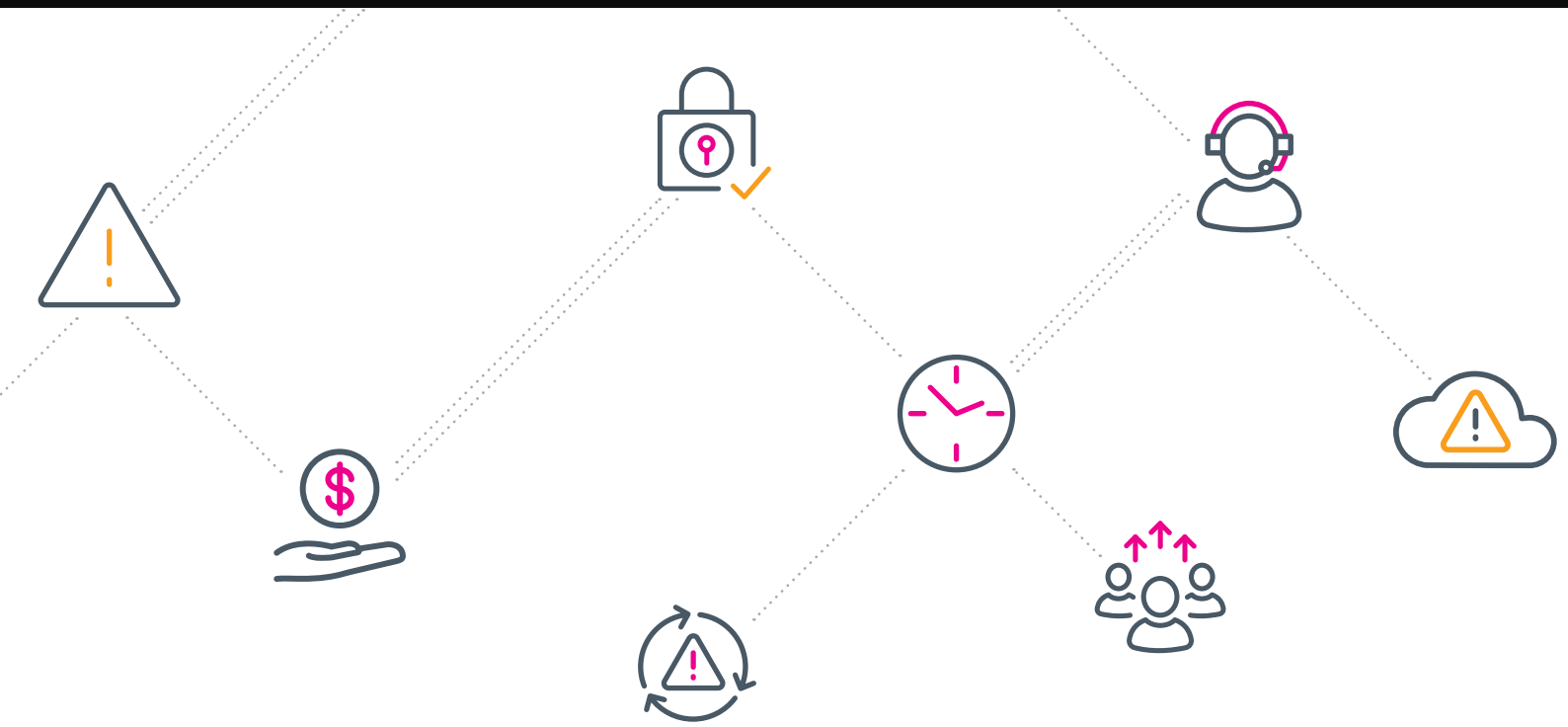
Der Leitfaden für SOAR-Käufer

Das Wer, Was, Wo, Wann und Warum rund um den Kauf einer SOAR-Lösung

splunk> turn data into doing™

Inhalt



1. Einleitung

Die Investition in eine SOAR-Plattform (Security Orchestration, Automation and Response) ist nicht nur eine kluge, sondern auch höchst strategische Entscheidung. Schließlich ist die Wahl der Plattform, auf der Sie Ihr SOC (Security Operation Center) aufbauen, zweifellos wichtiger als die Wahl eines beliebigen Sicherheitsprodukts als Punktlösung. Die von Ihnen gewählte SOAR-Plattform wird zu einem zentralen Bestandteil Ihrer Sicherheitsinfrastruktur und fungiert im Prinzip als Betriebssystem für Ihre Sicherheitsinvestitionen.

In diesem Leitfaden werden die wichtigsten Kriterien beschrieben, die Sie bei der Bewertung einer SOAR-Plattform anlegen sollten.

Definition von SOAR (Security Orchestration, Automation and Response)

Obwohl Automatisierung in anderen Softwarebereichen seit Jahren immer größeren Anklang findet – etwa in den Bereichen Vertrieb, Marketing, Personalwesen und IT – beginnen Sicherheitsteams gerade erst, die Vorteile der Automatisierung und Orchestrierung zu erkennen. Diese Erkenntnis steigert wiederum das Interesse an SOAR-Plattformen unter Einkäufern. Infolgedessen schwenken viele Sicherheitsanbieter in Richtung SOAR, um Marktanteile zu gewinnen, und nutzen ihre bestehenden Angebote aus benachbarten Marktsegmenten. Dies hat unglücklicherweise zur Folge, dass durch den zunehmenden Hype rund um SOAR immer mehr Anbieter mit unterschiedlichen Ansätzen die Marktdefinitionen verwischen und Angebotsvergleiche erschweren.

Um Klarheit zu schaffen, schlagen wir die folgenden Definitionen vor:

Sicherheitsorchestrierung
Die Sicherheitsorchestrierung ist die maschinenbasierte Koordination einer Reihe voneinander abhängiger Sicherheitsmaßnahmen innerhalb einer komplexen Infrastruktur.

Sicherheitsautomatisierung
Sicherheitsautomatisierung bezeichnet die maschinenbasierte Ausführung von Sicherheitsmaßnahmen.

Sicherheitsreaktion
Die Sicherheitsreaktion ist die richtlinienbasierte Koordination von manuellen (d.h. von Menschen durchgeführten) und maschinenbasierter Aktivitäten für Event-, Ticket- und Incident-Workflows.

Im folgenden Leitfaden werden wir diesen Definitionen folgen, während wir näher darauf eingehen, welche Fähigkeiten, Attribute und Aspekte zusammenwirken, um eine erstklassige SOAR-Plattform zu bilden.

Identifizieren von Sicherheitsanwendungsfällen

In der Regel identifizieren Teams Sicherheitsanwendungsfälle, die mit SOAR-Plattformen implementiert werden sollen. Die Anwendungsfälle orientieren sich an bestehenden manuellen Workflows und stellen meist die größten operativen Schwachstellen dar. Die Workflows umfassen häufig viele manuelle Aufgaben, und es muss oftmals in mehreren Produkten gearbeitet werden, um sie abzuschließen.

Vor Beginn der Bewertung ist es wichtig, neben den Schwachstellen auch alle potenziellen Anwendungsfälle auszuarbeiten. Hieran sollten die wichtigsten Stakeholder aus Ihrem Security Operations-Team beteiligt sein. Das Identifizieren umfassender Anwendungsfälle, selbst wenn sie nicht sofort implementiert werden, ist wichtig, um sicherzustellen, dass die gewählte Plattform auch künftige Anforderungen erfüllt.

Nachfolgend finden Sie eine Auswahl von Anwendungsfällen aus diversen Bereichen:

Alert Triage
Das Ziel von Alert Triage besteht darin, eingehende Benachrichtigungen zu prüfen und zu priorisieren. Anwendungsfälle, deren Schwerpunkt auf Alert Triage, also der Sichtung von Benachrichtigungen liegt, beinhalten zudem die Veredelung bzw. Anreicherung von Events mit zusätzlichem Kontext. Sie können auch Logik umfassen, die die weitere Verarbeitung höchst verlässlicher False Positive-Benachrichtigungen verhindert.

Incident Response
Incident Response-Anwendungsfälle können abhängig vom Incident-Typ stark variieren. Die Reaktion auf einen Phishing-Versuch unterscheidet sich beispielsweise stark von der Reaktion auf einen erfolgreichen Ransomware-Angriff.

Suche nach Kompromittierungsindikatoren (Indicator of Compromise, IOC)
Durch eine automatisierte Suche nach Kompromittierungsindikatoren können Teams die empfangenen Bedrohungsinformationen voll verwerten, anstatt die gesuchten IOCs aufgrund von Ressourcenbeschränkungen eingrenzen zu müssen. Sie könnten auch ein Intelligence-Scoring implementieren, das sie bei der Auswahl von Quellen für Bedrohungsinformationen unterstützt.

