Security Tool Integrations through APIs—A SecOps Best Practice You Can’t Ignore
Security operations (SecOps) teams are on constant watch for security events of interest and indicators of compromise. But as attacks continue to grow in sophistication and stealth, it gets harder and harder to identify them. That’s especially true if you primarily rely on a single security tool, no matter how powerful that tool might be. That’s why most organizations employ a variety of different security tools and point solutions, each with different strengths and capabilities. Still, mature SecOps recognize just having an arsenal of security tools isn’t sufficient and often complicates workflow. Unless those tools are integrated, analysts will find themselves jumping from tool to tool and suffering from a ‘swivel chair’ security workflow that negatively impacts visibility, speed, and efficiency.

To avoid these security disconnects, many vendors offer prebuilt integrations with other tools that simplify and help with needed security integration efforts. Although beneficial, these prebuilt integrations often have limitations with what other tools they can integrate and what integrated functionality they provide. These limitations may still leave you with silos that disrupt your security efforts. In contrast, tools designed and built to be open can greatly simplify integration efforts for SecOps teams and their ability to facilitate communication between disparate systems and vendor tools. This can be accomplished through APIs (application programming interfaces), which are by far the best way for any security organization to achieve the optimal level of tool integration.
Why APIs Are Great—Integrate, Communicate and Interoperate, Etc.

Formally defined, an API is "a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service." In other words, APIs enable you to create system-to-system communication among your tools, giving your different tools direct access to the functions and data of other tools and platforms. It defines a common language that can be used to query and provide responses between applications. As such, an API is one of the best ways to extract information from a system or application because you can control what data is requested and what actions you might want to initiate based on that data. In a security environment this opens the door to aggregating and sharing security information between different security tools, allowing them to be focused on their niche expertise, be it data collection and storage, reporting, visualization, correlation or alerting. Keeping tools focused leads to faster, more accurate alerting and investigation of events of interest. Additionally, APIs can enable automated interactions and responses among your different security tools freeing up valuable analyst man-hours, improving the overall SOC efficiency.

In a connected environment, an accepted integration method employs APIs built on REST (Representational State Transfer) principles. Although not required, it’s most commonly implemented as a way to exchange data using constructs that are familiar to anyone who is accustomed to using the internet’s Hypertext Transfer Protocol (HTTP). RESTful methods can be done in any programming language but use an architectural style for developing these web services, which includes:

- **Client-server architecture** that allows both components to evolve independently of each other.
- **Stateless** in that all communications from client to server contain all the information necessary to understand requests without relying on any server-side information.
- **Cacheable** option to allow client, server, and intermediary components to cache resources in order to reduce interaction latency and thereby improve efficiency, scalability, and performance.
- **Uniform interface** between components to simplify the overall system architecture and improve the visibility of interactions, while adhering to defined architectural constraints that guide component behavior.
- **Layered system constraints** within a hierarchical layered architecture that prevent components from being able to see or have knowledge beyond the layer where they currently interact, enabling component independence and facilitating scalability and security.

---

Leveraging APIs at the SIEM level can make wonders happen. Scripts can be written to automate manual tasks. Customizations can be developed that give organizations and individuals complete control over what data can be accessed and how the data can be displayed.

“ArcSight is generally integrated into the core of your network where you don’t want customers having access. We resolved this by using APIs because we can control what data comes out and what information should be presented to customers”

MARIUS IVERSEN
Senior Security Engineer

SIEM as a Unifying Integration Hub

While APIs enable you to create the tight integrations and communications you need among your different security tools, security information and event management (SIEM) solutions can serve as the central hub to align, unify, and bring together those integrations. As a centrally located platform that collects events across your entire organization from all your different sources, it makes sense that SIEM should be used to communicate back to those sources, as well as bi-directionally communicate with other security tools that need to consume data from those different sources. Leveraging APIs at the SIEM level can make wonders happen. Scripts can be written to automate manual tasks. Customizations can be developed that give organizations and individuals complete control over what data can be accessed and how the data can be displayed.

For example, if an analyst using a hunt tool sees an event of interest generated by a SIEM, the analyst should be able to grab more contextual data from the SIEM and the source surrounding that event without ever leaving the hunt tool. The generation of that particular event could even automatically trigger a workflow process or response. Leveraging the SIEM and its APIs not only should enable this, but they should also allow for the central orchestration of a wide variety of actions and responses, such that analysts and SOCs no longer have to shift from tool to tool, clicking different buttons and pulling up different screens and dashboards to further investigate or initiate different actions. Intelligent security operations now realize the power of automating and orchestrating security tools. This is evident in the rise of Security Operations for Orchestration and Automation (SOAR) tools themselves.

In an example specific to SOAR, perhaps an analyst only wants to work from a SOAR dashboard. Using SIEM APIs, as well as APIs to other systems as desired, the SOAR dashboard could have API-driven links to all the functionality the analyst needs. If a potential breach occurs on a particular system, after the SOAR tool automatically takes steps to remediate the situation, the analyst will want to investigate the extent of the breach. From the SOAR dashboard the analyst could click on one of the API-driven links to pull up specific logs from the SIEM and then click another link to compare those logs to search results from another tool to determine if the breach has spread or been contained.

