure Lighthouse [integrieren](#). Dafür erstellst du eine Konfigurationsdatei, die festlegt, welchen Gruppen oder Anwendungen Berechtigungen für die jeweiligen Abonnements erteilt werden. Nach der Integration kannst du dann die Microsoft Sentinel-Umgebungen in diesen Mandanten zentral verwalten.

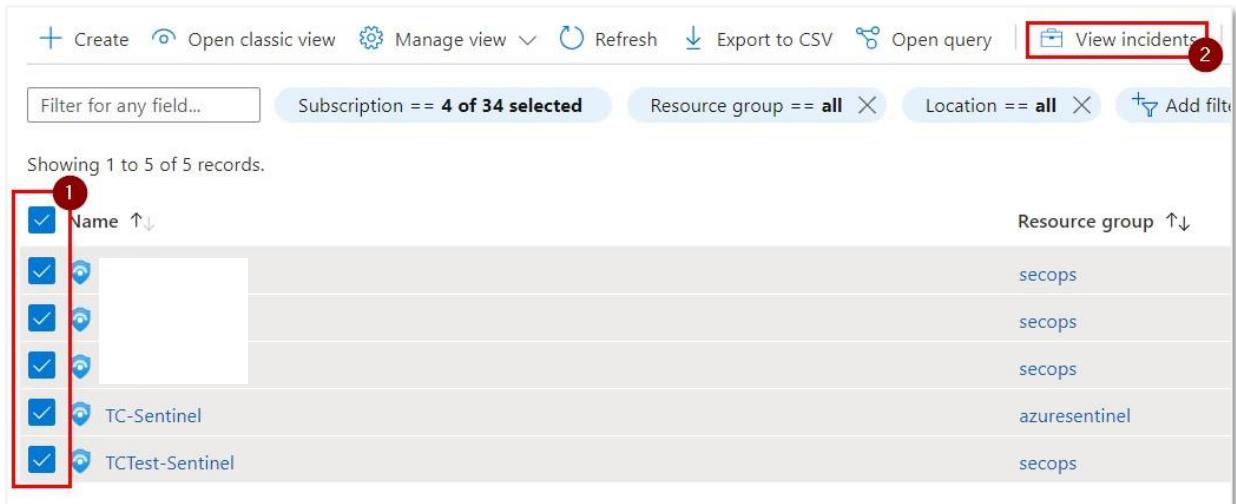
**Wählen der richtigen Berechtigungen:** Bevor du andere Mandanten in Azure Lighthouse integrierst, musst du überlegen, welche Benutzer in deinem Mandanten welche Berechtigungen erhalten. Wenn du Microsoft Sentinel verwalten möchtest, empfehle ich, deinen SOC-Analysten die Rolle "Microsoft Sentinel Responder" zuzuweisen, den Architekten die Rolle "Contributor" (da die Rolle "Microsoft Sentinel Contributor" nicht die Berechtigungen zum Bereitstellen von Playbooks und Arbeitsmappen bietet) und dem Service Principal die allgemeine Rolle "Contributor", wenn du die Bereitstellung über Azure DevOps automatisieren möchtest. Beachte, dass die Verwaltung von Automatisierungsregeln in verschiedenen Mandanten möglicherweise [zusätzliche Berechtigungen](#) erfordert.

Nachdem du die Abonnements in Azure Lighthouse integriert hast, kannst du die Microsoft Sentinel-Umgebung entsprechend den von dir konfigurierten Berechtigungen verwalten. Das Navigieren in der Microsoft Sentinel-Ressource funktioniert dabei genauso wie die Verwaltung von Microsoft Sentinel in deiner eigenen Umgebung.

Microsoft Sentinel bietet zwei Funktionen, die besonders für Organisationen nützlich sind, die mehrere Mandanten verwalten. Diese Funktionen stehen dir unabhängig davon zur Verfügung, ob du Azure Lighthouse nutzt oder mehrere Microsoft Sentinel-Arbeitsbereiche in einem einzelnen Mandanten eingerichtet hast. Die beiden Funktionen sind:

- Multi-Workspace-incidentansicht
- Arbeitsbereichsübergreifende Abfragen

Wenn du im Azure-Portal zu „Microsoft Sentinel“ navigierst, kannst du bis zu 30 Arbeitsbereiche gleichzeitig auswählen und oben rechts auf „**Incidents anzeigen**“ klicken, wie in Abbildung 13-61 zu sehen.



Name ↑↓	Resource group ↑↓
TC-Sentinel	secops
TCTest-Sentinel	secops
	azuresentinel
	secops
	secops

Abbildung 13-61: Starten der Multi-Workspace-Ansicht

Damit öffnest du die **Multi-Workspace-incidentansicht**, in der du Incidents mandantenübergreifend sehen kannst. Diese Ansicht entspricht der regulären Incidentübersicht, mit der Ausnahme einer zusätzlichen Spalte „Verzeichnis“, die angibt, zu welchem Mandanten ein bestimmter Incident gehört.

Eine Multi-Workspace-Ansicht ist derzeit nur für Incidents verfügbar. Andere Funktionen wie Analysen, Automatisierung oder Arbeitsmappen unterstützen keine mandantenübergreifende Darstellung. Wenn du die Microsoft Sentinel-Ressourcen all deiner Kunden im Blick behalten möchtest, musst du die Microsoft Sentinel-Instanzen nacheinander öffnen oder auf eine API zurückgreifen.

**Hinweis:** UEBA unterstützt Azure Lighthouse aktuell nicht vollständig. Wenn du eine Benutzerentitätsseite aufrufst, werden Informationen aus Entra ID herangezogen – zum Beispiel die Berufsbezeichnung oder der Standort des Benutzers. Da Lighthouse jedoch keine Berechtigungen für Entra ID bereitstellt, können diese Informationen nicht angezeigt. Auch wenn das möglicherweise kein Ausschlusskriterium ist, solltest du diese Einschränkung im Hinterkopf behalten.

Eine weitere Funktion, die sowohl für Single- als auch für Multi-Tenant-Umgebungen hilfreich ist, sind **arbeitsbereichsübergreifende Abfragen**. Damit kannst du andere Arbeitsbereiche von einem zentralen aus abfragen. Mit der Funktion workspace() in KQL lassen sich Daten aus mehreren Arbeitsbereichen kombinieren.

```
workspace('Workspace ID 1').SecurityIncident | union workspace('Workspace ID 2').SecurityIncident
```

Diese Funktion kannst du in manuellen Abfragen, Analyseregeln, Hunting-Abfragen und Arbeitsmappen verwenden. Wenn du viele Arbeitsbereiche verwaltet, empfehle ich dir,

eine [Log Analytics Funktion](#) einzurichten, die den Zugriff auf die verschiedenen Arbeitsbereiche vereinfacht.

