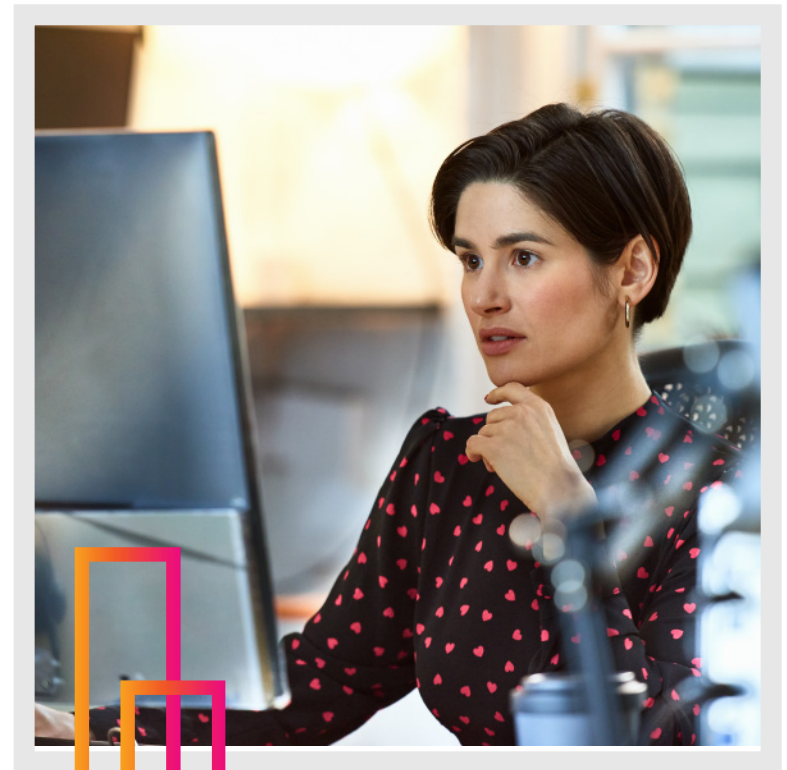
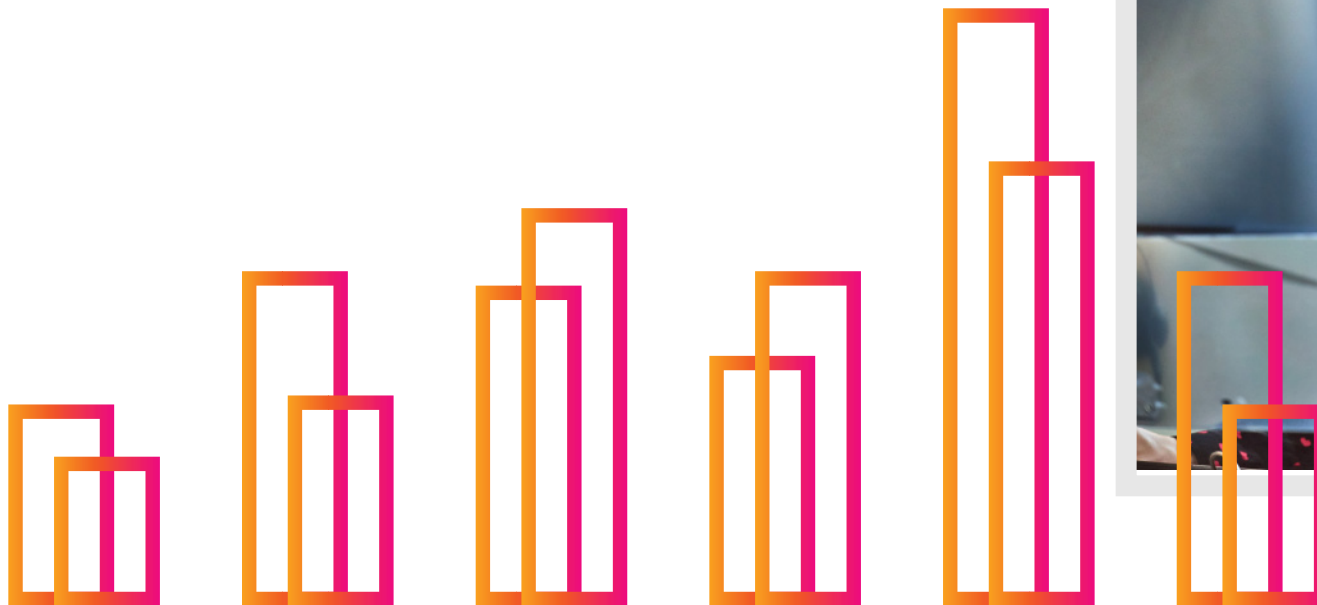
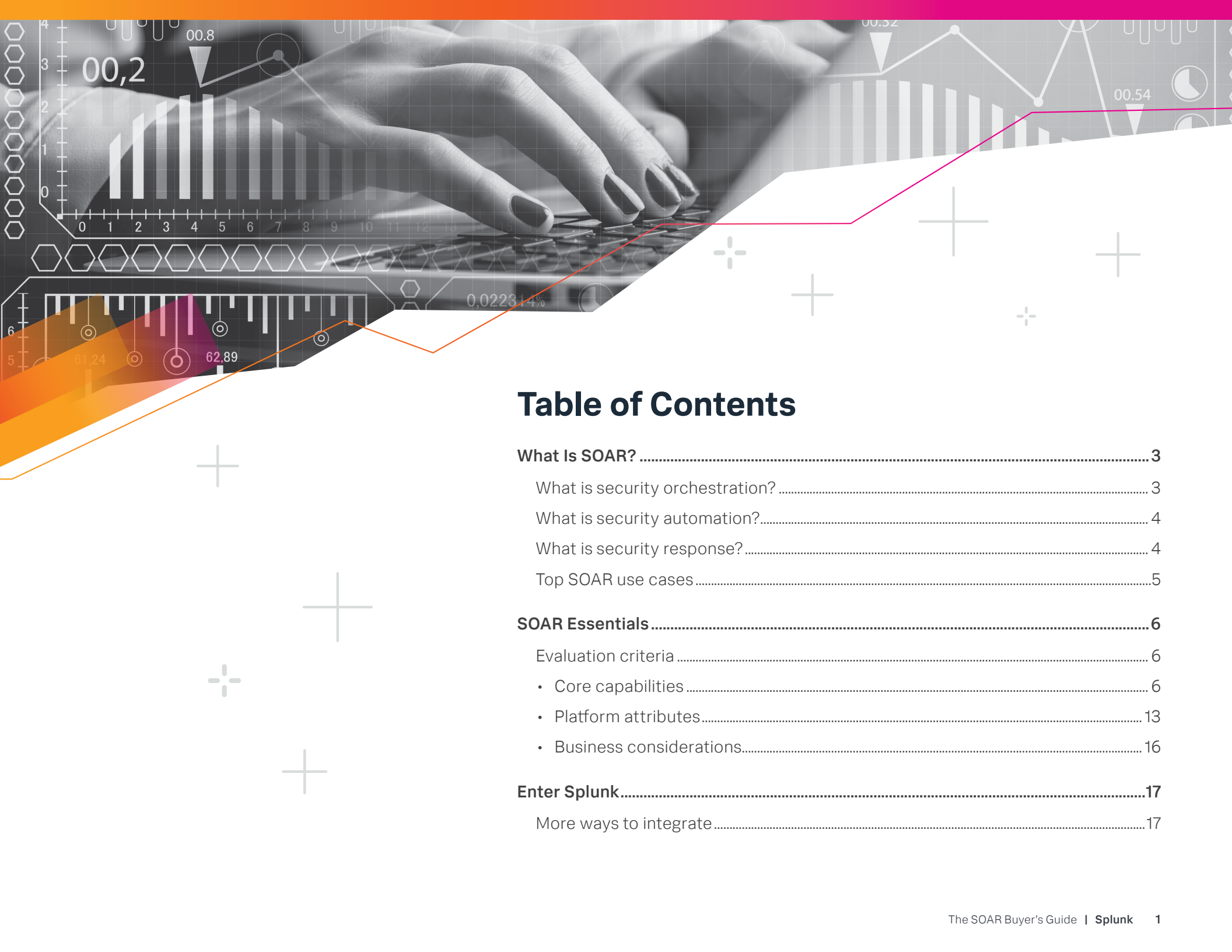


