What is the Log4j Vulnerability?

Log4j is a ubiquitous logging library used in Java applications. On December 9th, a critical flaw in Log4j was disclosed that a llows it to execute remote code on a host system, which can then steal data, mine cryptocurrency, install malware, or attack lateral systems. Due to Log4j's popularity and the devastating nature of this flaw, it is one of the most critical security threats IT teams have faced.

Is Cribl LogStream or Cribl AppScope vulnerable to this flaw?

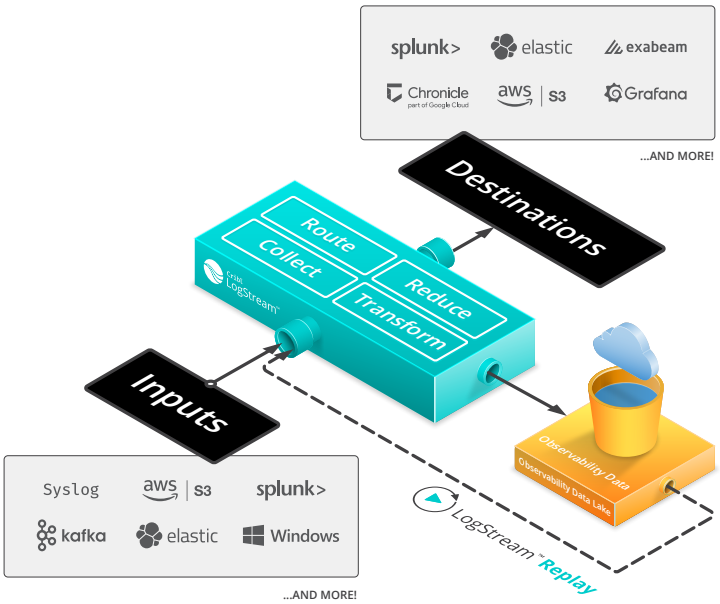
Cribl applications are not written in Java and aren't vulnerable to the Log4j flaw.

How can LogStream help my security teams manage the Log4j vulnerability?

As an observability pipeline, LogStream sits between the sources and destinations of observability data. This allows LogStream to enhance your security teams' monitoring of malicious requests that can trigger the remote code execution exploit. LogStream augments each stage of the security data pipeline:

- *Data Ingestion*
- *Enrichment*
- *Detection*
- *Incident Response*
- *SOAR*

Cribl LogStream links these processes together to provide effective and efficient data processing to enhance core security monitoring.



DATA INGESTION

LogStream parses and optimizes web server logs to make malicious inbound Log4j data more visible and faster to search.

ENRICHMENT

Next, LogStream offers the ability to enrich data so it offers more value to your SOC. For example, LogStream can take an export of indicators of compromise (IOC) data for known Log4j scanners. LogStream matches this data with inflight data and, if it sees a match, tags the event with a custom tag. The SOC can search on that tag and find results quickly. Additionally, LogStream supports tagging based on GeoIP data sources like MaxMind. LogStream's enrichment capabilities enhance data to help give SOC analytics better context and reduce the need for additional research. No more spreadsheets with dozens of tabs for manual data processing. You can make faster decisions using high quality data that was generated with less effort than previously possible.

DETECTION

Since LogStream is seeing all data in real time, security teams have the option to create a detection pipeline for data patterns. This can generate an alert to something like PagerDuty and/or trigger a SOAR playbook. The options are endless. Be careful since you are relying on static patterns so consider a lookup to scale the number of patterns that are being matched against inflight data. It is very easy to obfuscate strings in data so you will either need many samples or a very broad regex that considers all options as well. Use with care and thought. This is a power capability that can shorten the detention timeline and drive a faster response.

INCIDENT RESPONSE

When your SOC detects a compromise, the incident response (IR) process will start. One of the first tasks is determining the event timeline and to start looking for attacker pivots to assets in other parts of your network. If possible, you want to use raw, unprocessed logs to construct the attack timeline, but getting access to raw data can be a challenge. LogStream has a feature called Replay that makes this process fast and self-service.

SOAR

Finally, LogStream can be integrated into a SOAR like FireEye Helix or Splunk Phantom to execute a SOAR playbook and shutdown an attack in seconds. Since LogStream sees all inflight data, this process accelerates response.

CONCLUSION

LogStream acts as a force multiplier for your security teams grappling with the Log4j vulnerability and is foundational to how your teams manage detection and response in the future.

ADDITIONAL RESOURCES

To learn more about the Log4j vulnerability or to connect with others leveraging LogStream to combat it, check out our blog or join the Cribl Community.