≽ Cribľ Double

# CRIBL DATA DATA STORIES

YOUR GUIDE TO THE DATA ENGINE FOR CYBERSECURITY



Copyright © 2024 Cribl, Inc. | cribl.io

All rights reserved. No portion of this book may be reproduced in any form without written permission from Cribl, Inc., except in the case of copying for personal, non-commercial use or as otherwise permitted by U.S. copyright law. For permission requests, please email legal@cribl.io.

This book is designed to provide accurate and authoritative information for the use of Cribl, Inc.'s products. While best efforts have been used in preparing this book, Cribl, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaims any liability in connection with the use or result from the use of the information in this book. The information contained in this book might not be suitable for your situation. The actual use of Cribl, Inc.'s products must be in accordance with the applicable software license agreements and relevant documentation. Please visit docs.cribl.io for official documentation of Cribl, Inc.'s features and functionality.

The Cribl trademarks and brand names displayed in this book, including, without limitation, Cribl®, Cribl.Cloud®, Cribl Stream<sup>™</sup>, Cribl Edge<sup>™</sup>, Cribl Search<sup>™</sup>, Cribl Lake<sup>™</sup>, CriblCon<sup>™</sup>, Data Engine for IT and Security<sup>™</sup>, AppScope®, and the Cribl logo displayed above, are the property of Cribl, Inc. You may not use or display any trademarks or service marks owned by Cribl, Inc. without our prior written consent. All other product names, logos, brands, trademarks, registered trademarks, and other marks listed in this document are the property of their respective owners. All such marks are provided for identification and informational purposes only. The use of these marks does not indicate affiliation, endorsement, or ownership of the marks or their respective owners.

# Cribl Amazing Data Stories

Your Guide to the Data Engine for Cybersecurity

### Contents

Foreword
<b>Preface</b>
Acknowledgments/Editor's Note09
Optimize Security Data11
1. Route and Optimize Security Data13
2. Optimize High-Value Data for Cybersecurity Platforms
Enrich, Observe, Analyze, Respond45
3. Better Threat Hunting with Cribl Search Data Enrichment47
4. If You Can't Search It, Does It Even Exist?
5. Streamline Windows Monitoring with Cribl Edge

### Foreword

by Clint Sharp, Dritan Bitincka, and Ledion Bitincka, Cribl

At Cribl, we stand for choice. Choice for what to do with your data and what tools best fit your needs, today and in the future. After all, your data is your data. That's why the Cribl product suite is vendor-agnostic by design and purpose-built for you, the IT and Security professionals who keep the business running.

Your needs are unique and ever-changing. And what got you through the past 10 years won't get you through the next 10 years. Cribl is here to help you transform your data strategy and, ultimately, we want data to work for you, freeing up your time to focus on the many mission-critical tasks at hand.

From the outset, we've taken a first principles approach to innovating for you. We seek to understand your biggest problems, then we build the right solutions and continuously iterate to ensure our products work for you. As the founders, we have firsthand experience with many of the challenges facing IT and Security professionals today, but we don't pretend to know everything. We're insatiably curious and partner closely with you, our customers and users, as we continue to build our products. Before we dive in further, we want to give you a brief overview of our product suite. We started back in 2018 with Cribl Stream, the industry's leading observability pipeline that can collect data from almost any source, process billions of events per second, and transform and route data to optimize your analytics, storage, and retention.

Since then, we've broadened the scope of the company to meet your needs across data's entire lifecycle. We launched Cribl Edge to move intelligence closer to the data source and simplify managing fleets of agents, enabling you to deal with thousands more agents than you ever dreamed possible.

Next came Cribl Search, the industry's first search-in-place solution. Search gives you the insights you need, while minimizing the cost and complexity of ingesting and shipping data before analysis.

Most recently we launched Cribl Lake, a turnkey data lake solution that you can set up and configure in a few clicks, offering automated optimization of storage and open formats for easy access and retrieval without being a data expert.

That's where we are today on this journey, together with all of you, as we empower you to transform your data strategy. Now, you might be asking yourself: So what's this book all about? This is our guide for you, full of tips and techniques to get the most value out of your data, based on insights from the people who build and support our products and solutions for customers. The stories in this book illustrate a variety of approaches to getting your data to the right places, in the most optimized formats, and in the most efficient ways.

