The Power of SIEM: Reducing Detection and Response Time in the Attack Chain

An Akamai and Splunk User Guide
Table of Contents

Akamai and SIEM 1

An Overview of Akamai Security Events 2
  1. Source of attack: 2
  2. Context of attack: 2
  3. Nature of attack 3

Four Steps to Connect Splunk and Akamai Security Events 5
  Step 1: Enable SIEM Integration 5
  Step 2: Create a new user with “Manage SIEM” role 5
  Step 3: Provision credentials to work with the SIEM API 6
  Step 4: Install and configure your Akamai SIEM Splunk Connector 7

Leveraging Splunk’s Alerting Capabilities for Actionable Threat Intelligence 10

Conclusion 12
It’s no secret that security threats and events are increasing in frequency and complexity with each passing nanosecond. Organizations of every size and in every geography understand that minimizing the time to detect and respond has become more important than ever.

Exhibit A: Last year, one of the most notable security breaches involved Equifax, wherein 145+ million people were affected by an attack that exploited a Struts vulnerability and, in the end, forced Equifax’s CEO to resign. Notably, this attack was carried out over time, as the attacker managed to upload 30 malicious web shells over the course of four months.

Ultimately, this catastrophic breach resulted from Equifax’s failure to monitor and take action on security events early enough in the attack lifecycle. Equifax’s response was simply too late. Adapting the terminology of the “Cyber Kill Chain” framework developed by Lockheed Martin, we’d like to show you how a better security information and event management (SIEM) strategy can help you identify attacks in the initial stages of the kill chain rather than in the final stages.

In this white paper, we will explore how you can build and automate actionable threat intelligence by using technologies from both Akamai and Splunk in a SIEM architecture. The benefits of this approach include:

- Greater context on cyber attacks
- Reduced detection time
- Reduced response time in the overall threat lifecycle

First, let’s start with an overview of Akamai’s approach to SIEM.

**Akamai and SIEM**

The Akamai SIEM Integration is a powerful tool for capture, retention, and delivery of security information and events in real time to SIEM applications (e.g., Splunk). Akamai SIEM Connector for Splunk maximizes security and event insights with the ease of integration with the robust Akamai SIEM API. Following is the architecture of Akamai SIEM Connector:

As shown above, the data flow takes place as follows:

1. SIEM Connector retrieves Akamai security events via pull request through the API exposed by the Akamai Security Events Collector (ASEC).
2. Security events are delivered to SIEM Connector in JSON format.
3. Once SIEM Connector receives the security events from ASEC, it processes and sends the events to the SIEM solution.
An Overview of Akamai Security Events

There are three major components included in the Akamai security events that get delivered in JSON format:

1. Source of attack
2. Context of attack
3. Nature of attack

Let’s briefly take a look at each component.

1. **Source of attack**: Akamai provides location data on the attack source, leveraging EdgeScape capability for superior accuracy regarding the attacker’s location. Here’s an example of the data you’d see:

   ![Geo Data](image)

   Here’s an explanation of the data:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>asn</td>
<td>The AS number or numbers that the IP belongs to</td>
<td>4808</td>
</tr>
<tr>
<td>city</td>
<td>City that the IP address maps to</td>
<td>BEIJING</td>
</tr>
<tr>
<td>continent</td>
<td>A 2-letter code for the continent that the IP address maps to</td>
<td>AS</td>
</tr>
<tr>
<td>country</td>
<td>An ISO-3166, 2-letter code for the country where the IP address maps to</td>
<td>CN</td>
</tr>
<tr>
<td>regionCode</td>
<td>An ISO-3166, 2-letter code for the state, province, or region where the IP address maps to</td>
<td>BJ</td>
</tr>
</tbody>
</table>