**Hinweis:** Wenn du als MSSP Microsoft Sentinel verwenden möchtest, findest du ein umfassendes [Whitepaper](#) von Microsoft zu diesem Thema, das alle relevanten Funktionen beschreibt.

## Workspace Manager

Im April 2023 hat Microsoft eine neue Funktion namens „Workspace Manager“ veröffentlicht, die speziell für MSSPs entwickelt wurde. Sie erleichtert es dir, Inhalte wie Analyseregeln, Arbeitsmappen und Playbooks mandantenübergreifend bereitzustellen. Um den Workspace Manager zu verwenden, gehst du folgendermaßen vor:

- Definiere einen **zentralen Arbeitsbereich**, der zur Verteilung von Inhalten verwendet wird. Normalerweise ist dies der Arbeitsbereich in der eigenen Umgebung des MSSPs.
  - Um einen zentralen Arbeitsbereich zu definieren, öffne Microsoft Sentinel, wähle Einstellungen und erweitere die Registerkarte "Workspace-Manager-Konfiguration". Hier kannst du den aktuellen Arbeitsbereich als "zentralen Arbeitsbereich", auch Parent genannt, konfigurieren.
- Als Nächstes kannst du Arbeitsbereiche als Mitglied zu diesem zentralen Arbeitsbereich hinzufügen. Navigiere dazu zu "Workspace Manager", wähle "Hinzufügen" und füge die Arbeitsbereiche hinzu, die du verwalten möchtest.
- Nach dem Hinzufügen der Arbeitsbereiche kannst du Inhalte übertragen, indem du eine Gruppe änderst und Inhalte hinzufügst. Dieser Prozess wird in der [Microsoft Sentinel-Dokumentation](#) ausführlich behandelt.

Der Workspace Manager eignet sich hervorragend, wenn du sicherstellen willst, dass eine bestimmte Sammlung von Inhalten in mehreren Sentinel-Arbeitsbereichen bereitgestellt wird. Die Nutzung ist einfach und erfordert keine komplexe Einrichtung oder Konfiguration.

Als diese Funktion veröffentlicht wurde, habe ich mich gefragt, wie sie sich von einem Setup mit CI/CD (z. B. mit Azure DevOps oder GitHub), entweder als benutzerdefinierte Implementierung oder über die Funktion „Repositories“, unterscheidet. Beide Ansätze haben ihre eigenen Vorteile:

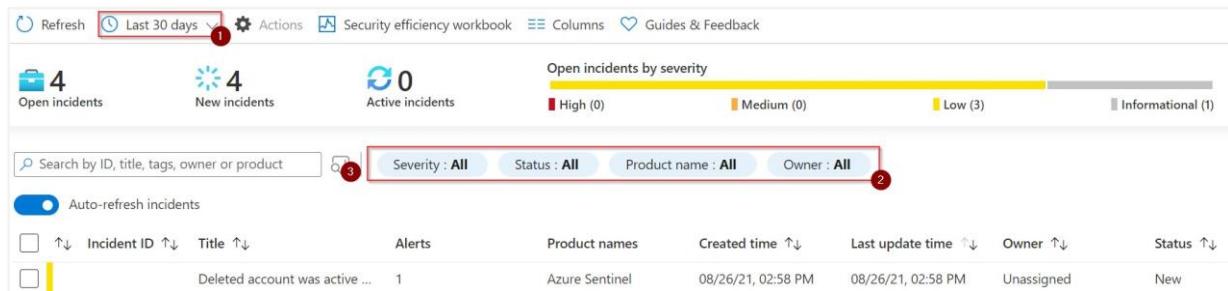
- **Workspace Manager**
  - Einfach einzurichten.
  - Ermöglicht die Konfiguration einer Reihe von Inhalten für jede Gruppe.
- **CI/CD**
  - Ermöglicht das "Vier-Augen-Prinzip", bei dem eine Änderung von jemand anderem genehmigt werden muss.

- Durch die Verwendung von Validierungsskripten kannst du sicherstellen, dass die Qualität der Inhalte überprüft wird, bevor sie bereitgestellt werden.
- Es unterstützt alle Microsoft Sentinel-Inhalte (während der Workspace Manager eine Reihe von [Einschränkungen](#) hat).

Welche Lösung für dich die bessere ist, hängt von der Komplexität deiner Umgebung und deinem Reifegrad als MSSP ab. Wenn du bereits ein fortgeschrittener MSSP bist, empfiehlt sich der Einsatz von CI/CD, da es dir deutlich mehr Kontrolle bietet. Der Workspace Manager ist ideal, wenn du manuelle Tätigkeiten reduzieren möchtest, ohne dich mit der komplexen Konfiguration und Verwaltung eines CI/CD-Prozesses auseinandersetzen zu müssen.

## Reagieren auf Incidents mit Microsoft Sentinel

Alle Incidents, die derzeit in deiner Umgebung aktiv sind, findest du im Blatt Incidents im Microsoft Sentinel-Portal. Ein Beispiel siehst du in Abbildung 13-62.



The screenshot shows the Microsoft Sentinel Incidents blade. At the top, there are three summary cards: 'Open incidents' (4), 'New incidents' (4), and 'Active incidents' (0). Below these are several filters: 'Last 30 days' (1), 'Actions', 'Security efficiency workbook', 'Columns', 'Guides & Feedback'. A 'Severity' dropdown is highlighted with a red circle containing the number 3. Other filter options include 'Status : All', 'Product name : All', and 'Owner : All'. A 'Severity' legend shows: High (0) in red, Medium (0) in orange, Low (3) in yellow, and Informational (1) in grey. Below the filters is a table header with columns: Incident ID ↑↓, Title ↑↓, Alerts, Product names, Created time ↑↓, Last update time ↑↓, Owner ↑↓, and Status ↑↓. The table body contains one row of data: 'Deleted account was active ...' with value '1', 'Azure Sentinel', '08/26/21, 02:58 PM', '08/26/21, 02:58 PM', 'Unassigned', and 'New'.