# The SOAR Buyer's Guide

The who, what, where, when and why  
of buying a security orchestration,  
automation and response solution




**splunk**>  
turn data into doing®



## Table of Contents

<b>What Is SOAR?</b> .....	<b>3</b>
What is security orchestration? .....	3
What is security automation? .....	4
What is security response? .....	4
Top SOAR use cases .....	5
<b>SOAR Essentials</b> .....	<b>6</b>
Evaluation criteria .....	6
• Core capabilities .....	6
• Platform attributes .....	13
• Business considerations .....	16
<b>Enter Splunk</b> .....	<b>17</b>
More ways to integrate .....	17



There's never been a better time to invest in a security orchestration, automation and response (SOAR) solution. Gone are the days where security teams had to manually respond to incidents; now, security teams can work smarter — not harder — by automating repetitive tasks, increasing analyst productivity and accuracy, and better protecting the business.

All too often, security teams find themselves plagued by analyst grunt work. Security operations work is rife with monotonous, routine and repetitive tasks, especially at the Tier-1 analyst level. There's also a shortage of over one million cybersecurity professionals with the necessary knowledge and expertise to staff security operations centers (SOCs) around the world.

Other common challenges include (but are certainly not limited to):

- **Too many alerts:** Analysts are overwhelmed by hundreds (if not thousands) of security alerts. The sheer volume can quickly overwhelm a security team — increasing security incident backlogs and leading to the much-dreaded “alert fatigue.”
- **Too many siloed point-products:** Teams are expected to juggle disconnected security tools, consisting of static, independent controls and zero interoperability. When tools don't work together, security gaps are inevitable, and attackers can (and will) exploit them.
- **The skills gap:** Staffing a SOC is no easy task. Qualified analysts are in short supply, and turnover is extremely high because of an increasingly competitive marketplace. Unsurprisingly, time and resources spent on analyst training and establishing institutional knowledge are often lost in the shuffle.
- **Lack of process:** Most security teams fail to establish workflows and standard operating procedures (SOPs) for different types of security events. Without this operational rigor, analysts are unable to act quickly and decisively when responding to an attack.

- **Lack of speed:** Attackers have ample opportunity to breach and exfiltrate data when the mean time to detect (MTTD) is too long. A typical human response time to an alert can take anywhere from minutes (best case scenario) to weeks or months (sometimes even longer). The latter — and sometimes even the former — is usually too long a dwell time for serious threats.