Whether you skim the chapters to see what's possible, or directly apply these techniques using the resources we provide, we hope this book will help you discover new ways to unlock the full value of IT and Security data. We also hope you have fun reading it. We know this is serious work, but we can still have fun doing it.

So we invite you to dive right in and join us on this ongoing journey. It's not science fiction. It's not a mystery. It's Cribl, the Data Engine for IT and Security.





### Preface

by Michael Katz, Cribl

It was midnight in the SOC. (It always feels like midnight here.) A cold wind blew the door open. I thought, good, the air conditioning's working. Then, in walked a dame wearing a million-dollar hoodie. That's what I noticed first, followed by the brightest eyes, the purplest hair, the whitest hat, and the highest red high-top sneakers I've ever seen. Nerds like that are one in a million.

"I've got a tip for you, Gumshoe," she said. She dropped a classy teal-colored business card on my desk. It seemed to float down in slow motion. (I thought, "Isn't it a little early in this movie for slow-mo?" Whatever.)

When I looked down, the card had just one word printed on it: "Cribl," it said.

That was it. The clue that changed my life. No longer did I need to pay big bucks to analyze a mess of irrelevant evidence, like events containing benign wellknown IP addresses. Stash that innocuous stuff on the cheap, in reliable long-term storage like Cribl Lake or S3. Focus on where the bad guys have left their fingerprints—like malicious domains, or access patterns that smell fishy.

Everybody's got an angle. But there ain't enough cash to go around, data volumes being what they are these days, to send every event to those big analytics shops downtown. See, we gotta be selective if we're gonna uncover the really crooked patterns, secure our infrastructure, investigate break-ins if they do occur, crack cases, and send the bad guys packing.

I followed the lead on that teal card, and I was glad I did. Cribl helped me put my money where it matters. I've got better overall data security, faster and leaner analytics, integrated data storage, and search-in-place. When crooks do attack, I can crack cases faster. And with my savings, I've even got a little extra to take to the track. I'm betting on a winning horse. After my winning bet on Cribl, I'm feeling lucky.

We get testimonials like this—but real ones, not made up—from our customers all the time. We're gratified by how well Cribl products have helped SecOps defenders up their game. In the following pages, Cribl's ace internal practitioners share tips on the techniques they've built for customers with Cribl Stream, Cribl Edge, Cribl Search, and Cribl Lake.

There are two sides to every amazing story, and you'll notice that there are two sides to this book. This side focuses on techniques of particular interest to Security sleuths. The flip side has a brief overview of Cribl's products and online resources, and then presents tips and techniques relevant to both Security and IT practitioners. We hope you'll read both!

There are a million stories in the city. Here, we present a few of our best. Turn the page and see where they take you.

(Pssst...yeah, you. If you ain't already in on the action, we got a hot tip for you: Sign up for a free Cribl.Cloud account at cribl.io/solved to start solving your own security hard cases. For the most up-to-date version of this book's contents, see cribl.io/amazing.)





## Acknowledgments/ Editor's Note

by Michael Katz, Cribl

Grigori Melnik had the vision for this book and set it in motion. Abby Strong, Joel Vincent, Erin Sweeney, and Meredith Torres kept it moving. Christopher Gales shaped it from an amorphous space blob into something flightworthy, adding a delicate balance of creativity and realism. Eric Wolfe and Mara Sahleanu provided graphics and illustration magic, developing some imagery in collaboration with ideogram.ai and Canva.

Mara and Alyssa Houk at Cribl, and Tamara Kreiss and Lesley Harrison at Bay Area Graphics, provided the overall design, layout, and production talent that gave this bookness. Douglas Howatt deftly copy-edited the whole manuscript and kept it real. Maliha Balala, Emil Mikhailov, Ed Bailey, Jackie McGuire, Igor Gifrin, and Josh Biggley provided crucial connections, content, encouragement, and more illustrations. Jennifer Marandola graciously allowed me a break from developing ongoing Cribl technical guidance to assemble this, working from the blueprint of a von Neumann probe. Marissa Dahlson, Diane Chiu, and Matt Lanza helped arrange my space boarding pass.