Abbildung 13-62: Übersicht über Microsoft Sentinel-Incidents

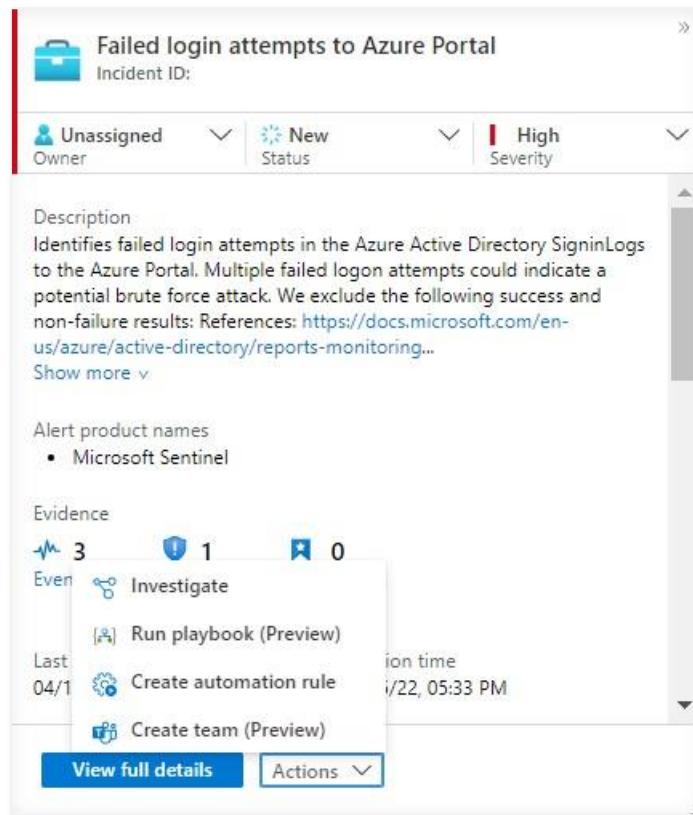
Hier erhältst du eine Übersicht über den Status deiner Umgebung. Dazu gehören die Anzahl der offenen Incidents und deren Kategorisierung nach Schweregrad. Es stehen dir einige Filter zur Verfügung:

1. Standardmäßig werden nur die Incidents der letzten 24 Stunden angezeigt. Du kannst diesen Zeitraum über den Zeitbereich oben auf der Seite anpassen.
2. Es gibt grundlegende Filter, um die aktuellen Incidents einzuschränken. Standardmäßig siehst du nur neue oder aktive Incidents. Geschlossene Incidents sind im Standardfilter nicht enthalten.
3. Wenn du nach Incidents mit anderen Eigenschaften als den in den Basisfiltern verfügbaren suchen möchtest, kannst du die "erweiterte Incidentsuche" verwenden. Wenn du auf das Symbol klickst, kannst du die Felder auswählen, in denen du suchen willst. Anschließend nutzt du das Suchfeld, das nach den Schlüsselwörtern in den

ausgewählten Feldern sucht. Das ist eine einfache Möglichkeit, die Filteroptionen zu erweitern und nach fast allem zu suchen.

Wenn du einen Incident auswählst, werden dir weitere Details angezeigt. Dadurch öffnet sich die Detailseite des jeweiligen Incidents. Ein Beispiel siehst du in Abbildung 13-63. Zu den Informationen gehören:

- Besitzer und Status dieses Incidents
- Die Anzahl der Ereignisse, Warnungen und Lesezeichen, die sich auf diesen Incident beziehen
- Betroffene Entitäten



The screenshot shows the Microsoft Sentinel interface for an incident titled "Failed login attempts to Azure Portal". The incident ID is listed as "Incident ID: [redacted]". The top navigation bar includes "Owner" (Unassigned), "Status" (New), and "Severity" (High). The main content area displays the following details:

- Description:** Identifies failed login attempts in the Azure Active Directory SigninLogs to the Azure Portal. Multiple failed logon attempts could indicate a potential brute force attack. We exclude the following success and non-failure results: References: <https://docs.microsoft.com/en-us/azure/active-directory/reports-monitoring...>
- Alert product names:** Microsoft Sentinel
- Evidence:** Shows 3 Events, 1 Warning, and 0 Lesezeichen (Bookmarks). Actions include "Investigate", "Run playbook (Preview)", "Create automation rule", and "Create team (Preview)".
- Last updated:** 04/12/2023, 05:33 PM

Abbildung 13-63: Details eines Microsoft Sentinel-Incidents

Von dieser Ansicht aus kannst du verschiedene Aktionen durchführen, die im Folgenden beschrieben werden.

## Aktualisieren von Incidenteigenschaften

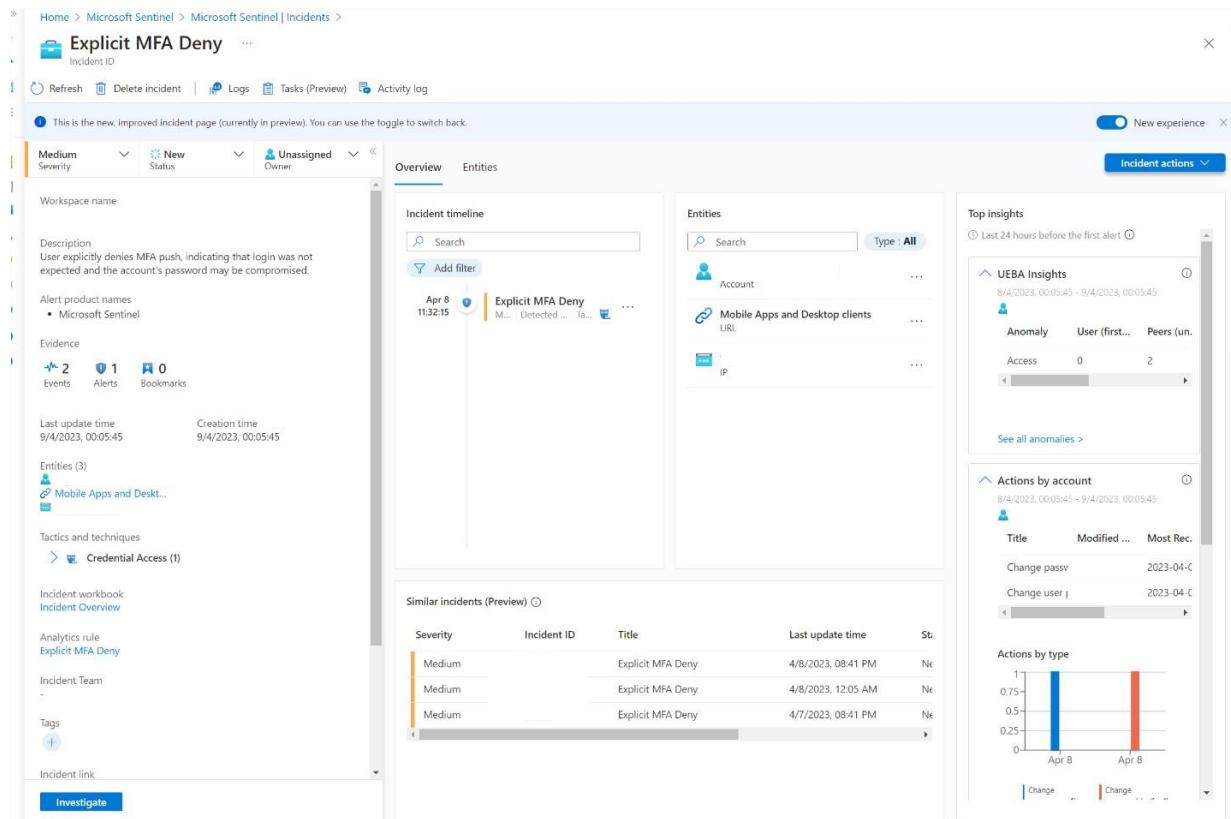
Jeder Incident besitzt einige spezifische Eigenschaften, die du aktualisieren kannst. Dazu gehören:

- **Der Besitzer:** Der Besitzer (oder auch Zugewiesener) ist für die Untersuchung des Incidents verantwortlich. Dieses Feld zeigt an, wer akutell daran arbeitet. Der Besitzer kann ein Benutzer oder eine Gruppe sein (sowohl Sicherheits- als auch Microsoft 365-Gruppen werden unterstützt).
- **Status:** Es gibt drei Statuswerte: Neu, Aktiv und Geschlossen. Incidents werden mit dem Status "Neu" erstellt. Sobald ein Analyst mit der Bearbeitung beginnt, sollte der Status auf "Aktiv" gesetzt werden. Nach Abschluss der Untersuchung wird der Status auf "Geschlossen" gesetzt – dabei ist ein Abschlussgrund bzw. eine Klassifizierung verpflichtend. Diese Klassifizierung gibt an, ob der Incident als bösartig eingestuft wurde oder beispielsweise durch eine falsch konfigurierte Analysergeregel entstanden ist.  
Folgende Optionen stehen dir zur Verfügung:
  - Richtig Positiv
  - Gutartig Positiv
  - Falsch Positiv -- falsche Warnungslogik
  - Falsch Positiv -- falsche Daten
  - Unbestimmt
- Um diese Klassifizierungen besser zu verstehen, nehmen wir die Regel "Anmeldung aus einem seltenen Land" als Beispiel. Diese Regel prüft, ob sich ein Benutzer aus einem Land anmeldet, aus dem er sich zuvor noch nicht angemeldet hat. Ein **richtig positiver** Incident liegt vor, wenn tatsächlich ein bösartiger Akteur erfolgreich Zugriff erlangt hat. Ein gutartig positiv Incident beschreibt eine verdächtige, aber erwartet Aktivität – etwa, wenn sich der Benutzer im Urlaub in einem ungewöhnlichen Land aufhält. Ein falsch positiv Incident bedeutet, dass die Aktivität nicht verdächtig ist. Hier gibt es zwei Ursachen: Bei "falsche Warnungslogik" ist die Regel falsch konfiguriert. (zum Beispiel werden fehlgeschlagene statt nur erfolgreiche Anmeldungen ausgewertet). Bei "falsche Daten" stimmen die Rohdaten in Microsoft Sentinel nicht (etwa eine fehlerhafte Geolokalisierung). Unbestimmt wird verwendet, wenn keine der anderen Klassifizierungen zutrifft – diese Option solltest du möglichst vermieden, da sie keine hilfreichen Informationen liefert.
- **Klassifizierungskommentar:** Du kannst einen optionalen Kommentar hinzufügen, um weiteren Kontext zum Incident zu dokumentieren. Es wird empfohlen, diesen auszufüllen, da er bei zukünftigen ähnlichen Incidents hilfreich sein kann.
- **Schweregrad:** Der Schwerpunkt wird in der Analysergeregel definiert, die den Incident erzeugt hat. Du kannst den Schwerpunkt nachträglich anpassen, falls die Untersuchung dies nahelegt. Zum Beispiel könnte ein Incident als besonders kritisch eingestuft werden, wenn das betroffene Konto dem CEI gehört und es sich um einen richtig positiven Vorfall handelt.

## Ansicht zur Incidentuntersuchung

Wenn du im Incidentbereich auf die Schaltfläche „**Vollständige Details anzeigen**“ klickst, wird die Seite zur Incidentuntersuchung geöffnet. Diese wurde Anfang 2023 vollständig überarbeitet, wie in Abbildung 13-64 zu sehen ist. Ziel dieser Überarbeitung war es, „Blade-Spam“ zu

vermeiden – also das Erlebnis, bei dem jedes Mal, wenn du auf eine Schaltfläche klickst, ein neues Blade geöffnet wird und du schnell den Überblick verlierst. Die neue Ansicht zur Incidentuntersuchung soll dir eine aussagekräftige Übersicht über den Incident bieten und es ermöglichen, bei Bedarf in detailliertere Informationen zu wechseln.