In the face of all this, security teams have an increasingly hard time identifying and responding to threats. Organizations need a solution that's powerful, flexible and fast — a solution powered by automation.

With the help of SOAR, analysts can respond to any threat that comes their way, no matter how big or small. A robust SOAR solution can execute a series of actions — from detonating files to quarantining devices — across an organization's security infrastructure in mere seconds (versus hours or days, if performed manually) by codifying workflows into automated playbooks.

Bottom line? You too can reap the rewards of SOAR. “The SOAR Buyer's Guide” will help you make sense of key criteria for evaluating your options — that way you can make the choice that's best for you and your organization's security operations — freeing up your analysts for more worthy causes (and a longer lunch break).



# What Is SOAR?

A SOAR solution clears out mundane tasks that would normally tie up a security team's time and resources. Thanks to SOAR, security teams can field more incidents, investigate issues closely, save time on critical security tasks, and improve an organization's overall security posture.

While automation is a standard practice across most industries, cybersecurity is arguably late to the game. But in recent years there's been a distinct shift, with practitioner interest only continuing to grow, along with the number of vendors entering the SOAR category — reimagining their existing security offerings from adjacent market segments.

However, because of how newer vendors have positioned themselves, market definitions around SOAR have become blurred, making comparisons difficult. To lend some much-needed clarity, here's our breakdown of the following categories:

## Security orchestration



Security orchestration is the machine-based coordination of a series of interdependent security actions across a complex infrastructure.

## Security automation



Security automation is the machine-based execution of security actions.

## Security response



Security response is the policy-based coordination of human and machine-based activities for event, case and incident workflows.

## What is security orchestration?

Security orchestration is the machine-based coordination of a series of security actions across a complex IT ecosystem. This helps everything (i.e., a wide range of independent security tools) work in concert with one another, while automating tasks across products and workflows. Basically, orchestration allows security teams to automate complex processes across disparate point products, maximizing the value from security staff, processes and tools.

### Security orchestration can:

- Collectively and automatically coordinate workflows across tools.
- Provide context around security incidents by aggregating data from different sources.
- Allow for deeper, more meaningful investigations.





## What is security automation?

Security automation is the machine-based execution of security actions with the power to programmatically investigate, respond and remediate threats without the need for human intervention. Security automation does most of the work for analysts, so they no longer have to weed through and manually address every alert as it comes in, or manually process every security action or task.

### Security automation can:

- Investigate threats in your environment.
- Triage potential threats by following the steps, instructions and decision-making workflows taken by security analysts to investigate the event and determine whether it's a legitimate incident.
- Decide whether to take action on the incident.
- Contain and resolve the issue.
- Automate vulnerability investigation and patching.

## What is security response?

Security response is the policy-based coordination of machine-based automated actions and human-based input for event, case and incident workflows. The technical details of a security event or alert should be organized in a way that allows an analyst to quickly digest the information at hand, so they can better understand the entire scope of the security scenario and respond accordingly. In short, a security analyst should be able to seamlessly issue investigative, containment or response actions against the data provided.

Once alerts or events are confirmed and escalated, a case management component should take over and drive a broader, cross-functional lifecycle from creation to resolution.

### Security response can:

- Confirm multiple events and escalate them into a single case.
- Seamlessly map incidents to an organization's existing processes.
- Issue investigative, containment or response actions against certain technical data.
- Provide an activity log that displays a record of all actions executed against an event or alert.
- Drive a broader, cross-functional security lifecycle from creation to resolution.



## Top SOAR security scenarios

The following use cases are modeled after existing manual workflows, and highlight common operational pain points. These workflows usually contain countless manual tasks that require coordination across different point products.

Before beginning your evaluation, you'll need to map out potential use cases specific to your organization. Ideally, this would include input from stakeholders across your security operations, as well as leadership. Identifying these key use cases — even if they aren't implemented right away — is critical to a successful security strategy.

Below is a selection of security use cases spanning investigation, enrichment, containment and remediation:

<b>Alert Triage</b>	Alert triage validates and prioritizes inbound alerts and contextualizes events. This includes certain methodologies and models to eliminate false-positive alerts from further processing.
<b>Incident Response</b>	Incident response will vary depending on the type of incident involved. For example, responding to a phishing attempt is a completely different effort than responding to a successful ransomware attack.
<b>Indicator of Compromise (IOC) Hunting</b>	By automating IOC hunting, teams can tap into the latest threat intelligence without exhausting their resources. They can also implement intelligence scoring to determine which threat intelligence sources they should be looking at.
<b>Vulnerability Management</b>	Automating (and subsequently, standardizing) the cycle of identifying, classifying, remediating and mitigating vulnerabilities will yield greater efficiency and consistency.
<b>Network Access Control (NAC)</b>	SOAR can augment dynamic access control strategies. One example is integrating a detection system that wasn't previously included in NAC decision-making.
<b>User Management</b>	User management ensures that specific accounts are enabled and disabled quickly and systematically to eliminate insider threats, account takeovers or credential abuse.
<b>Penetration Testing</b>	Activities — like asset discovery, classification and target prioritization — are automated, increasing the productivity of the pen testing team.
<b>Intelligence Sharing</b>	Organizations with intelligence sharing initiatives can benefit from an automation-assisted playbook. Automation can also increase an analyst's productivity and provide time-sensitive information much faster than manual processes.

Additional SOAR-specific use cases can stem from a host of other challenges, where security teams codify criteria for detection and automation. Be sure to check out our e-book, [Five Automation Use Cases for Splunk SOAR](#), for additional examples.