The greatest thanks go to all the authors for their ingenuity in writing these recipes, and for their patience in reworking text and graphics with me. And to Clint Sharp, Dritan Bitincka, and Ledion Bitincka for launching the generational starship that is Cribl. Any errors, misstatements, or anachronisms are my bad.





## SECTION 01

## Optimize Security Data

## 01 | Route and Optimize Security Data

## Unleashing the Cribl CrowdStrike Pack's model infrastructure

by Ahmed Kira, Cribl

#### PROBLEM

CrowdStrike FDR (Falcon Data Replicator) is one example of a class-leading endpoint monitoring solution. It collects a wealth of activity data from each managed endpoint, including network connectivity, DNS requests, process activity, and health checks. It reports more than 400 event types in all. These events are a gold mine for threat hunters and blue teams looking for unusual or malicious activity. But the resulting treasure trove is extremely voluminous, making it costly to place all this data in a SIEM (security information and event management) system.

#### SOLUTION

Cribl Stream offers a cost-effective solution, giving you the choice to send a full-fidelity copy of these logs to inexpensive object storage – such as Amazon S3, Azure Blob Storage, Cribl Lake, or your own object store managed by Cribl – while sending only events of interest to a SIEM.

#### WHY CONSIDER THIS APPROACH?

To accelerate this demonstration and your own setup, this chapter is written around the Cribl CrowdStrike FDR Pack, which offers predefined logic to address the common challenges of processing all this data. (You can apply the same principles to importing Cribl Packs for other SIEM destinations, or to building your own Cribl Stream optimization Pipelines.)

In this representative Stream case study, the end result you'll see is asset-level enrichment (for example, obtaining ComputerName from the aid field). You'll also see an impressive 40–95% reduction in data volumes sent to the SIEM. This reduction isn't just a matter of dropping unwanted events, but a combination of:

- Aggregating network events.
- Removing DNS requests that map to the 5,000 top popular websites.

Dropping unwanted events, based on the event\_simpleName field value.

- Removing unwanted fields.
- Removing null fields.
- Removing duplicate fields.
- Sampling noisy "External API" events.

In these screenshots from Stream's Pipeline diagnostics, you can see some examples of reduction in event volume and/or size, by transformation type.

	DNS Ever	nts – Drop requ	ests to top 5K a	ddresses			Network even	it aggregation	
	_raw Length ③	Full Event Length ③	Number of Fields ③	Number of Events (9)		_raw Length ③	Full Event Length ③	Number of Fields ()	Number of Events
N	238.80KB	261.09KB	1	336	119	289.4968	312.09KB	3	34
TUG	29,54KB	44.41KB	4	105	OUT	11.00%8	13.52KB	. 4	1
DIFF	÷ -87.63%	÷-82.99%	↑ 33.33W	÷ 68.75%	DIFF	¥-95.00%	÷↓-95.67%	† 33.33W	+-15.31
	Pro	cess Events - Re	emove select fie	alds		Diver	se dataset with	multiple event	types
	_raw Length ③	Full Event Length ③	Number of Fields ③	Number of Events		_raw Length ③	Full Event Length ③	Number of Fields 🖑	Number of Events
IN	288.40KB	302.53KB	3	213	IN	287.69KB	299.89KB	7	16
TUO	172.0388	198.85KB	4	213	OUT	150.30KB	171.28KB	5	10
DIFF	4-40.358	4.34 776	A 333386	0.00%	DIFE	4-12764	4 47 898	± 66.67%	4.11.41

Representative Stream optimization of security data

#### HANDS-ON RESOURCES

The CrowdStrike FDR Pack is a free resource, which provides its own instructions and sample data files. You can directly import the Pack to Cribl Stream, or download it, from the Cribl Packs Dispensary at packs.cribl.io. Browse the Dispensary for Packs that offer similar predefined optimization logic tailored to other SIEM and analytics destinations.