2. **Context of attack**: Akamai provides context and details on each HTTP request. Here’s an example:
Here’s an explanation on a few fields of the data:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestID</td>
<td>A globally unique ID created to identify this specific message</td>
<td>f7f359</td>
</tr>
<tr>
<td>start</td>
<td>The time, in epoch format, to millisecond precision, when the Edge server initiated the connection for the message exchange being monitored</td>
<td>1520556215</td>
</tr>
<tr>
<td>protocol</td>
<td>Protocol of the transaction being monitored</td>
<td>HTTP/1.1</td>
</tr>
<tr>
<td>method</td>
<td>Method of the incoming request</td>
<td>GET</td>
</tr>
<tr>
<td>host</td>
<td>Value of the incoming client request’s host header</td>
<td>test20.*******.net</td>
</tr>
<tr>
<td>port</td>
<td>Port number used by the incoming request. Should be equal to the value of AK_IN_PORT</td>
<td>80</td>
</tr>
<tr>
<td>path</td>
<td>Path used in the incoming URI from the client, not including query strings</td>
<td>/</td>
</tr>
<tr>
<td>query</td>
<td>The query strings passed in the incoming URI from the client</td>
<td>id=a&amp;msg=a</td>
</tr>
<tr>
<td>requestHeaders</td>
<td>All request headers collected</td>
<td>Host: test20.*******.net</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User-Agent: curl/7.55.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Accept: <em>/</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>X-Forwarded-For: 123.120.23.157</td>
</tr>
<tr>
<td>status</td>
<td>HTTP response status sent to the client</td>
<td>503</td>
</tr>
<tr>
<td>responseHeaders</td>
<td>All response headers collected</td>
<td>Server: AkamaiGHost</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mime-Version: 1.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Content-Type: text/html</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Content-Length: 272</td>
</tr>
</tbody>
</table>

3. Nature of attack: Akamai characterizes the nature of each attack and provides details on the set of configuration rules that intercepted it. Here’s an example:

```json
{
  "attackData": {
    "clientIP": "119.72.199.211",
    "configID": "61617",
    "policyID": "c20_51556",
    "rules": ["["],
  },
  "rules": ["["],
```
Here’s an explanation on few fields of the data:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Example</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>configId</td>
<td>The ID of the Security Configuration applied to the request</td>
<td>16167</td>
<td></td>
</tr>
<tr>
<td>policyId</td>
<td>The ID of the Firewall policy applied to the request</td>
<td>t20_51596</td>
<td></td>
</tr>
<tr>
<td>clientIP</td>
<td>The IP address of the client that connects to make the request</td>
<td>119.72.199.211</td>
<td></td>
</tr>
<tr>
<td>slowPostAction</td>
<td>If a Slow POST attack is detected, this shows the action taken: either W for Warn or A for deny (abort)</td>
<td>W</td>
<td>Above attack is Bot Attack, so the flag did not appear</td>
</tr>
<tr>
<td>slowPostRate</td>
<td>If a Slow POST attack is detected, this shows the recorded rate of the Slow POST attack</td>
<td>10</td>
<td>Above attack is Bot Attack, so the flag did not appear</td>
</tr>
<tr>
<td>Rules id</td>
<td>Rule IDs of rules triggered for the request, base64-encoded</td>
<td>3000036</td>
<td>It includes on rule or multiple rules</td>
</tr>
<tr>
<td>ruleVersions</td>
<td>Versions of rules triggered for the request, base64-encoded</td>
<td>2</td>
<td>Represents version</td>
</tr>
<tr>
<td>ruleMessages</td>
<td>Messages of rules that triggered for this request, base64-encoded</td>
<td>Detected LOIC/ HOIC client request based on query string</td>
<td></td>
</tr>
<tr>
<td>ruleTags</td>
<td>Tags of rules that triggered for the request, base64-encoded</td>
<td>AKAMAI/DDOS/LOIC_HOIC _1_v1</td>
<td></td>
</tr>
<tr>
<td>ruleData</td>
<td>User data of rules that triggered for this request, base64-encoded</td>
<td>LOIC/HOIC based on query string</td>
<td></td>
</tr>
<tr>
<td>ruleSelectors</td>
<td>Selectors of rules that triggered for the request, base64-encoded</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ruleActions</td>
<td>Actions of rules that triggered for the request, base64-encoded</td>
<td>alert</td>
<td></td>
</tr>
</tbody>
</table>

So, that’s the kind of data you see once you have this solution implemented. Now let’s talk about how to get you started.
Four Steps to Connect Splunk and Akamai Security Events

To get started, follow these four basic steps:

**Step 1: Enable SIEM Integration**

Set the toggle switch to “On” in your security configuration:

After enabling SIEM Integration, you’ll need to activate it on the production network.