Finally, another use case could be where a service provider offers SIEM as a service to its customers and it wants to provide a level of visibility of security activity to its customers through its own custom web portal. Likely, the service provider will employ a multi-tenant SIEM that monitors activity for all of its customers, but when an individual customer logs into the portal the service provider will only want to present activity specific to that customer. This can be accomplished using the SIEM APIs, giving the service provider complete granular control over what, how, and to whom data is presented.
Internal APIs provide the means for the technology provider to continually improve, upgrade, broaden, and evolve the capabilities of the solution they offer.

Figure 1. ArcSight

Overcoming API Complexity

While APIs offer the lofty interoperability promises of high levels of control, customization, and automation, cashing in on those promises is not always a simple effort. In large part this is because when a technology provider originally created a platform or tool, their main focus was to facilitate internal application communication development efforts. Internal APIs provide the means for the technology provider to continually improve, upgrade, broaden, and evolve the capabilities of the solution they offer. In fact, a technology provider might not have originally intended to ever make any of its APIs public. As a result, the documentation of those APIs often only cater to the needs of internal developers that already understand the intricacies of the solution’s codebase and internal development standards. As a result, these functions might be extremely difficult for an outsider to navigate, they may need additional security layers added before they can be safely used, and they may not fully adopt REST best practices.
With ArcSight ESM, an enterprise level SIEM, the focus was initially on its ability to communicate outbound to other tools. This was done through flexible integration commands and rule actions that can initiate most any type of script. More recently, customers have been requesting access to the rich set of data and correlation events within ESM, either for consumption into other tools, or for reporting/visualization back to stake holders. Thus, even though the ESM API was not originally created for public consumption, Micro Focus® published a 58-page ESM Service Layer (Web Services) Developer’s Guide that exposes some ESM functionalities as web services. Developers can use those web services to integrate ESM functionality into their own applications. But developers can also access the full non-polished API set through the ESM host. At more than 3,000 pages, the full, unabridged ESM API documentation can make it an arduous effort to simply figure out which APIs you might want to use and how they work.

Recognizing the extreme value that could be garnered by simplifying the consumption of the ESM API, the ArcSight community raised their voices for help. In partnership with Micro Focus, one community member took the initiative to make it happen. As a senior engineer for a leading Dutch telecommunications company, Marius Iversen works on the front lines of the cyber security landscape. On the job, he’s invested significant time and effort developing tools and creating customizations using ArcSight APIs to help his company maximize their use and investment in ESM. After hours, Marius has used that acquired expertise with the ESM API on a project aimed to help others in the community who recognize the value of leveraging ESM as a unifying center for integrating their different security tools.

To help in this initiative, Marius has taken advantage of the free tools Slate and Postman. Slate is an API documentation framework that gave Marius an automated way to publish to the web searchable information that developers need to understand and take advantage of the ESM APIs, including real-time examples. Slate enabled Marius to translate thousands of pages of API documentation into a few scrollable web pages. To make it easier for developers to test and start using the ESM APIs, Marius leveraged Postman. With a click of a button from Postman, developers can see on the fly the results of using any particular ESM API.

This Slate and Postman package for the ESM API provides a friendlier and easier way for organizations to take advantage of the API, facilitating security tool integrations and eliminating security silos. You can create greater customizations and interoperability that lead to higher efficiency, better visibility, faster detection, and more automated response.

### Simple ArcSight Logger Search API Flow

<table>
<thead>
<tr>
<th>Login</th>
<th>Search</th>
<th>Status</th>
<th>Retrieve Data</th>
<th>Logout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login returns a session ID used for all other API calls.</td>
<td>Queries against the database &amp; creates a SearchSessionID.</td>
<td>Status shows whether a query completes or fails. Also shows Hits, # Scanned &amp; Time Elapsed.</td>
<td>Several API calls can be used on the same search results for different formats.</td>
<td>Stops the session and removes content associated with this ID.</td>
</tr>
</tbody>
</table>
Fostering Openness and Community Support

Micro Focus recognizes the need for continued integration and the benefits it can deliver. Part of the new product vision incorporates open technologies, such as the Apache Kafka and Docker based Event Broker, which for the first time allows any external application access to the enriched data generated from the ArcSight connectors. ArcSight ESM will continue to lead as an open architecture that enables high levels of interoperability and Micro Focus plans to continue to deliver tools designed and built with the openness the industry need. Likewise, we welcome and support other community-led initiatives, such as the Slate and Postman package for the ESM API, that further expand openness and interoperability. We are committed to listening to the ArcSight Community and responding to its needs to ensure the success of our customers and community members.

Learn More
To get more involved in this effort or the ArcSight community, visit community.softwaregrp.com/t5/ArcSight/ct-p/arcsight.