Schwachstellenmanagement
Die Automatisierung des Zyklus aus Identifizierung, Klassifizierung, Behebung und Minimierung von Schwachstellen steigert nicht nur die Team-Effizienz, sondern führt auch zu konsistenteren Ergebnissen, indem sichergestellt wird, dass der Prozess jedes Mal auf die gleiche Weise ausgeführt wird.

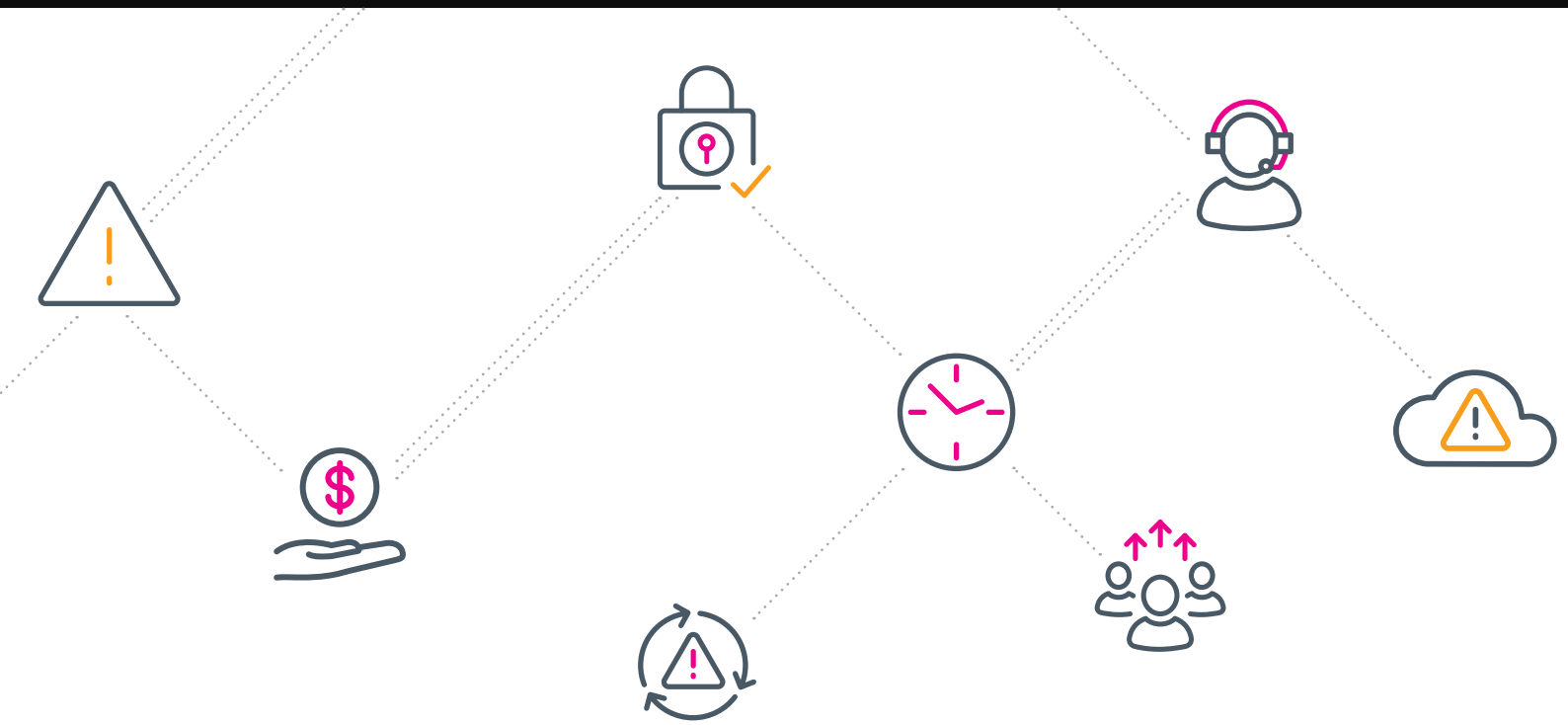
Netzwerkzugriffsteuerung
SOAR-Plattformen können dynamische Zugriffskontrollstrategien verbessern. Ein Beispiel dafür ist die Integration eines Erkennungssystems, das bisher nicht Teil der Entscheidungslogik der Netzwerkzugriffsteuerung war.

Benutzerverwaltung
Wenn sichergestellt ist, dass Benutzer präzise, schnell und systematisch aktiviert und deaktiviert werden, kann dies verhindern, dass ein Benutzerkonto böswillig von einem Bedrohungsakteur genutzt wird.

Penetrationstest
Aktivitäten wie Asset-Erkennung, Klassifizierung und Zielpriorisierung können automatisiert werden. Dies steigert die Produktivität des Penetrationstestteams.

Austausch von Sicherheitsinformationen
Unternehmen mit Initiativen zum Austausch von Sicherheitsinformationen können von automatisierten Playbooks enorm profitieren. Die Automatisierung kann auch die Produktivität eines Analysten steigern und zeitkritische Informationen schneller als manuelle Prozesse an eine Community zurückgeben.

Weitere Anwendungsfälle
Andere mögliche Anwendungsfälle für die Automatisierung ergeben sich aus bekannten Szenarien, in denen die Security Operations-Teams die Kriterien kodifizieren können, nach denen automatisch Entscheidungen getroffen und entsprechende Maßnahmen ergriffen werden.



2. Bewertungskriterien

Wir empfehlen, die SOAR-Bewertungskriterien in mindestens drei Abschnitte zu unterteilen: Kernfunktionalität, Plattformattribute und Unternehmensaspekte. Bei der Kernfunktionalität handelt es sich meist um Funktionen, die sich in der Plattform leicht identifizieren lassen. Plattformattribute sind weniger deutlich. Ein Beispiel hierfür sind etwa Architekturmerkmale, die Kriterien bilden, die sich auf die Plattformwahl auswirken. Unternehmensaspekte runden das Produktangebot ab und beinhalten Mehrwertdienste, die eine Firma zusätzlich zu ihrer Kerntechnologie bietet, wie etwa Schulungen und Support.

Kernfunktionalität

Die Kernfunktionalität können Sie sich als Grundbestandteil einer SOAR-Plattform vorstellen. Wir werden im Folgenden jede Funktion bzw. Komponente aufzählen und Überlegungen dazu angeben, um Sie bei Ihrer Bewertung und Auswahl zu unterstützen.

Orchestrator

Der Orchestrator sollte alle Aktivitäten im Zusammenhang mit einem bestimmten Sicherheitsszenario von Anfang bis Ende leiten und beaufsichtigen. Es ist in jedem Fall entscheidend, dass der Orchestrator konsistent vorhersehbare Ergebnisse und eine optimale Nutzung der verfügbaren Ressourcen liefert.

Datenintegration
Sicherheitsautomatisierung und -orchestrierung beginnt mit der Integration von Sicherheitsdaten. Ein Orchestrator sollte in der Lage sein, Sicherheitsdaten aus jeder Datenquelle und in jedem Format zu erfassen. Er sollte in der Lage sein, Daten zu empfangen, die an die Plattform übertragen werden, und muss Datenquellen abfragen sowie Daten in die Plattform einlesen können. Bei der Integration unstrukturierter Daten sollte der Orchestrator dem Benutzer ermöglichen, einen Daten-Handler bereitzustellen, der die Daten interpretiert und für die SOAR-Plattform nutzbar macht. Der Orchestrator sollte zudem in der Lage sein, Daten aus mehreren Quellen zu integrieren und die integrierten Daten logisch getrennt zu halten.

Entscheidungsmöglichkeit
Benutzer sollten die Möglichkeit haben, die Automatisierungs-Playbooks auszuwählen, die auf eine Datenquelle angewendet werden. So könnte beispielsweise ein E-Mail-Phishing-Playbook auf eine E-Mail-basierte Integrationsquelle angewendet werden, während ein Schadsoftware-Untersuchungs-Playbook auf eine Integrationsquelle angewendet wird, zu der es eine SIEM-Benachrichtigung gibt. Dieser Entscheidungsschritt steht in engem Zusammenhang mit den später beschriebenen Funktionen für die Verwaltung von Benachrichtigungen.

Ausführung von Aufgaben
Es ist in der Regel die Rolle des Orchestrators, Automatisierungsaufgaben zum richtigen und optimalen Zeitpunkt aus seiner Warteschlange zuzuteilen, indem er sie zur Ausführung an die Automatisierungs-Engine weiterleitet.

Überwachung durch Menschen
Ein Orchestrator sollte für ein ausgewogenes Verhältnis zwischen maschinenbasierter Automatisierung und der notwendigen Überwachung durch Menschen sorgen. Es gibt drei gängige Szenarien, in denen ein Analyst erforderlich ist: (a) wenn eine Genehmigung durch einen Asset-Besitzer erforderlich ist, um eine zielgerichtete Sicherheitsmaßnahme durchzuführen, (b) wenn die Überprüfung durch einen Analysten erforderlich ist, um sicherzustellen, dass es ein ausgeglichenes Verhältnis zwischen Sicherheit und Geschäftskontinuität gibt, und (c) wenn ein Analyst die kodifizierte Entscheidungslogik ergänzen muss (z. B. wenn ein Fehler auftritt).