#### SCENARIO

The diagram below shows the Pack-based workflow we'll examine. The Pack optionally uses Redis for enrichment and stateful aggregation across Worker Processes. As a UI for the Redis service, we used Redis Commander (npmjs.com/package/redis-commander).



Default workflow using Cribl CrowdStrike FDR Pack

#### WHAT WE'LL DO

We'll cover the following techniques for isolating and enriching high-value events:

- Capturing data from CrowdStrike FDR.
- Enabling/disabling chained Stream Pipelines, for applications like DNS filtering.
- Controlling which events to drop.
- Enriching events with computer names.
- · Enabling Redis-based network aggregation.
- Smart processing of DNS events.

#### CAPTURING DATA FROM CROWDSTRIKE FDR

The CrowdStrike FDR Pack is designed to work with CrowdStrike FDR logs written to the CrowdStrike-provided Amazon S3 bucket. Contact your CrowdStrike team to obtain access to CrowdStrike FDR. When this is set up, CrowdStrike will provide an SQS queue URL, an S3 bucket URL, and an access key and secret key.

Within Cribl Stream, you will typically configure a CrowdStrike FDR Source with these details, as shown on the next page.

#### 01 | ROUTE AND OPTIMIZE SECURITY DATA

Configure Status Ch	arts Live Data Logs Notifications
General Settings	Input ID* ⑦
Authentication	CorpDFR
A deliver the deliver	inputId=='crowdstrike:CorpDFR' 🗍
Assume Role	Queue* ⑦
Processing Settings	∧ corpfdr-SQS-based-S3
	✓ OPTIONAL SETTINGS
Event Breakers	Filename filter ③
Fields	/ .*
Pre-Processing	Region ③
Advanced Settings	US East (N. Virginia)
0	Tags ⑦
<b>Connected Destinations</b>	Enter tags

Configuring a CrowdStrike FDR Source in Cribl Stream

(You could instead choose to configure an Amazon S3 Source in Stream. But note that an S3 Collector would not work here.)

### ENABLING OR DISABLING THE PROCESSING YOU WANT OR DON'T WANT

As illustrated in the diagram above, there is one master Pipeline that processes all streaming CrowdStrike events. Within that Pipeline are various Chain Functions with filters. To enable/disable any processing, just toggle that row's slider on or off. For example, if you don't have Redis, ensure that the Enrichment Chain Function is toggled off, as you see below.

#### ROUTE AND OPTIMIZE SECURITY DATA | 01

Ę		ribl_	crowdstrike	Routes Pipelines	Knowledge	Settings			
Cr	owo	lstrike	_General Attach	ed to 1 Route			+ R	Inction	۲
0	81	#	Function	Description		Filter		© A	• 11
Ģ	) []	1	Eval			true		On O	÷+)
¢		2	Comment	Timestamp fixing b	ased on event content.	in place			-
¢		3-4	Timestamp Fix	Known Events have	timestam r ContextT	imeStamp			
¢		5	Comment	Drop unwanted eve	nt_simpleNames as de	fined in			
¢		6 F	Drop			C.Lookup('event_	simpleNames_DROP.csv').ma	On	
¢	Î	7	Comment	Chain function for c	lifferent events				***
¢	)	8	Chain	Network Events		C.Lookup('event_	simpleName_NetworkEvents	On O	
¢	) []	9	Chain	Process & Services		C.Lookup('event_	simpleName_Processes.csv'	Qn O	
¢	Ì	10	Chain	DNS Requests		_raw.event_simpl	eName=='DnsRequest'	On O	
¢	)	11	Chain	Event_ExternalApiE	vent	_raw.EventType==	'Event_ExternalApiEvent'	On O	195
¢		12	Comment	Other events filter i	s a workaround to chee	k if any			
¢	) []	13	Chain	Other Events		!cribl_pipe		On	
¢		14	Comment	Enable enrichment	from Redis. Disable If n	ot using	Toggle off to d	isable	-
¢		15		Prosta branc Rection		tenn		030	

Dropping unwanted events

#### DROPPING UNWANTED EVENTS

One of the first things you'll likely want to do is control which CrowdStrike events Cribl Stream keeps and which it drops. To simplify this process, the Pack ships with a lookup file (CSV) containing event\_simpleNames values to drop. The Pack's main Pipeline, CrowdStrike\_General, contains a Drop Function that relies on this lookup to drop matching events.