The screenshot shows the Microsoft Sentinel Incident Details page for an 'Explicit MFA Deny' incident. The page is divided into several sections:

- Left sidebar:** Includes navigation (Home > Microsoft Sentinel > Microsoft Sentinel | Incidents), a title 'Explicit MFA Deny' (Incident ID), and various filters (Medium Severity, New Status, Unassigned Owner). It also displays workspace name, description (User explicitly denies MFA push), alert product names (Microsoft Sentinel), evidence count (2 Events, 1 Alert, 0 Bookmarks), last update time (9/4/2023, 00:05:45), creation time (9/4/2023, 00:05:45), and entities (Mobile Apps and Desktop clients).
- Top right:** A toggle for 'New experience' (on) and 'Incident actions' dropdown.
- Middle section:**
  - Overview:** Shows the 'Incident timeline' with a single entry: 'Apr 8 11:32:15 Explicit MFA Deny'. Below it is a 'Entities' section listing Account, Mobile Apps and Desktop clients, and IP.
  - Similar incidents (Preview):** A table showing three similar incidents with Medium severity, Incident ID 'Explicit MFA Deny', Title 'Explicit MFA Deny', Last update time (4/8/2023, 08:41 PM, 12:05 AM, 08:41 PM), and Status (N/A, N/A, N/A).
- Right side:**
  - Top insights:** 'Last 24 hours before the first alert' showing UEBA Insights (Anomaly User (first... Peers (un.)), Access 0 2).
  - Actions by account:** A table showing actions like 'Change passw...' and 'Change user j...' with their respective dates (2023-04-04).
  - Actions by type:** A bar chart showing changes for April 8, with blue bars for 'Change' and red bars for 'Change'.

Abbildung 13-64: Details zur Incidentuntersuchung

Die wichtigsten Informationen werden wie folgt dargestellt:

- Der linke Bereich enthält eine Übersicht mit den zentralen **Informationen** zum Incident (Beschreibung, Beweise, Entitäten...).
  - Wenn du auf einen der **Beweise** klickst, öffnet sich ein neues Fenster mit der Log Analytics-Abfrage, während du auf der Incidentseite bleibst. So kannst du Daten analysieren, ohne den Kontext des Incidents zu verlieren.
- Der mittlere Bereich zeigt:
  - Eine **Incident-Timeline** mit allen Ereignissen, die zu dem Incident gehören. Aktuell ist diese auf Warnungen und Lesezeichen beschränkt. Über die drei Punkte neben einer Warnung kannst du ein warnungsbasiertes Playbook ausführen.
  - Entitäten:** Hier werden alle am Incident beteiligten Entitäten aufgelistet. Ein Incident kann mehrere Einträge desselben Typs enthalten. Eine fokussierte Ansicht findest du auch in der Registerkarte "Entitäten" oben auf der Seite.

Wenn du auf eine Entität klickst, gelangst du zu ihrer Detailseite. Weitere Informationen dazu folgen unten.

- Der untere Bereich zeigt **ähnliche Incidents**. Es wird eine Liste jener Incidents angezeigt, die dem aktuell angezeigten ähneln. Die Ähnlichkeit bezieht sich auf Regeln oder betroffene Entitäten und ist auf Incidents der letzten 14 Tage beschränkt. Eine Konfigurationsmöglichkeit gibt es derzeit nicht. Eine Option zur Anpassung des Zeitraums oder der Ähnlichkeitskriterien wäre jedoch sehr wünschenswert.
- Der rechte Bereich liefert nützliche Erkenntnisse zu den im Incident enthaltenen Entitäten. Hier findest du unter anderem Informationen zu den beteiligten Benutzern oder IP-Adressen.
- **Kommentare** sind etwas versteckt. Du findest sie, indem du oben auf der Seite "Aktivitätsprotokoll" klickst. Auch wenn sie nicht verpflichtend sind, empfehle ich dir, deine Untersuchung dort zu dokumentieren. So können anderen den aktuellen Stand der Analyse nachvollziehen.
- Oben rechts befindet sich eine Schaltfläche "**Incident-Aktionen**". Darüber kannst du folgende Aktionen ausführen:
  - Ein incidentbasiertes Playbook ausführen.
  - Einen Teams-Kriegsraum erstellen.
  - Eine Automatisierungsregel anlegen.

## Aufgaben

Im November 2022 wurde die Funktion „Incident-Aufgaben“ eingeführt. Sie ermöglicht es einem SOC-Ingenieur, einen Incident anhand vordefinierter Richtlinien zu untersuchen. Um auf die Aufgaben zuzugreifen, klicke oben auf der Incidentseite auf die Schaltfläche „Aufgaben“. Dort findest du eine Übersicht über alle Untersuchungsschritte. Während der Untersuchung kannst du Aufgaben abhaken, um deinen Fortschritt zu dokumentieren.

Incident tasks (Preview) X

↻ Refresh + Add task

1/4 completed

Search Status : All

<input checked="" type="checkbox"/> Verify user activity <span style="font-size: small;">Created by:</span>	...
<input type="radio"/> Check permissions of app registration <span style="font-size: small;">Created by:</span>	...
<input type="radio"/> If TP: Remove secret <span style="font-size: small;">Created by:</span>	...
<input type="radio"/> Track SP Sign-ins <span style="font-size: small;">Created by:</span>	...

Abbildung 13-65: Ansicht der Incident-Aufgaben

Aufgaben können auf drei verschiedene Arten hinzugefügt werden:

- Manuell für jeden Incident
- Durch Automatisierungsregeln
- Verwendung der integrierten Aktion in einem Playbook

Wenn du gerade erst beginnst, empfehle ich dir die Nutzung von Automatisierungsregeln, da sie den einfachsten Einstieg bieten. Für SOC-Architekten oder erfahrene Analysten sollte es das Ziel sein, strukturierte Aufgaben für jüngere Kollegen zu erstellen – so stellst du sicher, dass keine Untersuchungsschritte übersehen werden.

**Wann solltest du Aufgaben verwenden?** Diese Funktion eignet sich besonders, wenn du Microsoft Sentinel als zentrales ITSM-/Incident-Handling-Tool nutzt. Sie ist ideal, um sicherzustellen, dass Untersuchungen einheitlich durchgeführt werden. Wenn du hingegen ein anderes (Drittanbieter-)Ticketing-Tool im Einsatz hast, ist diese Funktion eventuell nicht nötig, da du dort bereits ähnliche Möglichkeiten hast. Aktuell befinden sich Aufgaben noch in der Vorschau und sind noch nicht vollständig ausgereift. Beispielsweise fehlt die Möglichkeit, Pflichtaufgaben festzulegen – also das sein Incident nicht geschlossen werden kann, bevor bestimmte Aufgaben erledigt sind. Auch kleinere Features wie das Hinzufügen von Kommentaren zu einzelnen Aufgaben wäre wünschenswert.