# SOAR Essentials



## Evaluation criteria

Our evaluation criteria for a SOAR solution is organized into three essential categories: **core capabilities**, **platform attributes** and **business considerations**.

### Core capabilities

Core capabilities are the fundamentals (or basic parts) of a SOAR solution. We'll cover each capability and component, as well as key considerations for evaluating your options.

#### Orchestrator

- **Data ingestion**

Security data needs to be ingested. An orchestrator can ingest and compile data from any source, in any format, while keeping it logically separated. If the data is unstructured, the user should be able to apply a data handler to interpret the data and make it accessible.

- **Decision-making**

Users should be able to apply automation playbooks to their data sources. For example, an email phishing playbook can be applied to an email-based ingestion source, while a malware investigation playbook can be applied to a security incident and event management (SIEM)-alert source.

- **Task execution**

Dispatch automated tasks at the appropriate and optimal time, passing them onto the automation engine for execution.

- **Human supervision**

Balance machine-based automation with the necessary human supervision. There are usually three scenarios where an analyst is required — 1.) When approval by the asset's owner is required to execute a security action on a target; 2.) When review by an analyst is required to ensure that security is balanced with business continuity; and 3.) When an analyst needs to augment codified decision-making logic (e.g., when an error occurs).

- **Data management**

Ensure that the output of data from one action is properly parsed, normalized and structured so that future actions can make use of it. The orchestrator should also support caching relevant data to avoid taxing other resources.

- **Fault tolerance**

SOAR regularly interacts with many discrete products and services; however, availability isn't always guaranteed. Access to external services can be interrupted and broken — in which case, an orchestrator should perform predictably, recovering and resuming operation seamlessly as configured.

### Automation engine

The automation engine is the workhorse of most SOAR solutions, receiving actions (or tasks) from the orchestrator, and then responding to them accordingly. Because automation runs independent of human interaction, criteria like platform scalability and extensibility are important to consider.

- **Scalability**

Additional use cases are added and automated over time. To account for the growing processing load, the automation engine should be able to scale vertically and horizontally.

- **Extensibility**

Security evolves quickly and new functions should be supported without major re-engineering. The automation engine should support the flexibility to adapt to the unique capabilities of its environment.



## Alert management

After your data is ingested, inbound alerts will be queued up and prioritized. Investigations are then performed using manual or automated actions to yield the highest level of productivity and accuracy.

To surface the right information at the right time, the interface should arrange and triage alerts in an easily digestible format. That way, analysts can avoid extensive searches or switching between contexts, and can quickly make sense of notable events.

- **Alert details**

The details of a security alert should be organized in a way that allows an analyst to quickly digest and understand the security event. This includes an organized view of relevant technical data, including IP addresses, domain names, file hashes, user names, email addresses and other data fields. Use of a standard format like “common event format” (CEF) or an equivalent is highly beneficial for data exchange.

- **Issuing actions**

A security analyst should be able to issue manual actions when investigating, containing or correcting an alert, and the interface should allow a user to execute an action by selecting the data to operate on. An analyst should also be able to issue an automated collection of actions against an alert, which can be referred to as a “playbook.”

- **Action results**

Action results should be available in a summary format (e.g., a table view) as well as in a more comprehensive format (e.g., JavaScript Object Notation or JSON, a common data format), so that they’re readily available and easy to view.

- **Activity log**

A comprehensive activity log displays a record of all actions executed against an alert — whether they were initiated manually or via an automation playbook. Each action should display its results, including an indicator of success or failure, making it clear whether the action was fully executed.

- **Alert status, severity and sensitivity**

Alerts should have a status indicator (e.g., “new,” “open” or “closed”), a severity indicator and a sensitivity indicator (e.g., “traffic light protocol” or TLP designations). Each indicator should be modifiable within the alert management interface, as well as from within a playbook.

- **Alert collaboration**

The interface should provide an area where analysts can collaborate, comment and provide information about an alert, and all its relevant or miscellaneous data.





## Case management

Case management takes a broader, cross-functional view of an incident's life cycle — from creation to resolution. Multiple alerts and/or events can be confirmed, aggregated and escalated as a single case. While alert management is usually technical and singular in its focus, case management can also incorporate non-technical steps into the process.

Also, overall case volume is usually much lower than alerts, with numbers typically in the single digits.

- **Case data organization**

All data relating to a specific case should be aggregated by the case management component. Displaying this information in a single location helps users digest everything without context switching.

- **Adding data to a case**

Relevant technical data should be attached to the case in question (e.g., source data, action results). Relevant non-technical data (e.g., notes, memos, emails, screenshots, recordings or any other arbitrary file with relevance) should also be included.

- **Linking cases to alerts**

Ideally, the case management interface should link to the alert management interface for each respective alert. This is especially handy if and when an analyst determines a piece of data requires further investigation or a containment action needs to be taken.