#### 01 | ROUTE AND OPTIMIZE SECURITY DATA

Ø	∥ 2	Comment	Drop unwanted event_simpleNames as defin
$\diamond$	<b>3 F</b>	Drop	
	Filter ⑦		
	C.Loo	<pre>kup('event_simp</pre>	<pre>leNames_DROP.csv').match(_raw.event_simpleName)</pre>
	Descript	tion ⑦	
	Entera	a description	
	Final	Yes	

Drop Function's default configuration

To control the list of events being dropped, click into the Pack's own **Knowledge** > **Lookups** section.

å Gi	roup akira	Data v	Routing V Processing / 🕮 Pac	ks 🗸 🛞 Settings
۲	cribl_crowdstrike	Routes	Pipelines Knowledge Setting	35
m	Lookups		Lookups	
B	Parsers		III Filename	Description
2.	Global Variables		Event_ExternalAPI_DROP.csv	ExternalAPIType Events to drop
(.*)	Regexes		Event_ExternalAplEvent_ALL.csv	Events with ExternalAPIType field
{:}	Grok Patterns		event_simpleName_ALL.csv	Most simpleNames
	Schemas		event_simpleName_Files.csv	Crowdstrike File Events
			event_simpleName_NetworkEvents.cs	v Crowdstrike Network Events
			event_simpleName_Processes.csv	Crowdstrike Process Events
			event_simpleNames_DROP.csv	High Volume-Low Value fields-add or i
			majestic_million_100k.csv	Majestic Million Top 100K
			network_protocols.csv	Network Protocols

Customizing the lookup file of event names to drop

Knowledge - Lookups event_simpleNames_DROP.csv		×
Filename*		
Description ③ High Volume-Low Value fields-ad	1 är Hernawe fram list	
Tegs 🗇		
Fill Mode: Tata Tee:		Referenced: Crowdstrike_General
-G, Search calcons		
0	event_simpleName	comment
1	Disease Version Requirem	High Volume-Low Value Notla-add in remove from list
2	CurrentSystemTage	
1	LightningLatencyInfo	
4	SessorHeartbeat	
5	Restriction	
÷	ResourceColligation	
7	FalconServiceStatus	
	Pedersonkints	
	FirmwareliegionMeasured	
10	SportightCostyDatch	
Add Row O Delete Row		

Editing the lookup file's rows of events to drop

#### ENRICHING EVENTS WITH COMPUTER NAMES

Enrichment is valuable because the streaming CrowdStrike events include a computer ID only as the aid field. The actual Computer Name or hostname is not populated in those events.

#	Event Sample CrowdStrike streaming event	
1 🕺	araw:	_
2021-07-26	n aid: febc8fdbd0474b35a74e99b3ce3d6966	
09:59:01.153	u aip: 74.133.2.190	
-07:00	<pre>a cid: 56879ccf959c4afc96dd17e8bbldcbb5</pre>	
	clientComputerName: 192.168.17.10	
	d ConfigBuild: 1007.3.0014003.11	
	🗉 ConfigStateHash: 794409939	
	d EffectiveTransmissionClass: 3	
	Entitlements: 15	
	<pre>m event_platform: Win</pre>	
	<pre>o event_simpleName: SmbClientShareClosedEtw</pre>	
	id: c6885d4f-ee32-11eb-a953-0a436fff9fdf	
	name: SmbClientShareClosedEtwV1	
	🗉 SmbShareName: c\$	
	u timestamp: 1627318741153	

Sample CrowdStrike streaming event, omitting computer/host names

Instead, CrowdStrike exposes inventory information in separate events, which are typically published nightly.