**Step 2: Create a new user with “Manage SIEM” role**
Step 3: Provision credentials to work with the SIEM API
Search for “SIEM” in the search box and the SIEM API will pop up.

Finally, you’ll get the API credential (aka “client secret”) as follows:
Step 4: Install and configure your Akamai SIEM Splunk Connector

There are a few sub-steps to get Splunk working with Akamai. First of all, you’ll need to download the Akamai SIEM Splunk Connector:

![Akamai SIEM Splunk Connector](image)

And install the connector on Splunk as follows:

![Splunk App Installation](image)

Finally, configure the connector to get Splunk working by fetching the username and password of the Splunk server and all the credentials that we obtained earlier in step 3.

Click “New”:
Then, fill in your credential info:

![Credential Form](image)

Once this is complete, Splunk can capture the attack data and display it within the Splunk dashboard. Just for demonstration purposes, we have attacked our site with web-based attacks as well as bot-based attacks. The attacks have been carried out globally by spoofing IP addresses in the XFF Header, as you can see here:

```bash
#!/bin/bash
if [ $# -ne 1 ]; then
  echo "Usage: hoge.sh filename"
  exit 1
fi
while true
do
    line=`shuf -n 1 ${1}`
    a=`echo $((RANDOM % 255))`
    b=`echo $((RANDOM % 255))`
    c=`echo $((RANDOM % 255))`
    d=`echo $((RANDOM % 255))`

    mline="${line} -H "X-Forwarded-For: "$a"" "$b"" "$c"" "$d"
    eval $mline
    slnum=`echo $((RANDOM % 10))`
    sleep 0.$slnum
  done
```
The dashboard in Splunk can be easily edited or added directly by changing XML source code, as seen here:

At this point, actionable alerts can be triggered within Splunk during attacks. So that brings us to our next subject: the alerting capability in Splunk.
Leveraging Splunk’s Alerting Capabilities for Actionable Threat Intelligence

Splunk has a very robust and user-friendly alert workflow. For starters, you can create alerts in the “Save As Alert” window, which looks like this:

![Save As Alert Window](image)

There are several interesting and helpful items in this window that I’d like to highlight. First is the “Alert type” control:

![Alert type Control](image)

You can adjust the alert type to configure how often the search runs. Use a scheduled alert to check for events on a regular basis, or use a real-time alert to monitor for events continuously.

Scheduled alerts offer granular date/time/frequency control.
The next item I’d like to highlight here is the “Trigger Conditions” control. An alert does not have to trigger every time it generates search results; you can set trigger conditions to manage when the alert triggers, whether that’s “Per-Result” or “Number of Results” or other conditions shown here:

![Trigger Conditions](image)

You can also throttle an alert to control how soon the next alert can trigger after an initial alert; your throttling intervals can be set to minutes or even seconds:

![Throttle](image)

When an alert triggers, it can initialize one or more alert actions. An alert action can notify you of a triggered alert and help you respond appropriately. You can configure alert action frequency and type in the “Alert Actions” window:
From the “Alert Actions” window, you can take actions such as:

- Send a log event to a Splunk receiver endpoint
- Send an email notification to specific recipients
- Create a generic HTTP Post to a specified URL
- Output the results of the search to a .csv lookup file
- Invoke a custom alert action

**Conclusion**

The SIEM approach described above provides a centralized view of security information from a large number of sources so you can easily access and analyze that information. This approach then enables you to quickly prioritize your mitigation efforts based on risk profiles. In addition, it can help you meet security log analysis and incident/event reporting requirements.

Whether you’re using Splunk, IBM, RSA, or any other SIEM solution, you can build and automate actionable threat intelligence by integrating Akamai security events within your SIEM. This gives you greater information on the source, context, and nature of cyber attacks, and enables you to reduce the time to detect and respond in the overall threat lifecycle.

For more information, watch the “Connecting Your SIEM Tool with Akamai Security Events” video presentation from the Developer Zone at Edge 2017.

Ajay Mishra is a Senior Enterprise Security Architect at Akamai Technologies.