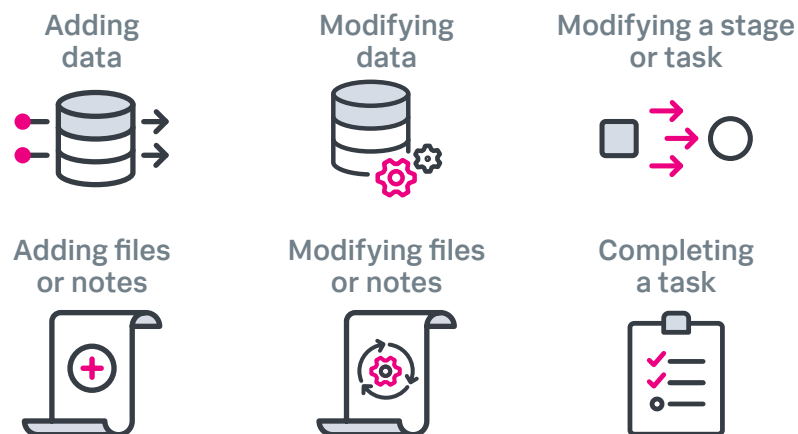
- **Mapping to existing processes**

Most organizations have standard operating procedures for incident response, emergency, disaster and other critical situations. Case management should let the user define their processes and workflows — multiple stages where each stage has one or more tasks, and each task can be assigned an owner — and then save them as a template.

- **Activity auditing**

New or updated information, including status updates, should be logged in an audit trail and easily exportable.

Changes to a case might include:





## Playbook management

Playbook management helps with the implementation and maintenance of standard operating procedures across an organization (and sometimes beyond). Ideally, this component has revision/version control and syndication management.

- **Playbook organization**

Analysts should be able to customize categories around different playbook groups. Groupings would be based on what works best — or what is most applicable — to the organization (e.g., sensitivity, organizational segments, asset types, themes).

- **Custom functions**

Beyond what's available out-of-the-box (OOTB), users should be able to write custom code and/or functions. These functions should be shareable across multiple playbooks, while providing centralized code management and version control.

- **Revision control and distribution**

Integration with a version control system (VCS) is highly recommended for successful playbook management. At the deployment level, a VCS helps with the systematic distribution of playbooks. At the development level, a VCS is important for tracking changes and having the option to roll back updates if necessary.

- **Bulk edits to playbooks**

The inner workings of each playbook are likely to be unique. However, there are commonalities between many playbooks at the administrative level.

A playbook management system should allow for the bulk editing of playbooks, including:

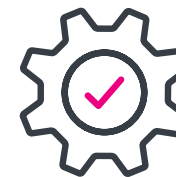
### Ingestion sources



### Enabling/disabling enhanced logging



### Enabling/disabling automatic execution enabling/disabling safe mode operation



### Setting playbook category grouping





## Automation editor

Analysts can codify processes into a playbook via an automation editor. Basic source code editors make this a difficult task; however, a visual automation editor allows all security experts — regardless of their programming experience — to write playbooks at the source code level, and to construct comprehensive and sophisticated playbooks.

The visual editor should adhere to Business Process Modeling Notation (BPMN) standards — a graphical notation for specifying business processes. BPMN supports intuitive symbols for business users, while providing technical users with different ways to represent highly complex processes.

- **User interface elements**

The user interface should start with a canvas where visual playbooks can be constructed. This part of the interface should provide an area where a desired action can be specified (for example `block_ip` or `file_reputation`). Once an action is selected, parameters will be required to configure the action (which can be manually entered or selected from a list). The interface should also have a place for testing and debugging, with a seamless transition between edit and test mode, along with a source code view.

- **Block-based representation of code**

Using blocks to represent meaningful steps within an automation platform allows users to write comprehensive, complex playbooks without touching the underlying source code. Blocks should be connected in a one-to-one, one-to-many or many-to-one fashion to dictate an order of execution.

- **Inserting humans into the decision process**

Supervised automation is a common requirement. This is where a human can be inserted into an automation sequence to approve, review or augment continued playbook execution. A playbook author should have the ability to specify who should be looped in, along with the type of notification or level of approval desired, as well as the type of error to be alerted on in the event that one or more services are unavailable.

- **Information exchange of action results**

The interface for the automation editor should allow for new information to be available as inputs, parameters, downstream actions, decision blocks, etc. The results of preceding actions should be accessible visually and selectable from a drop-down menu when populating the parameters of an upstream action.

- **Access to playbook source code**

While constructing the playbook in a visual editor, the playbook source code should be generated in real time and accessible to the author. Some users may prefer to draft all (or part) of the playbook via a traditional source code method, which can be viewed in place of a visual editor. Switching between visual and source code modes should be seamless.

- **Simultaneous visual and non-visual playbook construction**

When working with a playbook's source code, the automation editor should allow the author to modify the playbook at the source code level and have the ability to modify the playbook at the visual block level. At times, the author may require individual blocks (like actions and decision blocks) to be modified at the source code level for customizations beyond the scope of the visual editor. When these modifications are made, a user should still be able to modify the playbook visually.

- **Built-in testing and debugging and runtime logging**

It's the industry standard for integrated development environments (IDEs) to provide execution and debug capabilities. When it comes to an automation editor, a user should be able to execute playbooks against a security alert, and then observe the execution activity and results. The goal is to enable the author to quickly edit, test and debug playbooks within one interface.

- **Safe mode**

An automation editor should also provide a safe mode for new playbooks that need pre-production testing. This mode simulates the execution of automation targets without exacting change on them.

## App framework

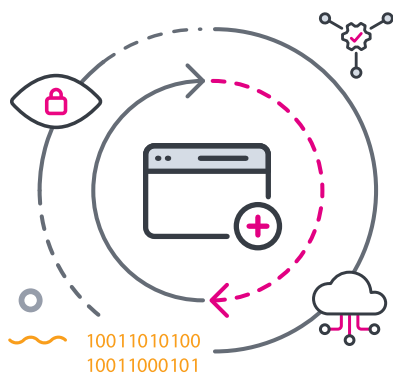
The app framework offers an extensible interface for new integrations, connecting the platform to any of the thousands of point products available on the market today.