#	Event	Sample CrowdStrike inventory event
1	# _time: 1650	503131.895
2022-04-20	a AgentLoadFl	ags: 0
18:05:31.895	a AgentLocal1	ime: 1630690863.6720002
-07:00	a AgentTimeOf	fset: 504.852
07.00	a AgentVersio	n: 6.27.14003.0
	a aid: a19c18	5c931a4607961efab81784eb58
	a aip: 73.168	.3.7
	a BiosManufac	turer: VMware, Inc.
	a BiosVersion	: VMW71.00V.16722896.B64.2008100651
	d ChassisType	: Other
	a cid: 56879c	cf959c4afc96dd17e8bb1dcbb5
	a City: Darie	n
	c ComputerNam	e: SE-SMC-MAC11-BL
	a ConfigIDBui	ld: 14003
	Continent:	North America
	a Country: Un	ited States
	a cribl_break	er: Break on newlines
	<pre>a event_platf</pre>	form: Mac
	a FalconGroup	ingTags: FalconGroupingTags/VMware
	a FirstSeen:	1620238836
	a HostHiddenS	tatus: Visible
	a MachineDoma	in: none
	a OU: none	
	a PointerSize	: none
	a ProductType	:: 1
	a SensorGroup	ringTags: -
	a ServicePack	Major: none

22

Because of how this information is sent, Cribl Stream can use Redis to enrich the ComputerName and/or other asset information in each streaming event. To enable this enrichment:

- 1. Stand up a Redis server. (See the end of this chapter for links to a setup tutorial, and to a video about Stream/Redis interaction.)
- Within the Pack, edit the Inventory\_Events\_Redis Pipeline, and open its first Redis Function.
- 3. As shown below, update the Redis URL, and create a user or admin password that you'll be able to reference in all other Redis Functions. Then resave this Function.

	S_REDIS ALLACH	ed to 1 Route				+ Function
11 # Fu	unction	Description		Filter		© /
2 Re	dis	entory') [  Comp	uterName.length > 0	true		On
Filter ③						Help
true						
Description	0					
source.inc	ludes('aidmaster')	source.includes("invent	tory')    ComputerNar	ne.length > 0		
Final 3	No					
Result	field 🗇	Command ③	Key ⑦		Args ⑦	
: _hmse	t_result	hmset	'aid.'+_raw.a	aid	['Computer	Name', 5
ii _hmse	edis URL to connec	hmset at to, format: redis[s]://[[u	'aid.'+_raw.a ser][:password@]]	aid .	「「Computer」 「」 600	Name', 兄
Honse	it_result edis URL to connec nost][:port][/db-nur umber[&password	hmset t to, format: redis[s]://[[u: nber][?db=db- =bar[&option=value]]]	'aid.'*_raw.a ser][:password@]]	aid .	「「Computer」 「「Gomputer」 「Gooo	Name', 7
Honse	edis URL to connec lost][:port][/db-nur umber[&password	hmset at to, format: redis[s]://[[u: mber][?db=db- =bar[&option=value]]]	'aid.'+_raw.a ser][:password@]]	aid d	兄 ['Computer 兄 600	Name', 5
iiihmse iiiex Rd If + Add n Redis URL* ( redis://redi	ec_result edis URL to connec lost][:port][/db-nur umber[&password ? s-cache:6379	hmset tt to, format: redis[s]://[[u: mber][?db=db- =bar[&option=value]]] Update Redis URL	'aid.'+_raw.a ser][:password@]]	aid .	500	Name', 5
hmse    ex R    ex R     redis URL* (     redis://redi	et_result edis URL to connec nost][:port][/db-nur umber[&password s-cache:6379 on Method	hmset tto, format: redis[s]://[u nber][?db=db- =bar[&option=value]]] Update Redis URL	'aid.'+_raw.a ser][:password@]] <sub>w.a</sub>	aid d	コ ['Computer コ <sup>600</sup>	Name', _
Authenticati	et_result edis URL to connect tost][:port][/db-nur umber[&password is-cache:6379 on Method Manual ③	hmset at to, format: redis[s]://[u mber][?db=db= =bar[&option=value]]] Update Redis URL User Secret ©	faid. f•_raw.a ser][password@]] Admin Secret ⊙	aid a	コ ['Computer 戸 600	Name', 5
Authenticati None @	et_result edis URL to connect iost][:port][/db-nur umber[&password is-cache:6379 on Method Manual ©	hmset at to, format: redis[5]://[[u mber][?db=db= =bar{&option=value]]] Update Redis URL User Secret ②	faid. ferraw.a	nid i	고 ['Computer	Name', 🤉
hmse    ex R     r     r    ex R     r     r     + Add     r     redis://redi     Authenticati     None ③     Username ③     Enter user	et_result edis URL to connect host][:port][/db-nur umber[&password is-cache:6379 on Method Manual @ D name	hmset at to, format: redis[s]://[[u mber][?db=db= =bar{&option=value]]] Update Redis URL User Secret ②	faid. f•_raw.: ser][:password@]] , Admin Secret ⑦	aid .	지 ['Computer 지 600	Name', 7
hmse    ex R    ex R     redis:URL* (     redis://redi     Authenticati     None ③     Username ④     Enter usern Password* ④	et_result edis URL to connect host][:port][/db-nur umber[&password is-cache:6379 on Method Manual @ b hamp	hmset at to, format: redis[s]://[[u mber][?db=db= =bar{&option=value]]] Update Redis URL User Secret ⑦	faid. f+_raw.a	aid	지 ['Computer 지 600	Name', 7