Datenmanagement
Ein Orchestrator sollte auch sicherstellen, dass die Ausgabedaten einer Aktion richtig geparst, normalisiert und strukturiert werden, sodass sie von zukünftigen Aktionen genutzt werden können. Der Orchestrator sollte auch das Caching relevanter Daten unterstützen, wenn dies notwendig ist, um die Belastung anderer Ressourcen zu vermeiden.

Fehlertoleranz
Eine SOAR-Plattform interagiert regelmäßig mit vielen eigenständigen Produkten und Services, um Automatisierungs-Playbooks auszuführen. Ein Orchestrator muss damit rechnen, dass die Verfügbarkeit von Produkten und Services nicht immer garantiert ist. Der Zugriff auf externe Services kann unterbrochen oder gestört werden. In solchen Fällen sollte ein Orchestrator vorhersehbar agieren, den Zugriff wiederherstellen und den Vorgang gemäß der Konfiguration ordnungsgemäß fortsetzen.

Automatisierungs-Engine

Die Automatisierungs-Engine ist das Arbeitstier der meisten SOAR-Plattformen. Die Engine empfängt Aktionen oder Aufgaben vom Orchestrator und führt diese zuverlässig aus. Da Automatisierungsaufgaben unabhängig und weitgehend ohne menschliche Interaktion ablaufen, sind Eigenschaften wie Skalierbarkeit und Erweiterbarkeit der Plattform wichtige Kriterien.

Skalierbarkeit
Es ist wichtig zu verstehen, wie die Automatisierungs-Engine sowohl vertikal als auch horizontal skaliert. Man kann davon ausgehen, dass ein Benutzer im Laufe der Zeit mehr Anwendungsfälle automatisieren wird. Mit jedem weiteren Anwendungsfall entsteht zusätzliche Verarbeitungslast für die Automatisierungs-Engine. Die Automatisierungs-Engine sollte so konzipiert sein, dass eine vertikale Skalierung (z. B. Vergrößerung der CPU- und RAM-Ressourcen) und eine horizontale Skalierung (z. B. Erweiterung der Serverinstanzen) möglich sind, um die Leistung zu steigern und den Return on Investment (ROI) der Automatisierung zu schützen.

Erweiterbarkeit
Aufgrund der schnellen Entwicklung im Sicherheitsbereich sollte die Automatisierungs-Engine neue Funktionen ohne großes Re-Engineering unterstützen. Die Automatisierungs-Engine sollte die Fähigkeit zur Anpassung an die spezifischen Möglichkeiten ihrer Umgebung unterstützen.

Verwaltung von Benachrichtigungen

Gleich nach der bereits besprochenen Datenintegration sollte eine Benachrichtigungsverwaltungs-Funktion einer SOAR-Plattform eingehende Warnmeldungen in eine Warteschlange stellen und priorisieren, damit Analysten sie effizienter sichten können. Benachrichtigungsuntersuchungen sollten mit manueller oder automatisierter Aktionsausführung erfolgen, um die höchste Produktivität und Genauigkeit bei der Sichtung zu erzielen. Die Oberfläche einer Benachrichtigungsverwaltungs-Funktion sollte so aufgebaut sein, dass alle Aspekte einer Sicherheitsbenachrichtigung schnell erfasst und effiziente Reaktionen ermöglichen werden können. Die Oberfläche sollte Informationen zudem so anordnen, dass die richtigen Informationen zur richtigen Zeit angezeigt werden, damit Analysten keine umfangreiche Suchen durchführen oder den Kontext wechseln müssen.

Benachrichtigungsdetails
Die technischen Attribute einer Sicherheitsbenachrichtigung sollten so organisiert sein, dass ein Analyst sie schnell erfassen kann, um das Sicherheitsszenario zu verstehen. Dazu gehört auch eine organisierte Ansicht folgender Daten: IP-Adressen, Domänennamen, Datei-Hashes, Benutzernamen, E-Mail-Adressen und alle anderen relevanten Datenfelder. Die Verwendung eines Standardformats wie CEF (Common Event Format) o. ä. ist für den Datenaustausch sehr vorteilhaft.

Auslösen von Aktionen
Bei der Untersuchung einer Benachrichtigung sollte ein Sicherheitsanalyst in der Lage sein, manuelle Aktionen, die Benachrichtigungsdaten nutzen, durch die Plattform auszulösen. Dazu zählen Aktionen zur Untersuchung, Schadensbegrenzung und Korrektur sowie allgemeine Aktionen. Die Oberfläche sollte dem Benutzer ermöglichen, eine Aktion auszuführen, indem er die Daten auswählt, für die die Aktion durchgeführt werden soll. Dieses Verhalten wird manchmal auch als kontextbezogene Aktionsausführung bezeichnet und ermöglicht Pivot-Analysen rund um neu entdeckte Informationen.

Wie auch bei der manuellen Ausführung von Aktionen sollte ein Analyst die Möglichkeit haben, eine Sammlung von Aktionen zu einer Benachrichtigung auszulösen. Diese Sammlung von Aktionen wird meist als Playbook bezeichnet.

Aktionsergebnisse
Wenn manuelle oder automatisierte Aktionen zu einer Benachrichtigung ergriffen werden, sollten die Ergebnisse nicht nur für einen Analysten sichtbar und verwertbar sein, sondern auch für die SOAR-Plattform, die Aktionsergebnisse möglicherweise für automatisierte Entscheidungen verwendet. Aktionsergebnisse sollten sowohl in einem zusammenfassenden Format (z. B. in einer Tabellenansicht) als auch in einem umfassenderen Format (z. B. JSON) vorliegen.

Aktivitätsprotokoll
Die Plattform sollte ein umfassendes Aktivitätsprotokoll bereitstellen, das Aufzeichnungen aller Aktionen zeigt, die zu einer Benachrichtigung ausgeführt wurden, unabhängig davon, ob sie manuell oder über ein Automatisierungs-Playbook ausgelöst wurden. Jede Aktion sollte ihre Ergebnisse anzeigen, einschließlich eines Indikators für den Erfolg oder Misserfolg der Aktion, und deutlich machen, ob die Aktion vollständig ausgeführt wurde

Benachrichtigungsstatus, -schwergrad und -sensitivität
Jede von der Plattform verwaltete Benachrichtigung sollte einen Indikator bezüglich Status (z. B. neu, offen oder geschlossen), Schweregrad und Sensitivität (z. B. TLP-Bezeichnungen (Traffic

Light Protocol)) beinhalten. Jede dieser Angaben sollte sowohl innerhalb der Oberfläche zur Benachrichtigungsverwaltung als auch innerhalb eines Playbooks geändert werden können.

Zusammenarbeit an Benachrichtigungen
Die Schnittstelle sollte einen Bereich bieten, in dem Analysten zusammenarbeiten, Anmerkungen anbringen und verschiedene Informationen über eine Benachrichtigung bereitstellen können. Im Idealfall wird diese Zusammenarbeit aufgezeichnet und zusammen mit allen anderen Benachrichtigungsdaten erfasst und organisiert.

Ticket-Verwaltung

Nachdem Benachrichtigungen oder Events bestätigt und eskaliert wurden, sollte eine Ticket-Verwaltungskomponente einen breiter gefassten, funktionsübergreifenden Lebenszyklus von der Erstellung bis zur Lösung anstoßen. Diese Komponente sollte zusätzliche Attribute eines Tickets aufnehmen, die es von einer Benachrichtigung unterscheiden. Mehrere Benachrichtigungen können als einzelnes Ticket bestätigt, aggregiert und eskaliert werden. Die Benachrichtigungsverwaltung ist in der Regel technisch, während die Ticket-Verwaltung üblicherweise technische und nicht-technische Prozessschritte beinhaltet. Außerdem ist die Menge an Tickets meist geringer als bei Benachrichtigungen: Viele Unternehmen erhalten Hunderte oder Tausende Benachrichtigungen pro Tag, während sich die Zahl der Tickets pro Tag eher im einstelligen Bereich bewegt.

Organisation von Ticketdaten
Alle Daten zu einem Ticket sollten von der Ticket-Verwaltungskomponente zusammengefasst werden. Durch die Anzeige der Informationen an einem zentralen Ort können Benutzer sie effizient verwerten, ohne den Kontext wechseln zu müssen.

Hinzufügen von Daten zu einem Ticket
Die Oberfläche zur Ticket-Verwaltung sollte das Anhängen relevanter technischer Daten, wie etwa der Quelldaten und Aktionsergebnisse der Benachrichtigung, an das Ticket unterstützen. Zudem sollte die Oberfläche das Anhängen relevanter nicht-technischer Daten wie Notizen, Memos, E-Mails, Screenshots, Aufzeichnungen oder beliebiger anderer Dateien mit Relevanz für das Ticket unterstützen. Das automatisierte Anhängen von Informationen an ein Ticket sollte auch aus einem Playbook heraus möglich sein.