- **Open ecosystem**

A SOAR solution can lose its value over time if it fails to integrate with new or popular offerings on the market. To support these types of integrations, a SOAR solution should adopt an open ecosystem to promote app development. New technologies must also be quickly integrated without requiring any modification to the core solution.

- **App development**

App development is a key component to an open ecosystem, since it allows users to integrate with multiple technologies in support of their playbooks. A SOAR solution should be able to streamline app development within the product itself, so that users can view, test, extend and edit existing apps, as well as create entirely new apps, all from the user interface.



## Metrics and reporting

Metrics and reporting are necessary for understanding and quantifying pretty much anything, and a SOAR solution is no exception. While automation promises increased performance and productivity, metrics are the way to gauge its respective effectiveness, and to also identify where improvements can be made.

- **Flexible dashboards**

The metrics for success vary; they're usually specific to the organization or the individual, and can rely on many factors. That's why users need to be able to organize their key performance indicators (KPIs) in a way that makes the most sense to their organization. A SOAR solution should allow for this data to be customized and organized accordingly.

- **Performance reporting**

Efficiency is usually the main driver behind automation. Understanding the quantitative performance gain and resource savings is key to justifying your investment.

Examples of these metrics that should be reported on:

- Mean time to resolve (MTTR).
- Mean dwell time (MDT), which is defined as the period of time between a compromise by a threat actor and taking an appropriate response.
- Analyst hours saved through automated execution.
- Number of full time equivalents (FTEs) gained through automated execution.
- Average time saved per playbook run.
- Money saved (FTE-cost x FTEs-gained).



### • Security effectiveness reporting

Automation should also increase security effectiveness and the overall security posture of the organization. Understanding the total number of security alerts managed, along with the rate at which they're being managed, is another important justification.

Examples of these metrics that should be reported on:

- MTTR and MDT.
- Total number of open alerts.
- Alerts opened daily/hourly/weekly/monthly.
- Alerts closed daily/hourly/weekly/monthly.
- Performance against service level agreements (SLAs).

### • App integration and playbook performance

Understanding the most frequently invoked playbooks can help shed light on where further investments can be made. Ideally, playbook design should strive for the automated closure of false positives or high-confidence true positive alerts.

To identify gaps in automation, as well as the effectiveness of tool integrations, the following metrics should be reported on:

- Alerts closed through automation (per hour, day, week, month or other timeframe).
- Most active app integrations.
- Most active actions (manual and automated).
- Most active automated playbooks.
- Playbook execution time.
- Action execution time.



### • Human workload

While automation is intended to close the human resource gap, there are still plenty of situations where an analyst needs to be involved in the day-to-day operations of a SOAR solution. These cases include manual triage and when other actions are required on an alert, or when human approvals are inserted into the playbook to achieve “supervised automation.”

The following example metrics should be provided to understand the human workload involved in the automation process:

- Alerts assigned to an individual.
- Alerts closed by an individual.
- Average approval time.
- Number of outstanding approvals.
- Approvals required (per hour, day, week, month, or other time window).



## Platform attributes

Platform attributes may be more qualitative in nature. Considering this, the following criteria are evaluated more often through observation and interaction with the platform.

### Deployment options

A SOAR solution should support on-premises, cloud or hybrid deployments. While organizations may prefer on-prem, others will invariably prefer a cloud-based solution. The type of delivery or deployment you decide on will largely depend on the needs of your organization — like budget, storage and security requirements, as well as how to best streamline security operations, and facilitate digital transformation within the scope of your existing framework.

### Community-powered

Ideally, a SOAR solution should support a community model by adopting an open ecosystem for app development. This helps promote long-term success by avoiding vendor lock-in, and technologies can easily transition without negatively impacting automated playbooks. The ever-evolving nature of security also fuels the need for professionals to work together to share playbooks, best practices and strategies for dealing with the latest threats.

#### A large and active community

Most users prefer to draw on the experiences of other like-minded users. A large-and-active user community provides the opportunity to share playbooks and apps, or brainstorm ideas for new automation use cases. To facilitate an exchange of ideas, connecting users within the community is crucial. Messaging/communication tools in particular are an effective means for technical and design support, providing answers to questions and brainstorming on automation use cases.

## Collaborative

Collaboration improves feature completeness, application integration and automated playbooks that address an evolving range of scenarios.

- **Collaboration across the community**

From a content perspective, user and vendor content should be accessible from a centrally located repository. This includes technical contributions, such as playbooks and app integrations, as well as non-technical contributions such as presentations, tech notes, blogs and other documentation methods.

- **Collaboration across the platform**

A SOAR solution should help users collaborate across varying circles of trust. The solution should support collaboration of sensitive information across privileged groups across the organization's security team.





### Cognitive

A cognitive SOAR solution applies knowledge from humans along with previous observations to guide future decisions. This is codified into a system in the form of playbooks. This methodology is based on execution statistics, characteristics of ingested data and action results.

This information can be used to recommend individual actions, playbooks or a set of actions in a sequence that would form a playbook. It's important to understand the current cognitive abilities of a SOAR solution, as well as the cognitive strategy and roadmap for future iterations.