Configuring first Stream Redis Function with URL and authentication details

4. In the Inventory\_Enrich\_FromRedis Pipeline, update all other Redis Functions with the same details you set in the first Function. (This Pipeline works in the opposite direction, closing the enrichment loop by pulling data back in from the Redis store.)

In	ventory_i	Enrich_FromRedis	Attach to Route		+ Function 🔅
**	III #	Function	Description	Filter	© All •
Ģ		bes.	Press graw an (b) - towardy a public adjust	incompleteray(_raw)	
¢	2.4	Enrich from Redis	ENABLE ON to minimize Redis polling		(0n 🔿 🖓
	Group	p Name* Enrich fr	om Redis		
	Descri	iption () ENABLE	ONLY ONE FUNCTION in this group to minimize R	edis patting	
	0	2 Redis	Enrich Comp th light from Re	dis true	<b>GT</b> ····
	0	3 Redis	Update all 3 Functions with your Redis info	dis true	0.0
	Ø	4 Redis	Enrich ALL fields etall from Re	dis true	GnO
é	5-6	Ratten Inventory	array Only applies ing inventory is enabled		Gn 🖸 …
6	7-9	Remove Empty A	rrays		010

Next, update the remaining Redis Functions to match

5. Still working within the Pack, enable the Route that writes inventory events to Redis.

0	cribl	_crowdstrike	Rout	es Pipelines	Knowledge	Settings	
a 7	Sea	ch courtes				+ Route -	
×		Route		Filter	Pipeline		© All
Q	4	And Providence of the Annal State of the Annal Stat	n Cymrae	source.index0f(	(rwentory.)	vents_Redis	2
		Route Name*	Redis_Inve	ntery_Population		Toggle this Route on to write to Rec	in.
		Filter 🕤	source.1	ndexOf('fdrv2/aima	ster') > -1		۶. ۲
		Pipeline* ③	Inventory_	Events_Redis			Ø
16	Enable	e Expression ③					
		Description @	Enler a des	scription			
		Final 💮	Yes 🔵				
3	2	Crowdstri	ke_Events	source.indexOf(	- Crowdstrike	g_General	••
3	3	default		true	main	G	0
	$\rightarrow$	endRoute	- Data reachi	ing this point will be ro	uted to the Defaul	Destination	

Enabling the Pack's included Redis\_Inventory\_Population Route

6. Enable the Chain Function that enriches events from Redis data. This Function enriches events by chaining processing to the Pack's separate Inventory\_Events\_Redis Pipeline. (As with other details here, this Function's position in the latest Pack might differ from this screenshot.)