## Entitätsseite

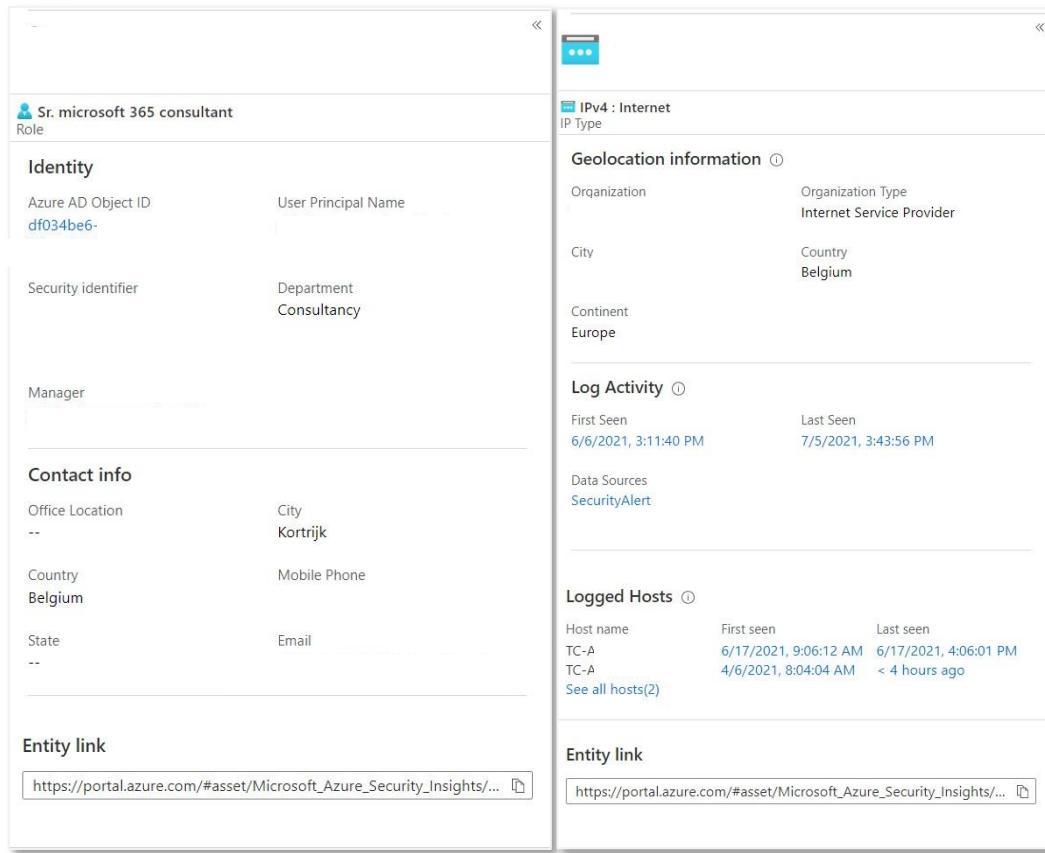
Durch das Auswählen einer Entität – entweder über die Detailseite des Incidents oder die Ansicht „Vollständige Details“ – wirst du zur Detailseite der Entität weitergeleitet. Diese Seite ist sehr hilfreich, da sie Informationen enthält, die du bei der Untersuchung eines Incidents verwenden kannst.

Die Entitätsseite besteht aus vier Hauptelementen:

- Entitätsinformationen
- Übersicht über Ereignisse und Warnungen
- Zeitleiste
- Erkenntnisse

Der Bereich „**Entitätsinformationen**“ befindet sich auf der linken Seite des Bildschirms und enthält allgemeine Informationen über die Entität. Ein Beispiel siehst du in Abbildung 13-66. Das linke Bild zeigt eine Benutzerentitätsseite, das rechte eine IP-Entitätsseite. Die Benutzerentitätsseite liefert aufschlussreiche Informationen wie Berufsbezeichnung, Abteilung und Standort des Benutzers. Diese Eigenschaften können bei der Untersuchung sehr nützlich sein. Beispielsweise ist ein Benutzer aus der IT-Abteilung eher geneigt, Tools wie einen IP-Scanner herunterzuladen, als jemand aus der Finanzabteilung. Wenn du außerdem eine Warnung untersuchst, bei der sich ein Benutzer von einem potenziell bösartigen Standort angemeldet hat, kannst du anhand des Standorts überprüfen, ob dieser geografisch in der Nähe des üblichen Standorts des Benutzers liegt.

Die IP-Entitätsseite liefert dir zusätzliche Geolokalisierungsdaten zur IP-Adresse sowie Informationen darüber, wie diese IP-Adresse in deiner Organisation verwendet wurde.



The image shows two side-by-side entity detail pages. The left page is for a user named 'Sr. microsoft 365 consultant' with the role 'Manager'. It displays sections for Identity (Azure AD Object ID: df034be6-), Contact info (Office Location: Kortrijk, Country: Belgium, State: --), and Entity link ([https://portal.azure.com/#asset/Microsoft\\_Azure\\_Security\\_Insights/...](https://portal.azure.com/#asset/Microsoft_Azure_Security_Insights/)). The right page is for an IP address 'IPv4 : Internet' (IP Type: Internet Service Provider). It shows Geolocation information (Organization: Internet Service Provider, City: Belgium, Continent: Europe), Log Activity (First Seen: 6/6/2021, 3:11:40 PM, Last Seen: 7/5/2021, 3:43:56 PM, Data Sources: SecurityAlert), and Logged Hosts (Host name: TC-A, First seen: 6/17/2021, 9:06:12 AM, Last seen: 6/17/2021, 4:06:01 PM; Host name: TC-A, First seen: 4/6/2021, 8:04:04 AM, Last seen: < 4 hours ago). Both pages have a header with back and forward navigation arrows.