Verknüpfen von Tickets mit Benachrichtigungen
Bei einer Ticket-Untersuchung werden oftmals Daten identifiziert, die weiter untersucht werden sollten, oder ein Szenario festgestellt, das die Durchführung einer sofortigen Aktion zur Schadensbegrenzung erfordert. Wenn ein Analyst daher feststellt, dass eine Aktion durchgeführt werden sollte, sollte die Ticket-Verwaltungsoberfläche dem Analysten eine nahtlose Verknüpfung zum Benachrichtigungsverwaltungs-Interface der jeweiligen Benachrichtigung bieten. In der Oberfläche bzw. dem Interce zur Benachrichtigungsverwaltung können weitere Aktionen durchgeführt werden, und Änderungen an relevanten Daten sollten in der Ticket-Verwaltungsoberfläche widergespiegelt werden.

Zuordnung zu bestehenden Prozessen
Viele Unternehmen haben standardmäßige Betriebsabläufe (Standard Operating Procedures, SOPs) für die Reaktion auf Incidents, Notfälle, Katastrophen und andere kritische Situationen entwickelt. Die Ticket-Verwaltungsfunktion sollte dem Benutzer ermöglichen, an seinem Prozess ausgerichtete Phasen zu definieren und als Vorlage zu speichern. Der Benutzer sollte die Möglichkeit haben, den SOP in mehrere Phasen aufzuteilen, wobei jede Phase eine oder mehrere Aufgaben beinhaltet und jede Aufgabe einem Verantwortlichen zugeordnet werden kann. Zusätzliche Kontextinformationen, die mit einer Aufgabe verknüpft sind, können in die Aufgabenbeschreibung aufgenommen werden. Ähnlich wie bei Anwendungen zur Aufgabenverwaltung sollten Aufgaben als abgeschlossen gekennzeichnet werden, wenn sie von jeweils zugewiesenen Person erledigt wurden. Die Oberfläche sollte eine Fortschrittsanzeige für das Ticket sowie den Ticketstatus anzeigen.

Aktivitäten-Auditing
Hinzugefügte Informationen oder Änderungen sowie Statusänderungen sind wichtige Ticket-Details. Jede Änderung an einem Ticket sollte in einem Audit-Trail protokolliert werden und exportiert werden können.

Änderungen an einem Ticket sind beispielsweise:

- Hinzufügen von Daten
- Ändern von Daten
- Ändern einer Phase oder Aufgabe
- Hinzufügen von Dateien oder Notizen
- Ändern von Dateien oder Notizen
- Abschließen einer Aufgabe
- Andere Aktivitäten oder Änderungen am Ticket

Playbook-Verwaltung

Die Playbook-Verwaltung unterstützt Sie bei der Pflege von SOPs. Im Idealfall sollte diese Komponente eine Revisionskontrolle und die Möglichkeit bieten, die Syndizierung von SOPs in Form von Playbooks innerhalb eines Unternehmens und möglicherweise innerhalb einer Community zu verwalten.

Playbook-Organisation

Die Playbook-Verwaltung sollte eine angemessene Organisation und Gruppierung von Playbooks ermöglichen. Benutzer sollten eigene Gruppierungen speziell für die Anforderungen ihres Unternehmens definieren können. Sie könnten dann beispielsweise Playbooks nach Themen, Sensitivität, Organisationsbereichen oder Asset-Typen organisieren und gruppieren.

Benutzerdefinierte Funktionen

Benutzerdefinierte Funktionen machen die Erstellung und Ausführung von Playbooks schneller und einfacher und ermöglichen es Sicherheitsanalysten, die Erkennung, Untersuchung und Reaktion auf Bedrohungen zu beschleunigen. Schreiben Sie Ihre eigenen benutzerdefinierten Funktionen, oder nutzen Sie unsere vorkonfigurierte Funktionsbibliothek, um eine schnellere Wertschöpfung zu erzielen. Benutzerdefinierte Funktionen lassen sich außerdem problemlos gemeinsam im Team nutzen und in mehreren Playbooks wiederverwenden, was die Teameffizienz erhöht und die Vielseitigkeit von Playbooks maximiert, um zusätzliche Sicherheitsprozesse zu automatisieren.

Bulk Edits von Playbooks

Die Abläufe innerhalb eines Playbooks sind mit großer Wahrscheinlichkeit spezifisch. Es gibt jedoch bei vielen Playbooks Gemeinsamkeiten auf Verwaltungsebene.

Ein Playbook-Verwaltungssystem sollte die Massenbearbeitung von Playbooks in folgender Hinsicht ermöglichen:

- Integrationsquellen
- Aktivieren/Deaktivieren der automatischen Ausführung der Aktivierung/Deaktivierung des Betriebs im sicheren Modus
- Aktivieren/Deaktivieren der erweiterten Protokollierung
- Festlegen der Playbook-Gruppierung in Kategorien

Revisionskontrolle und Verteilung

Die Integration mit einem Versionskontrollsystem wie Git empfiehlt sich unbedingt für eine erfolgreiche Playbook-Verwaltung im großen Stil. Was die Bereitstellung angeht, so ermöglicht die Nutzung eines Versionskontrollsystems die systematische Verteilung von Playbooks auf mehrere Systeme. Dies ist nützlich für die Synchronisierung von Playbooks zwischen Entwicklungs- und Produktionssystemen oder für die Synchronisierung über mehrere Produktionssysteme an mehreren Standorten hinweg. In Bezug auf die Entwicklung ist ein Versionskontrollsystem wichtig für die Nachverfolgung von Revisionsänderungen und gibt zudem die Möglichkeit, Änderungen bei Bedarf wieder aufzuheben. Ein weiterer Vorteil besteht darin, dass Entwickler Playbooks im Editor ihrer Wahl bearbeiten und die geänderten Versionen problemlos wieder mit der Plattform synchronisieren können.

Automatisierungseditor

Im Automatisierungseditor kodifizieren Analysten oder Führungskräfte ihre Prozesse in Automatisierungs-Playbooks. Der Vorgänger des visuellen Automatisierungseditors war der einfache Quellcode-Editor. Als automatisierte Playbooks ausschließlich in einem Quellcode-Editor bearbeitet werden konnten, war die Erstellung von Playbooks ein mühsamer und schwieriger Vorgang, den nur relativ wenige Programmierer meisterten. Ein visueller Automatisierungseditor ermöglicht es allen Sicherheitsexperten, umfassende und anspruchsvolle Playbooks zu erstellen – auch wenn sie möglicherweise nicht das Knowhow hätten, Playbooks auf Quellcode-Ebene zu schreiben. Der visuelle Editor sollte den BPMN-Standards (Business Process Model and Notation) entsprechen. Bei BPMN handelt es sich um eine grafische Spezifikationssprache für Geschäftsprozesse. Die BPMN unterstützt intuitive Symbole für Business-Anwender und gibt technischen Nutzern die Möglichkeit, äußerst komplexe Prozesse abzubilden.

Benutzeroberflächenelemente

Die Elemente des User Interface sollten zuallererst eine Arbeitsfläche umfassen, auf der visuelle Playbooks erstellt werden können. Dieser Teil der Oberfläche sollte einen Bereich bieten, in dem eine gewünschte Aktion angegeben werden kann (z. B.

block_ip oder file_reputation). Nach der Auswahl einer Aktion sind wahrscheinlich Parameter erforderlich, um die Aktion richtig zu konfigurieren. Die Oberfläche sollte die Möglichkeit bieten, Parameter manuell einzugeben oder in einer Liste auszuwählen. Benachrichtigungsdaten und/oder Aktionsergebnisdaten werden eventuell ebenfalls als Parameter genutzt.

Die Schnittstelle sollte zudem einen Bereich bieten, in dem das Testen und Debuggen stattfinden kann, damit der Übergang vom Bearbeitungsmodus zum Testmodus nahtlos erfolgen kann. Außerdem sollte eine Quellcodeansicht verfügbar sein, falls der Benutzer den Quellcode für das automatisierte Playbook anzeigen möchte.

Blockbasierte Codedarstellung
Die Verwendung von Blöcken zur Darstellung sinnvoller Schritte in der Automatisierungsplattform ermöglicht es Benutzern, umfassende, komplexe Playbooks zu schreiben, ohne den zugrunde liegenden Quellcode anzufassen. Blöcke sollten auf 1:1-, 1:n- oder n:1-Weise verbunden werden, um die Reihenfolge der Ausführung festzulegen. Der Benutzer sollte visuell ein Playbook erstellen können, das Aktionsausführungen, Plattform-API-Aufrufe, Bedingungsanweisungen (if/then) und verzweigte Anweisungen enthält, die ein Playbook mit einem anderen verbinden.