### Dialable automation

Teams usually adopt automation use cases one at a time, slowly building up trust in the system. To help this along, a SOAR solution should support a set of features that allow selective human interaction with the automated playbook. Inserting humans into a workflow should be possible on a per-asset (point security tool or technology) or per-action basis.

The former (per-asset) should notify the asset administrator each time an action is executed on that asset. The latter (per-action) should insert a prompt at any point in an automated playbook. The prompt gives the user an option to continue, pause or abort the request. This level of supervision allows users to gain confidence with the programmed steps.

### Secure

Unsurprisingly, one of the most important aspects of a security automation and orchestration solution is security. A SOAR solution holds authentication credentials and other highly-sensitive information — encrypting sensitive information and supporting a robust role-based access control facility.

Security best practices of a SOAR solution include:

#### Encrypted security credentials



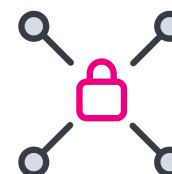
#### Support for authentication management systems



#### Make sure credentials are not stored in memory



#### Support multi-factor authentication







### Scalable

A SOAR solution needs to be able to scale vertically and horizontally. As an organization adds use cases over time, there will be an even greater processing load to consider. The solution should be designed in a way that allows for vertical scaling by increasing hardware resources (for example CPU and RAM) as well as horizontal scaling by increasing the number of server instances supporting the deployment.

### Open and extensible

Security is always evolving, evidenced by the multitude of point products available today. A SOAR solution should be designed for openness and extensibility. It should easily support new security scenarios, new products, new actions and new playbooks.

### Open integration framework

The net effect of an open integration framework is that technologies should be able to transition in and out of the platform without negatively impacting automated operations. Users should also have the option to develop additional integrations without relying on the SOAR vendor.

Good examples of where this applies is homegrown applications, a custom or early access API from a vendor, or if they choose to extend the functionality of the automation platform. This open framework should follow a common standard and programming model. There should also be an abundance of documentation and examples available.

### No interface restrictions

Some technologies expose interfaces using REST APIs, SSH, syslog, custom API, or some other protocols or methods. An extensible integration framework should not enforce restrictions on interface types. If there is connectivity to the point product or application from the automation platform, the method of interface should not impact the app integration — allowing any interface method to be used.

### Mobile

A SOAR solution is designed to accelerate response times — in other words, reduce dwell time and mean time to resolve. Rapid response means security analysts need to be reachable when a case or security prompt requires human intervention. But analysts are not always sitting at their desk with their laptop open, ready to answer prompts at a moment's notice.

That's why it's important for a SOAR solution to offer access, interactivity and control of the platform from the convenience of the analyst's mobile device. This way, analysts can run playbooks on the go, review security artifacts and triage events without opening a laptop, respond to prompts from the palm of their hand, and always be reachable whether they're sitting at their desk or on the go.

### Ease-of-use

Though enterprise software is very rarely simple, it's possible to reduce the friction in deploying and using a SOAR solution.

- **Installation and setup**

Virtual appliance form factor makes deployment simple, as most organizations already leverage virtualization with other infrastructure.

- **Onboarding**

A SOAR solution can greatly help overcome an initial learning curve by using an onboarding process to help a user configure system settings, connect to a data source and activate their first few playbooks.

- **Accelerate the time-to-automate**

A SOAR solution should help users to get started with automation quickly. This is achieved by supplying a robust set of automated playbooks out of the box. Empowering users to quickly draft, test and deploy automated playbooks is another significant accelerator.

## Business considerations

No matter how great a company's core technology is, there are considerations outside of what is traditionally thought of as the product which can heavily influence a buyer's decision-making. One major consideration is the attributes of the company marketing the offer. Another consideration is the set of services offered by the company that augments the core technology to form the whole product that the buyer ultimately experiences.

### Company attributes

When making a decision about procurement, it's important to consider the profile, quality and future potential of the company you choose. The reality is that many new vendors with new solutions will fail. You should choose a company that has the strength to deliver on the promises they make.

### Company history

The vendor you select should have plenty of experience in developing security solutions. While security orchestration, automation and response is a relatively new segment of the market, its origins can be traced back many years. It's important to understand how the company was formed and how they decided to pursue the SOAR segment.

### Ability to execute

You should look for a company that is supported by a seasoned team of experienced professionals. Predicting a company's ability to execute is often directly linked to the track record of team members.

### Customer base

The quality and profile of a company's customer base is a reflection on the company itself. Sophisticated enterprise customers perform rigorous diligence on a potential vendor in several areas prior to making a purchase.

## Awards and recognition

Look at the company's awards and other types of recognition they've received. These are endorsements that prove the vendor and its products live up to their claims. Like the companies themselves, the quality of the awards vary as well.

### Ancillary services

The auxiliary services that a company offers for their technology can greatly influence an organization's deployment and the success of a project.

### Professional services

Maturity levels across security operations can vary greatly from one organization to the next. It's important to consider whether the company provides professional services that increase the chances of a successful deployment. It's also important for subject matter experts to be available for service engagement to help build processes (if lacking) and help convert manual workflows into automation playbooks.

### Post-sales support

Many startups provide excellent technology and presales support, only to stumble when it comes to post-sales support. Examine the range of support options and determine whether the company provides the type of support you'll need.



# Enter Splunk

**Splunk can take your team from overwhelmed, to in control.**

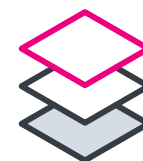
**Splunk SOAR** lets your team work smarter, respond faster and strengthen your organization's security defenses. You'll be able to automate repetitive tasks; triage security incidents faster with automated detection, investigation and response; increase productivity, efficiency and accuracy; and strengthen your defenses by connecting and coordinating complex workflows across your team and tools.