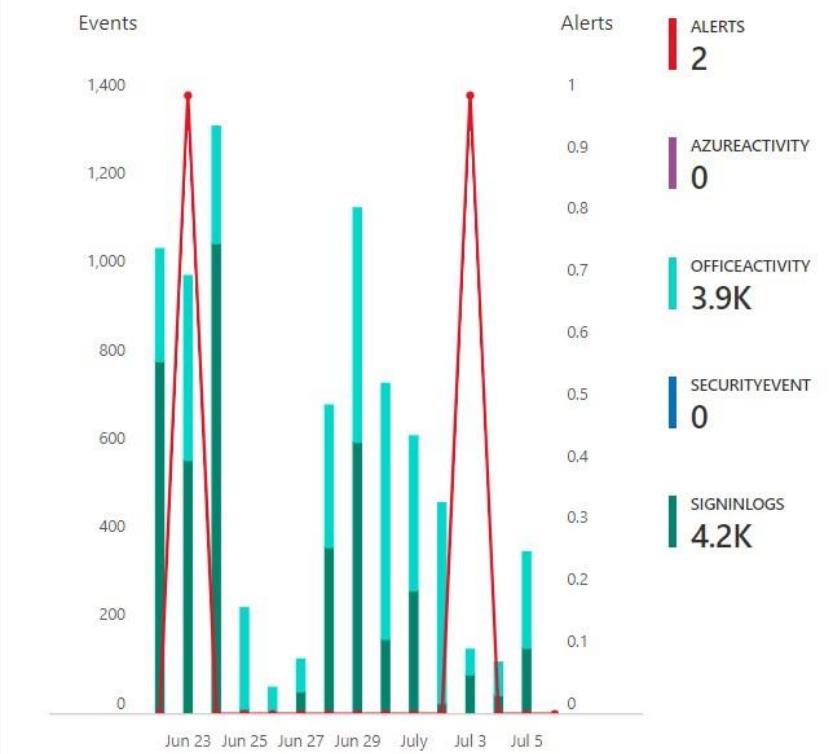
Abbildung 13-66: Beispiel für eine Benutzerentitätsseite.

**Hinweis:** Entitätsseiten sind von der Aktivierung von UEBA abhängig. Wenn du nicht die erwarteten Daten siehst, stelle sicher, dass UEBA aktiviert ist.

**Vorsicht bei der Geolokalisierung:** Der dargestellte Standort einer IP-Adresse entspricht nicht zwangsläufig dem tatsächlichen physischen Standort des Benutzers. Einige Online-Dienste (wie VPN-Anbieter) verwenden IP-Adressen, die in einem Land registriert sind, aber in einem anderen aktiv sind. Nutze diese Informationen also mit Bedacht und nimm nichts als selbstverständlich an.

Rechts neben dem Bereich mit den Entitätsinformationen findest du den Graph „**Ereignisse und Warnungen**“. Dieser bietet eine visuelle Darstellung, wie oft der Benutzer in bestimmten Protokollen erscheint. Wie in Abbildung 13-67 zu sehen ist, ist diese Darstellung jedoch nicht besonders hilfreich und skaliert schlecht bei vielen Protokollen.

Events and alerts over time


*Abbildung 13-67: Beispiel für den Graphen "Ereignisse und Warnungen"*

Die **Warnungs- und Aktivitätszeitleiste** zählt zu den nützlichsten Funktionen auf der Entitätsseite. Sie zeigt alle Warnungen, Aktivitäten und Lesezeichen, die sich auf diese Entität beziehen, in einer chronologischen Reihenfolge. Diese Darstellung hat zwei große Vorteile:

- Du erhältst eine Übersicht über alle Warnungen, die sich auf die Entität beziehen. Dadurch kannst du erkennen, ob etwa viele falsch-positive Ergebnisse auftreten oder ob möglicherweise ein gezielter Angriff auf die Entität erfolgt – z.B. bei einem plötzlichen Anstieg von Warnungen.
- Du kannst Aktivitäten hervorheben, die für deine Untersuchung relevant sind.

**Den Aufwand investieren:** Dieser Bereich entfaltet sein volles Potenzial nur, wenn du dir die Zeit nimmst, eigene Aktivitätsabfragen zu konfigurieren. Jede Organisation sollte ihre Anforderungen analysieren und geeigente Abfragen einrichten. Beispiele für hilfreiche Abfragen sind etwa Passwortzurücksetzungen, Änderungen an MFA-Methoden oder das Löschen von Dateien.

Der letzte Teil der Entitätsseite ist der Bereich „Erkenntnisse“. Hier findest du zusätzliche, teilweise sehr nützliche Informationen zur Entität, wie etwa:

- Peers eines Benutzerkontos
- Verknüpfte TI-Indikatoren (am sinnvollsten für IP-Adressen)
- Zugriff auf Azure-Abonnements
- Watchlists-Integrationen

Auch wenn die Registerkarte „Erkenntnisse“ hilfreich sein kann, ist sie häufig überladen. Der größte Nachteil ist, dass du sie nicht nach deinen eigenen Anforderungen anpassen kannst – du kannst also weder auswählen, welche Erkenntnisse angezeigt werden, noch ihre Reihenfolge beeinflussen. Ich empfehle dir, für dich nützliche Erkenntnisse gezielt zu nutzen, aber dich nicht von irrelevanten Informationen ablenken zu lassen.

## Untersuchungsgraph

Wenn du auf der Incidentseite den Befehl „**Untersuchen**“ auswählst, wirst du zum Untersuchungsgraphen weitergeleitet. Dieser zeigt dir visuell alle mit dem Incident zusammenhängenden Warnungen und Entitäten. Bei einfachen Incidents ist die Darstellung klar und verständlich (siehe Abbildung 13-67). Bei komplexeren Fällen mit vielen Entitäten kann der Graph jedoch schnell unübersichtlich und überladen werden.



Abbildung 13-68: Beispiel für den Untersuchungsgraphen

Wenn du mit der Maus über eine Entität fährst, kannst du zusätzliche Ereignisse abrufen, zum Beispiel verwandte Hosts oder Warnungen. Dadurch kannst du spezifische Abfragen im Zusammenhang mit dem Incident ausführen. Wenn du eine bestimmte Abfrage auswählst, werden diese Daten dynamisch zum Untersuchungsgraphen hinzugefügt. Auf diese Weise kannst du den Graphen entsprechend deiner Untersuchungsschritte erweitern.

Auf der rechten Seite des Bildschirms ist eine Navigationsleiste mit verschiedenen Optionen verfügbar:

- Zeitleiste
- Info
- Entitäten
- Erkenntnisse

- Hilfe

Die **Zeitleiste** ist vergleichbar mit der Zeitleiste in den vollständigen Incidentdetails, mit dem Unterschied, dass hier eine Zeitleiste für jede Warnung erstellt wird, die derzeit im Untersuchungsgraphen verfügbar ist. Wenn du Warnungen im Zusammenhang mit einer bestimmten Entität geöffnet hast, werden diese hier ebenfalls angezeigt. Wenn du mit der Maus über eine Warnung fährst, kannst du die zur spezifischen Warnung gehörenden Entitäten anzeigen.