Einbinden von Menschen in den Entscheidungsprozess
Beaufsichtigte Automationsunterstützung ist eine gängige Anforderung. Sie ermöglicht Menschen in einen Automatisierungsablauf einzubinden, um die weitere Playbook-Ausführung zu genehmigen, zu prüfen oder zu erweitern. Der Automatisierungseditor sollte diesen Überwachungsschritt durch Menschen unterstützen, indem er ermöglicht, in einem Playbook Genehmigungspunkte neben einer oder mehreren Sicherheitsaktionen einzufügen. Ein Playbook-Autor sollte die Möglichkeit haben festzulegen, welche Person(en) in die Automatisierungsschleife eingefügt werden sollen, und auch die Angabe der Art der gewünschten Benachrichtigung oder Genehmigung ermöglichen. Der Playbook-Editor und die zugrunde liegende Plattform sollten eine Fehlerbehandlungslogik definieren können, die einen Menschen in die Automatisierungsschleife einbinden würden, wenn beispielsweise ein oder mehrere Reputationsservices nicht verfügbar sind, um die Entscheidungsfindung zu unterstützen.

Informationsaustausch mit Aktionsergebnissen
Die Oberfläche des Automatisierungseditors sollte die Möglichkeit bieten, neue Informationen, die sich aus vorhergehenden Aktionsausführungen ergeben, als Eingaben oder Parameter für nachgelagerte Aktionen oder Entscheidungsblöcke zur Verfügung zu stellen. Die Aktionsergebnisse vorhergehender Aktionen sollten visuell zur Verfügung stehen und aus einer Dropdown-Liste ausgewählt werden können, wenn die Parameter einer vorgeschalteten Aktion ausgefüllt werden.

Zugriff auf Playbook-Quellcode
Beim Erstellen des Playbooks in einem visuellen Editor sollte der daraus resultierende Playbook-Quellcode in Echtzeit erzeugt werden und dem Autor zugänglich sein. Manche Benutzer ziehen es vor, das Playbook ganz oder teilweise mit einer traditionellen Quellcode-Methode zu erstellen. Die Oberfläche sollte die Möglichkeit bieten, einen visuellen Editor auszublenden und durch einen Quellcode-Editor zu ersetzen. Der Wechsel zwischen visuellem und Quellcode-Modus sollte nahtlos und mühelos erfolgen.

Gleichzeitige visuelle und nicht-visuelle Playbook-Erstellung
Bei der Arbeit mit dem Quellcode eines Playbooks sollte der Automatisierungseditor dem Autor ermöglichen, das Playbook auf Quellcode-Ebene zu bearbeiten, und ihm gleichzeitig weiterhin die Möglichkeit geben, es auf der visuellen Blockebene zu ändern. In manchen Fällen muss der Autor einzelne Blöcke (wie etwa Aktionen oder Entscheidungsblöcke) auf Quellcode-Ebene ändern, um Anpassungen umzusetzen, die über die Möglichkeiten des visuellen Editors hinausgehen. Wenn diese Änderungen abgeschlossen sind, sollte der Benutzer weiterhin die Freiheit haben, das Playbook visuell zu bearbeiten.

Integriertes Testing und Debugging sowie Laufzeitprotokollierung
Standardmäßig bieten integrierte Entwicklungsumgebungen (Integrated Development Environments, IDEs) Ausführungs- und Debugging-Möglichkeiten. Bei einer Automatisierungsplattform sollte der Benutzer die Möglichkeit haben, das Playbook für eine Sicherheitsbenachrichtigung auszuführen und die Ausführungsaktivitäten und -ergebnisse zu beobachten. Protokolleinträge und Fehlercodes sollten in einem Debug-Fenster aufgeführt werden, das parallel zum visuellen Block-Editor oder Quellcode-Editor angezeigt werden kann,

falls der Autor die Quellcode-Methode vorzieht. Das Ziel ist es, damit dem Autor zu ermöglichen, Playbooks innerhalb einer Oberfläche schnell zu bearbeiten, zu testen und zu debuggen.

Sicherer Modus
Ein Automatisierungseditor sollte zudem einen sicheren Modus für neue Playbooks bieten, die vor der Umstellung in den Produktionsstatus getestet werden müssen. Dieser Modus simuliert die Ausführung für Automatisierungsziele, ohne diese wirklich zu ändern. Sobald der Autor oder ein anderer Plattformnutzer genügend Vertrauen in die Logik des Playbooks gewonnen hat, kann dieser sichere Modus deaktiviert werden und das Playbook ohne Einschränkungen eingesetzt werden.

App Framework

Das App-Framework bietet eine umfangreiche Oberfläche für neue Integrationen, die die Plattform mit beliebigen der Tausenden von Punktlösungen verbindet, die heute auf dem Sicherheitsmarkt angeboten werden.

Offenes Ökosystem
Eine SOAR-Plattform kann mit der Zeit an Wert verlieren, wenn sie keine Integrationslösungen für neue Produktangebote bietet. Um eine vorhersehbare Roadmap für App-Integrationen sicherzustellen, sollte eine Plattform ein offenes Ökosystem nutzen, das es jedem ermöglicht, Integrationen zu entwickeln. Dies gibt Benutzern die notwendige anbieterunaghängige Freiheit. Technologien können ausprobiert und verworfen werden, ohne dass sich dies negativ auf automatisierte Playbooks auswirkt. Neue Technologien müssen schnell in die Plattform integriert werden, ohne dass dazu eine Änderung der Kernplattform notwendig wird. Letztendlich müssen Benutzer die Möglichkeit haben, Unterstützung für zusätzliche Plattformen zu schaffen, ohne sich auf den SOAR-Anbieter für zusätzliche Entwicklungen verlassen zu müssen.

Metriken und Berichte

Metriken und Berichte sind für jede Automatisierungsplattform wichtig und SOAR-Plattformen bilden da keine Ausnahme. Die Automatisierung verspricht mehr Produktivität und höhere Qualität. Metriken sind entscheidend dafür, die Effektivität der Automatisierungsplattform zu bewerten und Verbesserungsmöglichkeiten zur Steigerung des ROI zu erkennen.

Flexible Dashboards
Metriken werden spezifisch für Unternehmen und Einzelpersonen definiert. Aufgrund dieser Tatsache müssen Benutzer ihre Metriken so organisieren können, wie es für ihr Unternehmen am sinnvollsten ist. Die SOAR-Plattform sollte die Möglichkeit bieten, Metrikdaten höchst individuell zu organisieren. Dazu gehört die Konfiguration der Reihenfolge, in der Informationen im Dashboard angezeigt werden, sowie die Angabe, welche Metriken in welchen Zeitfenstern angezeigt werden.

Leistungsberichte
Automatisierung wird eingesetzt, um die operative Effizienz zu steigern. Es ist daher wichtig, den quantitativen Leistungsanstieg und die Ressourceneinsparungen durch die Automatisierung zu verstehen und diese Informationen über ein Dashboard jederzeit verfügbar zu haben.

Dies sind einige Beispiele für wichtige Leistungsmetriken, die in der Plattform zur Verfügung stehen sollten:

- MTTR (Mean Time To Resolve)
- MDT (Mean Dwell Time), die als der Zeitraum zwischen einer Kompromittierung (durch einen Angreifer) und dem Ergreifen einer geeigneten Maßnahme liegt
- Analystenstunden, die durch die automatisierte Ausführung eingespart werden
- Anzahl der Vollzeitäquivalente (Full Time Equivalent, FTE), die durch die automatisierte Ausführung gewonnen wird
- Durchschnittliche Zeitersparnis bei jeder Playbook-Ausführung
- Kosteneinsparungen (FTE-Kosten x gewonnene FTEs)

Bericht zur Sicherheitseffektivität
Automatisierung wird auch eingesetzt, um die Sicherheitseffektivität und das Sicherheitsniveau des Unternehmens zu erhöhen. Um die Sicherheitseffektivität der Automatisierung zu beurteilen, muss man die Gesamtzahl der verwalteten Sicherheitsbenachrichtigungen und die Geschwindigkeit kennen, in der sie verwaltet werden.

Hier sind einige Beispiele für wichtige Metriken zur Sicherheitseffektivität, die die Plattform bereitstellen sollte:

- MTTR und MDT (Erläuterung siehe oben)
- Gesamtzahl der offenen Benachrichtigungen
- Pro Tag geöffnete Benachrichtigungen (auch Angaben pro Stunde, Woche oder Monat sind sinnvoll)
- Pro Tag geschlossene Benachrichtigungen (auch Angaben pro Stunde, Woche oder Monat sind sinnvoll)
- Leistung in Bezug auf Service Level Agreements (SLAs)

App-Integration und Playbook-Performance
Wenn man die am häufigsten aufgerufenen Playbooks versteht, kann dies deutlich machen, wo weitere Automatisierungsinvestitionen umgesetzt werden können. Im Idealfall ist das Playbook-Design so konzipiert, dass verlässliche False Positive- oder höchst verlässliche True Positive-Benachrichtigungen automatisch behandelt und geschlossen werden. In Fällen, in denen die Automatisierung die Sichtungslücke bei Benachrichtigungen nicht schließt, sollte das Playbook eventuell überarbeitet werden.