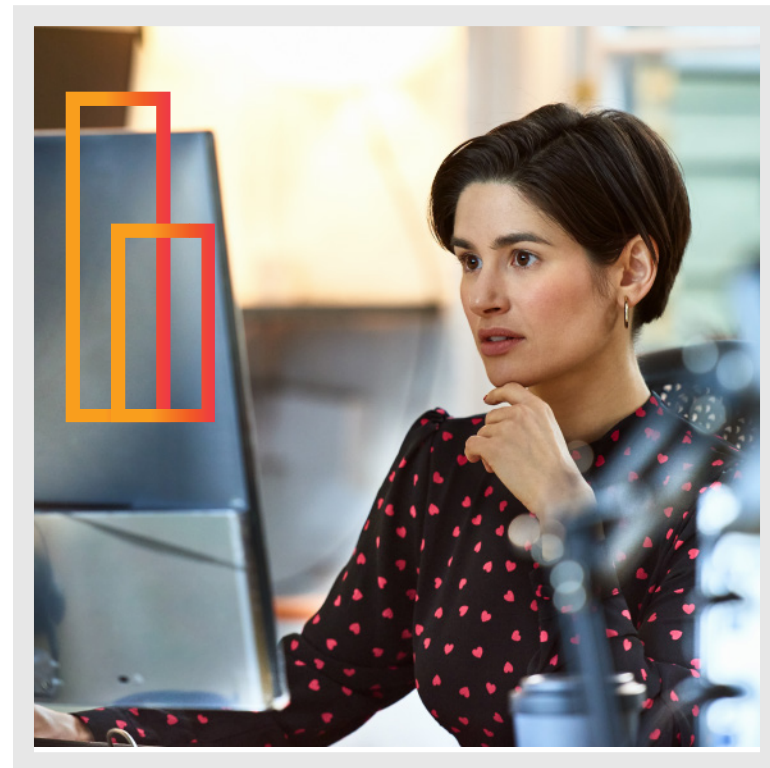
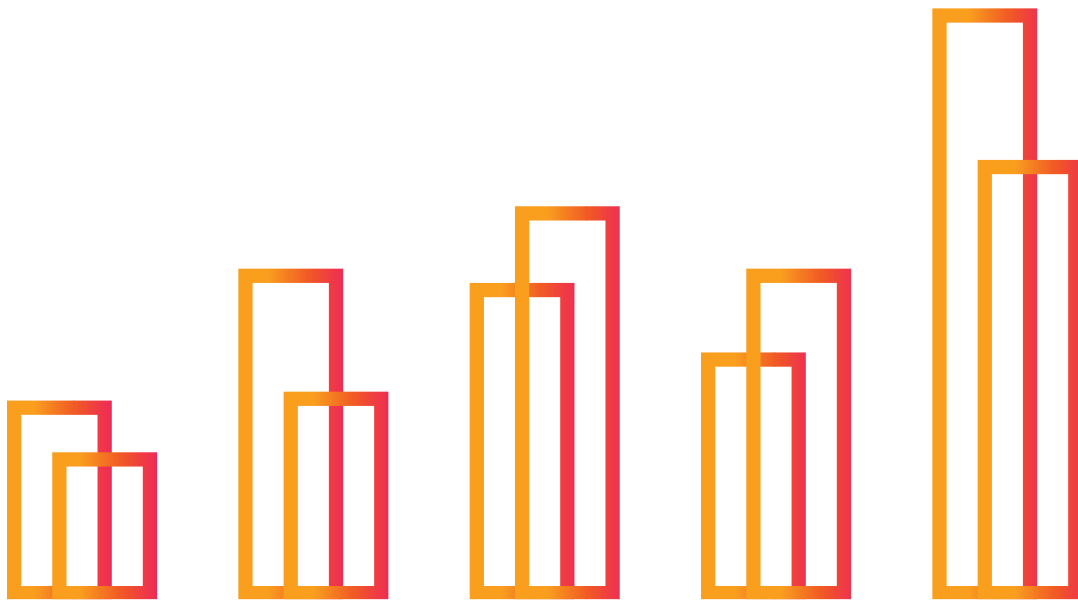
Splunk SOAR also supports a broad range of security functions including event and case management, integrated threat intelligence, collaboration tools and reporting, as well as integrating your existing security infrastructure so that each part actively participates in the defense strategy, while all working in concert.



## More ways to integrate

For more ways to integrate, **Splunkbase** offers thousands of third-party security apps to connect and integrate with Splunk SOAR. Thanks to these integrations, Splunk SOAR can direct your security tools to perform a wide array of actions — whether it's asking VirusTotal to check file reputation or Cisco Firewall to block an IP. Splunk SOAR's app model supports integration with over 350 tools and over 2,100 different actions, all available on Splunkbase. These ready-to-use apps, utilities and add-ons can help your team with security monitoring, next-generation firewall, advanced threat management and a whole lot more.





# Get Started.

To learn more about the Splunk SOAR, [download the free](#) Splunk SOAR Community Edition or [ask sales](#) for more information.

Splunk, Splunk> and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners. © 2022 Splunk Inc. All rights reserved.

22-23828-Splunk-SOAR Buyers Guide-EB-101

**splunk>**  
turn data into doing™