Durch Auswahl der Option **Info** erhältst du weitere Informationen über die aktuell ausgewählte Warnung oder Entität.

Die Entitätenliste zeigt eine Übersicht über alle Entitäten, Warnungen und Lesezeichen, die derzeit im Untersuchungsgraphen sichtbar sind. Ein wirklich cooles Feature hier ist die [URL-Detonation](#). Wenn du eine URL als Entität konfiguriert hast, öffnet Microsoft Sentinel diese URL in einer Sandbox, bewertet sie (bösertig oder sauber) und stellt einen Screenshot der Website bereit. Dadurch kannst du URLs validieren, ohne eine Sandbox-Umgebung öffnen zu müssen.

Durch Auswahl einer Entität und Klicken auf die Schaltfläche Erkenntnisse erhältst du eine Übersicht über die Entitätserkenntnisse, wie du sie bereits auf der Entitätsseite gesehen hast.

Die letzte Schaltfläche, Hilfe, bietet einige Tipps für den Einstieg in den Untersuchungsgraphen.

## Automatisierungsregeln

Mithilfe der Schaltfläche „Automatisierungsregel erstellen“ kannst du schnell eine neue Automatisierungsregel erstellen, die auf den untersuchten Incident zugeschnitten ist. Alle Bedingungen, zum Beispiel der Name der Analyseregel und die Entitäten, werden für dich vorgefüllt. Du kannst einfach eine „Unterdrückungsregel“ erstellen, die alle Incidents schließt, die diesem spezifischen Incident ähneln. Eine Unterdrückungsregel ist eine spezifische Automatisierungsregel, die Incidents schließt, wenn bestimmte Kriterien erfüllt sind. Das ist eine großartige Möglichkeit, bestimmte falsch positive Incidents automatisch zu schließen. Dies wird im nächsten Abschnitt ausführlicher erläutert.

**Hinweis:** Diese Art von Automatisierungsregeln wird automatisch mit einem Standardablauftag von einem Tag erstellt. Beachte dies, wenn du eine Regel erstellen möchtest, die unbegrenzt läuft.

## Incident-Zusammenarbeit mit Microsoft Teams

Der Kommentarbereich für Incidents ist eine großartige Möglichkeit zur Zusammenarbeit, hat aber einige Einschränkungen. Während er perfekt ist, um einige schnelle Informationen

zwischen Analysten auszutauschen, die am selben Incident arbeiten, ist er nicht das richtige Tool, wenn man wirklich zusammenarbeiten muss. Microsoft Sentinel bietet die Möglichkeit, ein Team für einen bestimmten Incident zu erstellen. Dies kannst du durch Klicken auf „**Aktionen > Team erstellen**“ tun. Dadurch wird der Assistent „Incident-Team“ geöffnet, wie in Abbildung 13-69 dargestellt. In diesem Assistenten sind die Optionen auf Folgendes beschränkt:

- Bereitstellen eines Teamnamen
- Einrichten einer Beschreibung
- Hinzufügen vorhandener Gruppen zum Team

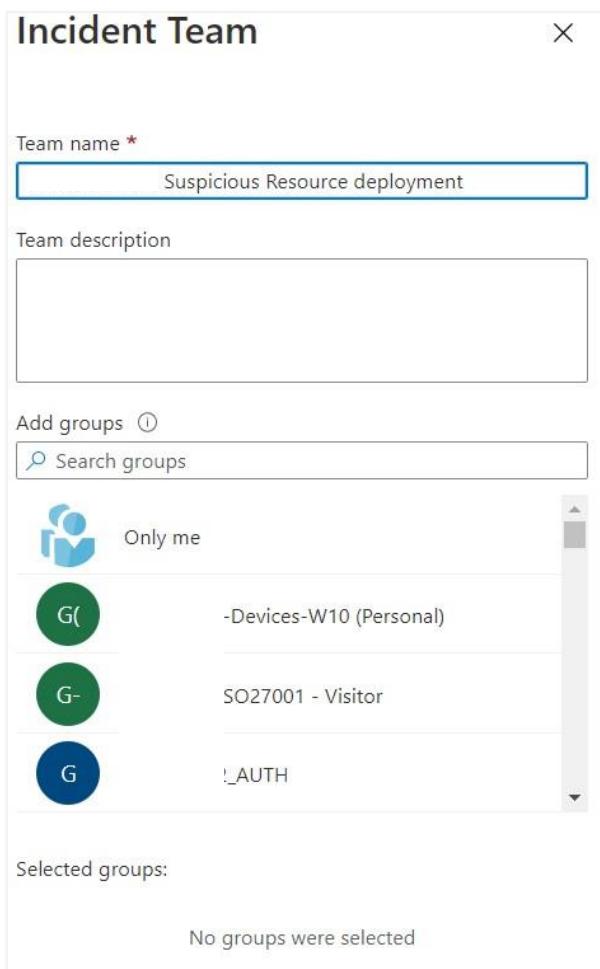


Abbildung 13-69: Erstellen eines Teams für eine Untersuchung

Obwohl dies eine großartige Funktion ist, um schnell ein Team aufzusetzen, gibt es einige Dinge zu beachten:

- Es ist wahrscheinlich nicht ratsam, für jeden Incident ein neues Team zu erstellen, da du auf lange Sicht eine Menge Teams haben wirst, wenn du dies tust.

- Du kannst Mitglieder nur auf der Grundlage vorhandener Gruppen hinzufügen, du kannst keine Gruppe erstellen oder den Zugriff auf das Team gewähren, indem du eine Reihe von Benutzern auswählst. Wenn du bestimmte Benutzer einladen möchtest, solltest du sie in der Teams-Anwendung einladen.
- Das Team wird durch Identitätsklicken auf die Aktion "Team erstellen" erstellt. Das bedeutet, dass das Konto die Möglichkeit haben muss, Microsoft 365-Gruppen zu erstellen, was in vielen Organisationen für reguläre Benutzerkonten deaktiviert ist.

**Teams vs. Kanal:** Obwohl dies eine großartige Funktion ist, die die Integrationsmöglichkeiten zwischen Microsoft Sentinel und anderen Microsoft-Produkten zeigt, solltest du mit dieser Funktion vorsichtig sein. Ich persönlich bevorzuge es, ein generisches Team für das SOC zu erstellen und separate Kanäle für große Incidents hinzuzufügen.