Damit Automatisierungslücken erkannt und die Wirksamkeit von Tool-Integrationen beurteilt werden können, sollte die Automatisierungsplattform folgende Beispielmetriken bereitstellen:

- Benachrichtigungen, die durch Automatisierung geschlossen werden (pro Stunde, Tag, Woche, Monat oder einem anderen Zeitfenster)
- Aktivste App-Integrationen
- Aktivste Aktionen (manuell und automatisiert)
- Aktivste automatisierte Playbooks
- Playbook-Ausführungsdauer
- Aktionsausführungsdauer
- Menschliche Arbeitslast

Die Automatisierung soll zwar den Mangel an menschlichen Ressourcen ausgleichen, doch es gibt immer noch Fälle, in denen Menschen in die täglichen Aktivitäten einer SOAR-Plattform einbezogen werden müssen. Dies ist beispielsweise der Fall, wenn die manuelle Sichtung oder andere Aktionen für eine Benachrichtigung notwendig sind oder für die „überwachte Automatisierung“ menschliche Genehmigungen in das Playbook eingefügt wurden. Wenn man die menschliche Arbeitslast kennt, lassen sich auch leichter Bereiche identifizieren, in denen eventuell weitere Automatisierung und Optimierung erforderlich sind. Die folgenden Beispielmetriken sollten von der Automatisierungsplattform bereitgestellt werden, um die menschliche Arbeitslast innerhalb des Automatisierungsprozesses aufzuzeigen:

- Einer Einzelperson zugewiesene Benachrichtigungen
- Von einer Einzelperson geschlossene Benachrichtigungen
- Durchschnittliche Genehmigungsdauer
- Anzahl ausstehender Genehmigungen
- Erforderliche Genehmigungen (pro Stunde, Tag, Woche, Monat oder einem anderen Zeitfenster

Plattformattribute

Wie bereits erwähnt, können Plattformattribute auch eher qualitativer Art sein. In Anbetracht dessen werden diese Kriterien häufiger durch Beobachtung und Interaktion mit der Plattform bewertet.

Bereitstellungsoptionen

Eine SOAR-Plattform sollte lokale, Cloud-basierte oder Hybrid-Bereitstellungen unterstützen. Manche Sicherheitsexperten bevorzugen lokale Installationen, andere dagegen Cloud-Bereitstellungen. Wählen Sie für Ihre SOAR-Lösung die Bereitstellungsart, die die Anforderungen Ihres Unternehmens am besten erfüllt, die Sicherheitsprozesse optimiert und die digitale Transformation erleichtert.

Unterstützung durch die Community

Der Fokus einer SOAR-Plattform auf die Security Operations-Community ist entscheidend für ihren langfristigen Erfolg. Es gibt viel zu viele Produkte, Services und Plattformen, als dass ein Unternehmen Integrationslösungen für alle bereitstellen könnte. Die Entwicklungen im Bereich der Sicherheit machen es auch notwendig, dass eine Gemeinschaft von Fachleuten zusammenarbeitet, um Playbooks, Best Practices und Strategien für den Umgang mit den neuesten Bedrohungen auszutauschen. Eine SOAR-Plattform muss daher ein starkes Community-Modell unterstützen und die Freigabe von App-Integrationen und Playbooks erleichtern.

Große und aktive Community
Die Installationszahlen einer Plattform sind ein guter Gradmesser für das Zusammenarbeitspotenzial der zugehörigen Community. Die meisten Benutzer verlassen sich gerne auf die Erfahrungen anderer,

gleichgesinnter Benutzer. Eine große, aktive Benutzer-Community bietet die Möglichkeit, Playbooks, Apps oder Brainstorming-Ideen für neue Automatisierungs-Anwendungsfälle auszutauschen. Darüber hinaus ist die Beteiligung von Anbietern an der Community ein guter Indikator für ihr Engagement sowohl für die Community als auch für die Zusammenarbeit.

Um den Ideenaustausch zu erleichtern, ist es wichtig, den Community-Benutzern den Dialog miteinander zu ermöglichen. Dazu wird meist ein Community-Kommunikationstool wie Slack bereitgestellt, das Direkt- und Gruppennachrichten ermöglicht. Messaging-Tools sind sehr effektiv, was technischen und Design-Support, schnelle Antworten auf Fragen und Brainstorming zu Automatisierungs-Anwendungsfällen angeht. Weitere Kommunikationstools zur Verbreitung neuer Ideen sind Github-Seiten von Community-Benutzern, auf denen Einzelpersonen ihre Arbeit veröffentlichen, sowie ein zentrales Community-Repository, das Benutzerpräsentationen, Community-Playbooks und App-Integrationen hostet.

Zusammenarbeit

Die Zusammenarbeit verbessert die Substanz einer Plattform erheblich durch Vollständigkeit der Funktionen, Abdeckung der App-Integration und automatisierte Playbooks für eine wachsende Zahl von Szenarien.

Zusammenarbeit über die Community hinweg
Was den Inhalt angeht, so sollten von Benutzern und Anbietern beigesteuerte Inhalte in einem zentralen Repository zur Verfügung stehen, das für alle Benutzer in der Community zugänglich ist. Dies schließt technische Beiträge wie Playbooks und App-Integrationen ebenso wie nicht-technische Beiträge wie Präsentationen, technische Hinweise (Tech Notes), Blogs und andere Dokumentationsmethoden mit ein. Das Informations-Repository sollte parallel zur Community-Größe ebenfalls ständig an Größe zunehmen.

Zusammenarbeit über die Plattform hinweg
Das Puzzle der Zusammenarbeit ist unvollständig, wenn die Plattform keine Möglichkeit zur Zusammenarbeit bietet. Die SOAR-Plattform sollte den Benutzern die Möglichkeit bieten, die Zusammenarbeit innerhalb verschiedener vertrauenswürdiger Kreise zu nutzen. Die Plattform muss die sensibelste Arte der Zusammenarbeit unterstützen: die Kommunikation innerhalb des Sicherheitsteams des Unternehmens. Dazu zählen eine integrierte Chat-Funktion sowie das Anhängen von Notizen und zugehörigen Daten an Benachrichtigungen und Tickets.

Kognitiv

Eine kognitive SOAR-Plattform sollte das Wissen von Menschen und frühere Beobachtungen nutzen, um künftige Entscheidungen zu lenken. Menschliches Wissen muss im System kodifiziert werden, um eine Automatisierung in Form von Playbooks zu ermöglichen. Die Plattform könnte auch frühere Beobachtungen nutzen, indem sie Ausführungsstatistiken, Eigenschaften integrierter Daten und Aktionsergebnisse nachverfolgt. Diese Informationen können genutzt werden, um einzelne Aktionen, Playbooks oder Aktionsgruppen in einer Abfolge zu empfehlen, die ein Playbook bilden würde. Es ist daher wichtig, die aktuellen kognitiven Fähigkeiten einer SOAR-Plattform sowie die kognitive Strategie und Roadmap für zukünftige Versionen der Plattform zu kennen.

Wählbare Automatisierung
Die Integration der Automatisierung bei Sicherheitsabläufen nimmt in der Regel mit der Zeit zu. Meistens nutzen Teams die Automatisierung nur bei jeweils einem Anwendungsfall, um erst Vertrauen zu der Plattform aufzubauen. Um das Vertrauen in die Automatisierung zu stärken, sollte die SOAR-Plattform eine Reihe von Funktionen unterstützen, die die selektive menschliche Interaktion mit dem automatisierten Playbook ermöglichen.

Das Einbinden von Menschen in einen Workflow sollte jeweils auf der Basis einzelner Assets (Sicherheitspunktlösung oder -technologie) oder einzelner Aktionen möglich sein. Die Einbindung pro Asset sollte die Möglichkeit bieten, den Asset-Administrator jedes Mal zu benachrichtigen, wenn eine Aktion für dieses Asset ausgeführt wird. Die Einbindung pro einzelner Aktion sollte erreicht werden, indem eine Aufforderung für eine menschliche Eingabe an einem beliebigen Punkt in ein automatisiertes Playbook eingefügt wird. Diese Aufforderung gibt dem Empfänger die Wahl, die Ausführung fortzusetzen, zu unterbrechen oder abzubrechen. Dieses Maß an Überwachung ermöglicht Benutzern, Zutrauen in die programmierten Automatisierungsschritte zu fassen.

