WHITE PAPER

The 10 Essential Capabilities of a Best-of-Breed SOAR

Work smarter and respond to threats faster using automation

splunk> turn data into doing™

Security Operations Must Evolve

Ask a group of security analysts about the challenges of working in cybersecurity, and you'll likely hear some common themes:

- A shortage of skilled cybersecurity talent
- A high volume of security alerts
- Too many security point-products to manage
- Lack of interoperability between those products
- Inability to scale security operations over time
- Increasing costs, shrinking budgets
- Increasing sophistication of malware
- Slow speed of threat detection and response

Considering these challenges, it's no surprise that security teams feel perpetually overwhelmed.

Many teams have turned to security orchestration, automation and response (SOAR) tools as a remedy.

A SOAR tool can orchestrate security actions (like investigations, triage, response) across various security products in a team's arsenal, and automate otherwise manual repetitive security tasks.

But not all SOAR tools are created equal. A best-of-breed SOAR solution will provide a set of capabilities that can completely revolutionize how you do security operations. These capabilities will allow you to:

- Work smarter by automating manual and repetitive tasks.
- Respond faster and reduce dwell time with automated detection, investigation and response.
- Strengthen defenses by integrating existing security infrastructure together so that each part is an active participant in your defense.

There are 10 essential capabilities of a best-of-breed SOAR tool that will allow you to achieve these outcomes.

Essential Capabilities of a Best-of-Breed SOAR	
Orchestration	The machine-based coordination of complex workflows across disparate security tools should increase the efficiency and speed of your security operations.
Automation	The machine-based execution of otherwise manual, interdependent security actions using "playbooks" should allow you to execute in seconds versus hours.
Event and Alert Management	An event and alert management capability in a SOAR tool should queue and prioritize inbound security events and alerts to help analysts perform triage more efficiently.
Case Management	A case management component should drive a broader, cross-functional lifecycle (from creation to resolution) of a security case.
Collaboration	Built-in chat and notes can facilitate communication across the security team, and thereby accelerate the resolution of security events.
Metrics and Reporting	Metrics and reporting are critical to understanding the effectiveness of the SOAR tool and identifying where improvements can be made to increase ROI.
Mobility	Control of the SOAR tool from the convenience of the analyst's mobile device will allow for faster response times and easy alert triage — all on-the-go.
Scalability	A SOAR tool should grow with you as your organization grows. As an organization adds more use cases over time, there will be additional processing load placed on the platform.
Open and Extensible	A SOAR tool should easily support incorporating new security scenarios, new products, new actions and new playbooks.
Community Powered	A SOAR tool must support a strong community model and make sharing of integrations and playbooks easy.

Let's take a deeper look at each of these capabilities:

Orchestration

Orchestration is defined as the machine-based coordination of complex workflows across different security tools, and is an essential capability for a SOAR tool.

When a security team responds to a security incident, they use a multitude of different security tools to respond. Each tool plays a different role within a defined workflow (depending on the type of security incident). For instance, tell VirusTotal to check a file's reputation, use your firewall to block an IP, and then use your endpoint security tool to block an executable. Without orchestration from a SOAR tool, the security team would coordinate these workflows manually. But a SOAR tool will integrate across all of these deployed security tools via their API, and then coordinate workflows across these tools to detect, investigate or respond to particular security incidents. For comparison, if your security tools are instruments that comprise a symphony orchestra, your SOAR tool is the conductor, ensuring that every instrument is playing in sync and on time.

When evaluating a SOAR tool, the orchestration function should direct and oversee all activities relating to a given security scenario from beginning to end. It should be able to ingest security data from any data source and in any format. It should be able to receive data that is pushed to the platform, and it must have the ability to poll data sources and ingest data into the platform. Furthermore, an orchestrator should ensure that the output data from one action is properly parsed, normalized and structured so that future actions can make use of it.

Automation

Automation is defined as the machine-based execution of otherwise manual, interdependent security actions using "playbooks." In other words, it's the workhorse of most SOAR tools. While the orchestrator enables integrations and coordination across security tools, playbooks automatically execute the interdependent actions from each security tool in a particular sequence — without the need for human interaction.

For most security analysts, their day is filled with too many repetitive and mind-numbing security tasks or actions. These actions are manually executed by the team. Automation using playbooks should allow the security team to execute a collection of these actions in seconds, versus minutes or hours if performed manually. For instance, phishing investigations that may require the use of multiple actions across four to five different security tools, and take approximately 40 minutes to perform if done manually, should now take under a minute using an automated playbook. In this way, SOAR tools can drastically reduce mean time to detect (MTTD) and mean time to respond (MTTR).

Playbooks should be easy to create and modify. The automation editor within a SOAR tool is where an analyst or manager codifies their processes into automation playbooks. The editor should allow for both source code editing and visual editing. This allows all security team members, regardless of preference or coding expertise, to construct comprehensive and sophisticated playbooks. While constructing the playbook in a visual editor, the resulting playbook source code should be generated in real time and be accessible to the author — with seamless toggling and editing between the visual and source code editor.

The visual playbook editor should be intuitive and user-friendly, providing a canvas where visual playbooks can be constructed. Using blocks and other shapes to represent meaningful steps in the playbook, a user should be able to build a playbook that connects actions in a one-to-one, one-to-many or many-to-one fashion to dictate the order of execution. Each shape should represent different action executions, platform API calls, conditional statements (if/then), human interaction prompts and branching statements. By clicking each shape, you can manually enter the action or parameter, or select them from a list. Also, new information resulting from preceding action executions should be available as inputs, or parameters, to downstream actions or decision blocks.