Sicher

Einer der wichtigsten Aspekte einer Sicherheitsautomatisierungs- und -orchestrierungsplattform ist die eigene Sicherheit. Da eine SOAR-Plattform Authentifizierungsdaten und andere hochsensible Informationen enthält, ist sie abgesichert, verschlüsselt sensible Informationen und unterstützt eine robuste, rollenbasierte Zugriffskontrolle.

Wichtige Security Best Practices, auf die Sie bei einer SOAR-Plattform achten sollten, sind:

- Verschlüsselte Authentifizierungsdaten
- Authentifizierungsdaten werden nicht im Speicher abgelegt
- Unterstützung für Authentifizierungsverwaltungssysteme
- Unterstützung Multi-Faktor-Authentifizierung

Skalierbar

Es ist wichtig zu verstehen, wie eine SOAR-Plattform sowohl vertikal als auch horizontal skaliert. Mit jedem weiteren Anwendungsfall, den ein Unternehmen im Lauf der Zeit hinzufügt, entsteht zusätzliche Verarbeitungslast für die Plattform. Die Plattform sollte so konzipiert sein, dass eine vertikale Skalierung durch Vergrößerung der Hardwareressourcen (z. B. CPU und RAM) und eine horizontale Skalierung durch die Erweiterung der Serverinstanzen, die die Installation unterstützen, möglich sind.

Offen und erweiterbar

Der gesamte Bereich der IT-Sicherheit entwickelt sich ständig weiter, was sich auch an der Vielzahl der heute verfügbaren Punktlösungen zeigt. Eine SOAR-Plattform sollte offen und erweiterbar angelegt sein. Sie sollte die Integration neuer Sicherheitsszenarien, neuer Produkte, neuer Aktionen und neuer Playbooks problemlos unterstützen.

Offenes Integrations-Framework

Der Effekt eines offenen Integrations-Frameworks besteht im Prinzip darin, dass Technologien in die Plattform integriert bzw. daraus entfernt werden können, ohne dass sich dies negativ auf die automatisierten Abläufe auswirkt.

Benutzer sollten zudem die Freiheit haben, Unterstützung für zusätzliche Integrationen zu entwickeln, ohne vom SOAR-Anbieter abhängig zu sein. Dies bedeutet, dass Benutzer die Möglichkeit haben sollten, ggf. eigene Integrationen zu schreiben. Gute Beispiele für solche Fälle sind selbst erstellte Anwendungen, eine benutzerdefinierte oder vorab für den Zugriff bereitgestellte API eines Herstellers oder die Erweiterung der Funktionalität der Automatisierungsplattform. Dieses offene Framework sollte einem gängigen Standard- und Programmiermodell folgen. Dazu sollte es jede Menge Dokumentation und Beispiele geben.

Keine Schnittstelleneinschränkungen

Manche Technologien stellen Schnittstellen mit REST-APIs, SSH, Syslog, benutzerdefinierten APIs oder anderen Protokollen bzw. Methoden zur Verfügung. Bei einem erweiterbaren Integrations-Framework sollten keine Einschränkungen hinsichtlich des Schnittstellentyps gelten. Wenn von der Automatisierungsplattform eine Verbindung zur Punktlösung oder der Anwendung besteht, sollte die Schnittstellenmethode keinen Einfluss auf die App-Integration haben, sodass eine beliebige Schnittstellenmethode verwendet werden kann.

Mobilität

SOAR-Plattformen sind dafür gemacht, Reaktionszeiten zu beschleunigen, d. h. MDT und MTTR zu reduzieren. Um schnell reagieren zu können, müssen Sicherheitsanalysten erreichbar sein, sobald menschliches Eingreifen aufgrund eines Sicherheitshinweises erforderlich ist. Aber Analysten sitzen nicht den ganzen Tag am Schreibtisch und warten darauf, einzugreifen. Deshalb ist es wichtig, dass eine SOAR-Plattform Zugang, Interaktivität und Kontrolle vom mobilen Gerät des Analysten aus bietet. So können Analysten auch ganz leicht von unterwegs Playbooks ausführen und Ereignisse überprüfen, ohne dabei einen Laptop öffnen zu müssen.

Benutzerfreundlich

Obwohl Unternehmenssoftware nie einfach ist, ist es dennoch möglich, die Reibungsverluste bei Bereitstellung und Verwendung einer SOAR-Plattform gering zu halten.

Installation und Einrichtung

Der Virtual Appliance-Formfaktor macht die Bereitstellung einfach, da die meisten Unternehmen Virtualisierung bereits bei anderen Infrastrukturelementen nutzen.

Einarbeitung

Eine SOAR-Plattform kann erheblich dazu beitragen,

Anfangsschwierigkeiten zu meistern, indem sie einen Einarbeitungsprozess bietet, der Benutzern hilft, Systemeinstellungen zu konfigurieren, sich mit Datenquellen zu verbinden und die ersten Playbooks zu aktivieren.

Verkürzen der Zeit bis zur Automatisierung

Eine SOAR-Plattform sollte Benutzern ermöglichen, schnell mit der Automatisierung zu beginnen. Dies lässt sich durch die Bereitstellung robuster, direkt einsetzbarer automatisierter Playbooks erreichen. Wenn Benutzer die Möglichkeit haben, automatisierte Playbooks schnell zu erstellen, zu testen und bereitzustellen, ist dies ein weiterer wichtiger Beschleunigungsfaktor.

Eine visuelle IDE macht es Nicht-Programmierern leicht, Playbooks zu erstellen und zu bearbeiten. Eine SOAR-Plattform kann dies durch Bereitstellen eines Tools erreichen, mit dem Automatisierungs-Playbooks visuell erstellt oder geändert werden können. Die visuelle Erstellung von Playbooks verkürzt die Entwicklungsdauer und steigert die Qualität, da Coding-Fehler vermieden und ein Coding-Standard festgelegt wird, der für Konsistenz unter Playbooks sorgt.

Unternehmensaspekte

Unabhängig davon, wie gut die Kerntechnologie eines Unternehmens ist, gibt es Überlegungen, die über das hinausgehen, was traditionell als das „Produkt“ angesehen wird und den Entscheidungsprozess eines Käufers stark beeinflussen. Ein wichtiger Aspekt sind die Eigenschaften, die mit dem anbietenden Unternehmen verknüpft werden. Eine weitere wichtige Überlegung dreht sich um die Services, die das Unternehmen anbietet und die die Kerntechnologie zu einem Gesamtpaket ergänzen, das der Käufer letztlich erlebt.

Eigenschaften des Unternehmens

Bei einer Kaufentscheidung ist es wichtig, das Profil, die Qualität und das Zukunftspotenzial des von Ihnen gewählten Unternehmens zu berücksichtigen. Es ist eine Tatsache, dass viele neue Unternehmen mit neuen Lösungen in neuen Marktsegmenten scheitern. Sie sollten ein Unternehmen wählen, das die Kraft hat, die gegebenen Versprechen auch einzuhalten.

Firmengeschichte

Das gewählte Unternehmen sollte über langjährige Erfahrung in der Entwicklung von Sicherheitslösungen verfügen. SOAR ist zwar ein relativ neues Marktsegment, doch seine Ursprünge liegen Jahrzehnte zurück. Sie sollten sich unbedingt darüber informieren, wie das Unternehmen gegründet wurde und wie es sich für SOAR als Marktsegment entschieden hat.

Ausführungsfähigkeit

Sie sollten ein Unternehmen suchen, das auf ein erfahrenes Team qualifizierter Experten zurückgreifen kann. Die Vorhersage zur zukünftigen Ausführungsfähigkeit eines Unternehmens ist mit großer Wahrscheinlichkeit direkt an die bisherige Erfolgsbilanz der Teammitglieder gekoppelt.

Kundenstamm

Qualität und Profil des Kundenstamms eines Unternehmens lassen Rückschlüsse auf das Unternehmen als solches zu. Anspruchsvolle Unternehmenskunden führen vor dem Kauf in mehreren Bereichen eine sorgfältige Prüfung des potenziellen Anbieters durch.

Preise und Auszeichnungen

Sehen Sie sich an, welche Preise und Auszeichnungen das Unternehmen eventuell bereits bekommen hat. Preise und Auszeichnungen sind eine Bestätigung seitens der Wirtschaft und der Branche, dass das Unternehmen und seine Produkte ihre Versprechen erfüllen. Wie die Unternehmensqualität variiert auch die Qualität der Auszeichnungen.