Event and Alert Management

Just after data ingestion, an event and alert management capability in a SOAR tool should queue and prioritize inbound events and alerts. This will enable alerts to be rapidly consumed and efficiently acted upon, without the need for extensive searching or switching between contexts. Events and alerts should include a status

indicator (for example new, open or closed), a severity indicator, and a color-coded sensitivity indicator to facilitate quick consumption of information.

The technical attributes of a security event or alert should be organized to allow for rapid understanding of the security scenario. This includes an organized view of data like IPs, domains, file hashes, user names, and email addresses. A security analyst should be able to seamlessly issue investigative, containment, or response actions (or a collection of actions, i.e. playbooks) against this data.

And finally, the SOAR tool should provide a comprehensive activity log that displays a record of all actions that have executed against an event or alert, whether they were initiated manually or via a playbook. Each action should display its results, including an indicator of action success or failure.

Case Management

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. The SOAR tool will take multiple events and then confirm, aggregate and escalate them into a single case. The case management interface should support attaching relevant technical data such as the alert's source data and action results to the case. The interface should also support attaching relevant non-technical data such as notes, memos, emails, screenshots, recordings or any other arbitrary file with relevance to the case. Any changes to a case should be logged in an audit trail and be exportable. Also, automated attachment of information to a case should also be possible from within a playbook.

Case management should also easily map to an organization's existing processes. Many organizations have developed standard operating procedures (SOPs) for incident response. The case management functionality should provide a user with the ability to define stages according to their process and save them as a template. A user should have the ability to break the SOP into multiple stages where each stage has one or more tasks, and each task can be assigned an owner. The interface should provide an indicator of progress for the case as well as the case status.

Collaboration

Security is a team sport. Analysts must collaborate with each other to respond quickly to security events. The more connected and in-sync your team, the faster and more effective they'll be at protecting the organization.

Therefore, a best-of-breed SOAR tool should include built-in collaboration features. Collaboration features like integrated chat and the ability to attach and share case notes should be available alongside the investigation or response workflow to provide in-context collaboration. With real-time chat and notes alongside event, alert, and case information, analysts can achieve a level of situational awareness that allows for efficient and fast resolution of security incidents.

This also creates an easy audit trail. It's ideal for the record of this collaboration to be captured and organized alongside the relevant event data and recorded actions that were taken. That's not so easy if your communication is on an external tool, separated from the workflow information in your SOAR tool.

Metrics and Reporting

A security team must be able to easily measure the state of their security operations, and drive toward continuous improvement over time. Therefore, robust metrics and reporting are a must-have for any SOAR tool. They help the security team understand the productivity impact of functionality like automation, and identify where improvements can be made to increase ROI.

Automation is used to increase operations efficiency across multiple functions of a SOC (security operations center). It is critical to understand the quantitative performance gain and resource savings that automation provides, and to have this information readily available via a dashboard within the SOAR tool. Examples of key performance metrics that should be available on the SOAR platform include mean time to resolve (MTTR), mean dwell time (MDT), 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), total number of open alerts, alerts opened and closed per day (hour, week, month), and performance against service level agreements (SLAs).

All of this preceding information should be easily organized and aggregated into reports for upper management and CISOs to quickly understand the overall state of their security operations (as well as the improvements that the SOAR tool is driving).

Mobility

SOAR platforms are designed to accelerate response times. To achieve rapid response, 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 platform 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 a laptop, respond to prompts from the palm of their hand, and always be reachable whether they're at their desk or not.

Scalability

A SOAR tool should grow with you as your organization grows. As you add more use cases over time, there will be additional processing load placed on the platform.

It is important to understand how the automation engine will scale both vertically and horizontally. It is expected that a user will be automating more use cases over time. With each additional use case, there will be additional processing load on the automation engine. The automation engine should be designed in a way that allows for vertical scaling (for example increasing CPU and RAM resources) and horizontal scaling (for example increasing server instances) to increase performance and protect the automation return on investment (ROI).

Open and Extensible

A SOAR platform should be designed for openness and extensibility. It should easily support incorporating new security scenarios, new products, new actions and new playbooks. Without it, a SOAR platform can lose its value over time.

With an open integration ecosystem, that follows a common standard and programming model, security teams can capitalize on a few benefits. New technologies can be quickly integrated into the platform without requiring any modification to the core platform, or negatively impacting automated playbooks. Users can develop support for additional integrations without permission or development cycles from the SOAR vendor. For instance, they can write their own integrations, develop homegrown applications, or write an early access API from a vendor.

Community Powered

The evolving nature of security drives the need for a community of professionals working together to share playbooks, best practices and strategies for dealing with the latest threats. Therefore, a SOAR tool must support a strong community model and make sharing of app integrations and playbooks easy.

Measure the installed base of a platform to gauge its associated community's collaboration potential. A large and active user community lets you share playbooks, apps or brainstorm ideas for new automation use cases. Moreover, vendor participation in the community is a strong indicator of their commitment to both the community and collaboration. To facilitate the exchange of ideas, the SOAR vendor should provide a community communication tool, like Slack, which enables group and direct messaging for technical support and questions. Additional communication tools to spread new ideas include community user Github pages where individuals publish their work, and a centralized community repository that hosts user presentations, blogs, community playbooks, contributed app integrations, and general documentation.

Can a SOAR tool help you improve your security operations? See how **Splunk's best-of-breed SOAR technology** can supercharge your security team's efficiency and effectiveness.

20-13733-10 Essential Capabilities of a Best-of-Breed SOAR-104-WP