Zusätzliche Services

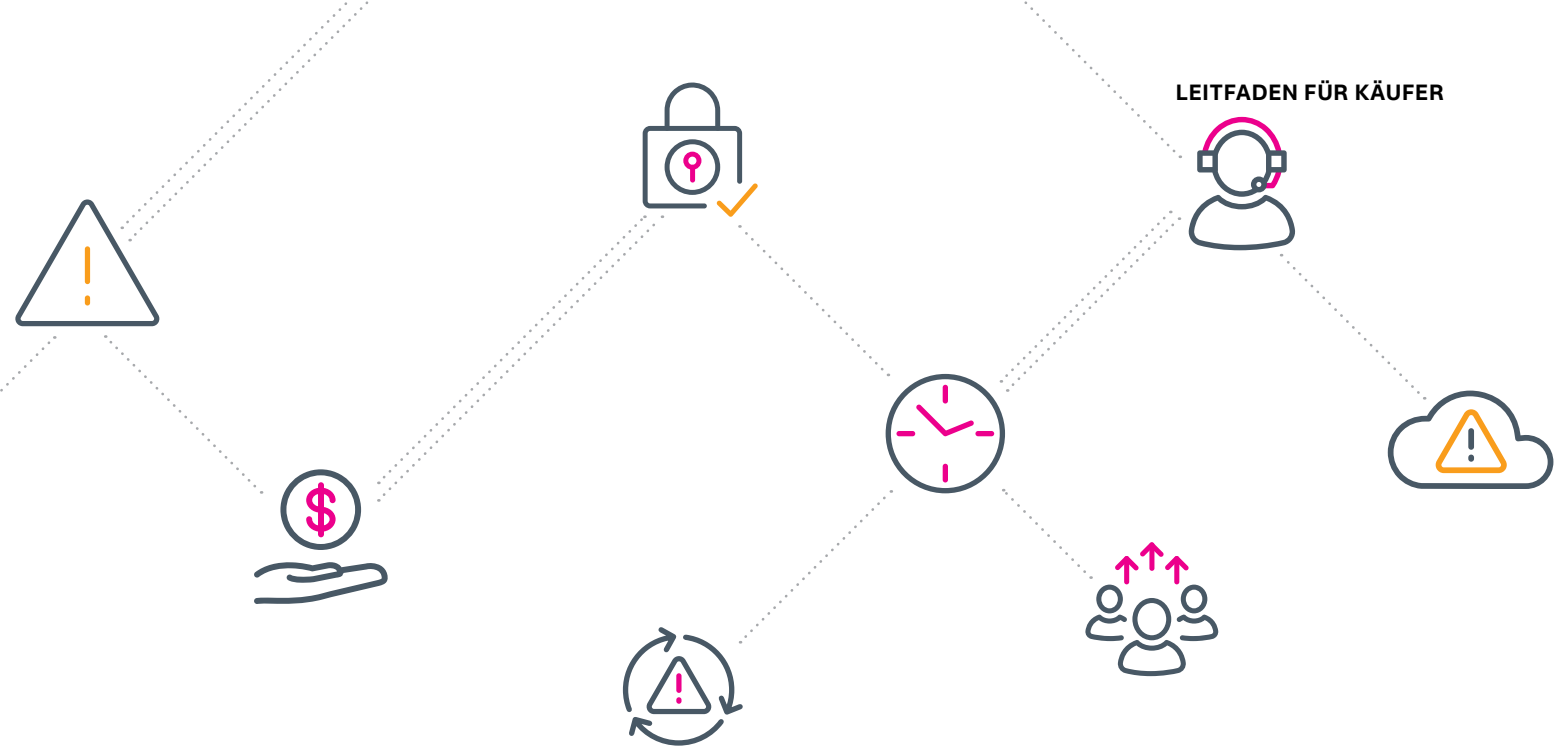
Die zusätzlichen Services, die ein Unternehmen für seine Technologie anbietet, können sich stark auf die Breitstellungserfahrung und letztendlich auf den Projekterfolg des Käuferunternehmens auswirken.

Professionelle Services

Die Reifegrade der Sicherheitsabläufe können von Unternehmen zu Unternehmen stark schwanken. Daher ist es wichtig zu prüfen, ob das Unternehmen professionelle Services anbietet, die die Chancen auf eine erfolgreiche Bereitstellung erhöhen. Es ist auch wichtig, dass Experten für die Servicebereitstellung zur Verfügung stehen, um Prozesse aufzubauen (falls nicht vorhanden) und manuelle Workflows in Automatisierungs-Playbooks umzuwandeln.

Unterstützung nach dem Kauf

Viele Start-up-Unternehmen bieten exzellente Technologie und Unterstützung vor dem Kauf, lassen dann aber beim Support nach dem Kauf stark nach. Prüfen Sie die Bandbreite der Support-Optionen und stellen Sie fest, ob das Unternehmen die Art von Unterstützung bietet, die Sie benötigen.



3. Fazit

In diesem Leitfaden werden viele Kriterien beschrieben, die bei der Bewertung von SOAR-Plattformen berücksichtigt werden sollten. Von Kernkomponenten über Plattformattribute bis hin zu Unternehmensaspekten: Es ist wichtig, einen kompletten Katalog mit Bewertungskriterien zu entwickeln, bevor Sie den Bewertungsprozess beginnen und eine Plattform auswählen.

Wenn Sie mehr über die Splunk SOAR-Plattform für Sicherheitsautomatisierung und -orchestrierung erfahren möchten, **laden Sie sich die kostenlose** Splunk SOAR Community Edition herunter oder **kontaktieren Sie den Vertrieb**.

21-13321-SPLK-SOAR Buyers Guide-108-WP_DE

Checkliste zur Bewertung

Nutzen Sie diese praktische Checkliste zur Bewertung verschiedener SOAR-Plattformen.

Name der Plattform: _____

Kernfunktionalität

- ☐ **Orchestrator**
  - ☐ Datenintegration
  - ☐ Entscheidungsmöglichkeit
  - ☐ Ausführung von Aufgaben
  - ☐ Überwachung durch Menschen
  - ☐ Datenmanagement
  - ☐ Fehlertoleranz
- ☐ **Automatisierungs-Engine**
  - ☐ Skalierbarkeit
  - ☐ Erweiterbarkeit
- ☐ **Benachrichtigungsverwaltung**
  - ☐ Benachrichtigungsdetails
  - ☐ Auslösen von Aktionen
  - ☐ Aktionsergebnisse
  - ☐ Aktivitätsprotokoll
  - ☐ Benachrichtigungsstatus, -schwergrad und -sensitivität
  - ☐ Zusammenarbeit an Benachrichtigungen
- ☐ **Ticket-Verwaltung**
  - ☐ Organisation von Ticketdaten
  - ☐ Hinzufügen von Daten zu einem Ticket
  - ☐ Verknüpfen von Tickets mit Benachrichtigungen
  - ☐ Zuordnung zu bestehenden Prozessen
  - ☐ Aktivitäten-Auditing
- ☐ **Playbook-Verwaltung**
  - ☐ Playbook-Organisation
  - ☐ Bulk Edits von Playbooks
  - ☐ Revisionskontrolle und Verteilung
- ☐ **Automatisierungseditor**
  - ☐ Benutzeroberflächenelemente
  - ☐ Blockbasierte Codedarstellung
  - ☐ Einbinden von Menschen in den Entscheidungsprozess
  - ☐ Informationsaustausch mit Aktionsergebnissen
  - ☐ Zugriff auf Playbook-Quellcode
  - ☐ Gleichzeitige visuelle und nicht-visuelle Playbook-Erstellung
  - ☐ Integriertes Testing und Debugging sowie Laufzeitprotokollierung
  - ☐ Sicherer Modus
- ☐ **App-Framework**
  - ☐ Offenes Ökosystem
- ☐ **Metriken und Berichte**
  - ☐ Flexible Dashboards
  - ☐ Leistungsberichte
  - ☐ Bericht zur Sicherheitseffektivität
  - ☐ App-Integration und Playbook-Performance
  - ☐ Menschliche Arbeitslast
- ☐ **Bereitstellungsoptionen**
  - ☐ Lokal
  - ☐ Cloud
  - ☐ Hybrid

Plattformattribute

- ☐ **Unterstützung durch die Community**
  - ☐ Große und aktive Community
- ☐ **Zusammenarbeit**
  - ☐ Zusammenarbeit über die Community hinweg
  - ☐ Zusammenarbeit über die Plattform hinweg
- ☐ **Kognitiv**
  - ☐ Wählbare Automatisierung
- ☐ **Sicher**
  - ☐ Skalierbar
  - ☐ Offen und erweiterbar
  - ☐ Offenes Integrations-Framework
  - ☐ Keine Schnittstelleneinschränkungen
  - ☐ Benutzerfreundlich
  - ☐ Installation und Einrichtung
  - ☐ Einarbeitung
  - ☐ Verkürzen der Zeit bis zur Automatisierung

Unternehmensaspekte

- ☐ **Eigenschaften des Unternehmens**
- ☐ **Firmengeschichte**
- ☐ **Ausführungsfähigkeit**
- ☐ **Kundenstamm**
- ☐ **Preise und Auszeichnungen**
- ☐ **Zusätzliche Services**
- ☐ **Professionelle Services**
- ☐ **Unterstützung nach dem Kauf**