Pack cribl_crow	vdstrike Routes	Pipelines Knowl	edge Pack Settings	
Crowdstrike_Gen	eral 0 In 0	Out 0 Err Attached t	o 1 Route	Add Function 🕀 🔯
≎ III All •	Function	Description	Filter	
0    🗖	Parse the event as JSON	ONLY IF it's truly a JSON event	otherwise, it _raw will become empty	
2 🛛 🗖	Eval	Stop parsing non-JSON	events !JSON.parse(_	raw)
3	Eval	SON parse _raw	true	
	Timestamp fixing based	on event content. In place inc	ase Event Breakers arent imported	
13 1 🔍	Chain	Event_ExternalApiEven	raw.EventTyp	e=='Event_ExternalApiEve
14	Chain	ScriptControlScanInfo	events _raw.event_si	mpleName=='ScriptControl
15)    🜑	Chain	Hex Code in ScriptCont	entBytes field _raw.ScriptCo	ntent8ytes
16 I 📮	Other events filter is a v	varkaround to check if any oth	er pipelines have been processed inside the Crow	vdstrike_General pipeline. Redis ····
1)    🜑	Chain	Other Events	!cribl_pipe	
18    📮	Enable enrichment from	n Redis. Disable if not using Red	lis	
19	Chain	Enrich from Rodes	true	
20 Toggie t	he Enrich from Re	dis Function on	k TA Jor FDK. Senij to Sphilor via нес RAW, empl	unt to have the 1A quarter events
2)    🔘	Eval	Splunk Destination Fiel	ds true	

Enabling the Chain Function that enriches Redis events

By default, this chaining populates all contents of the CrowdStrike inventory event in Redis, making them available for enrichment.

However, if all you care to enrich is ComputerName, it would be wise to update the Inventory\_Events\_Redis Pipeline to write only those fields into Redis.

The Pack's predefined Pipeline actually offers you three enrichment options: Enrich only the ComputerName field, or enrich a set of fields that you control, or enrich all available fields. These options are exposed as three separate Functions in the Inventory\_Enrich\_FromRedis Pipeline. To prevent unpredictable interactions, you should enable only one of these Functions.

#### 01 | ROUTE AND OPTIMIZE SECURITY DATA

Inver	ntory Enrich En	omRedis			
Ø	2-4 Enric	h from Redis	ENABLE ONLY ONE FUNCTION L. to minimize Redis polling		Enable the group
	Group Name*	Enrich from R	edis		
	Description ③	ENABLE ONLY	ONE FUNCTION in this group to minimize Redis polling		
	0 2	Redis	Enrich ComputerName ONLY with hget from Redis	true	
	0 -	Dist 1	construction for SymPress 10 and 10	anya -	Enable only one of these Functions
	6 .		Errock ALL Index was repared from konti-	True	· · · · ·

Enable the enrichment group, then enable only one nested enrichment Function

If you want to closely control which fields are enriched, edit the third "multiple fields" Function, and add the desired fields to its Args section. Maintain the same format, with comma and single quotes.

3 1001	Etimen musique helos with h	true true true			
Filter 🕥					Help 💽
true					刃
Description ①					
Enrich multiple fields with hmg	get from Redis		Add the field	s you'd like to enrich	
Final () () 140			to the Args in	st in the same format	
Result field ③	Command ③	Key ③		Args ③	
E Computer_OU	hmget	'aid.'+_raw.aid	2	['ComputerName','0U']	IJ X
+ Add			L.		

Customizing which fields to enrich

After you save, commit, and deploy this change, every event will be enriched with the information you want.

#### STATEFUL AGGREGATION THROUGH REDIS

The Pack offers two methods for aggregating the voluminous network connectivity events: Either enrich natively in Cribl Stream via an Aggregations Function (the default), or enrich statefully across all your Stream Worker Processes via Redis. You should enable only one of these groups (not both) within the CrowdStrike\_Network Pipeline. The aggregation works by counting the number of times the client connects to the same server on the same destination port. The aggregation logic assumes that the lower port in a network event is the server port, and that the higher port reflects a client-side source port. This logic preserves the values of the unique source ports, and passes them as a list in the aggregated event.

For example, assume that CrowdStrike reports the following seven events:

Source IP: 192.168.1.1 Source port: 12345 Destination IP: 172.16.9.9 Destination port: 53 Source IP: 192.168.1.1 Source port: 12346 Destination IP: 172.16.9.9 Destination port: 53 Source IP: 192.168.1.1 Source port: 12347 Destination IP: 172.16.9.9 Destination port: 53 Source IP: 192.168.1.1 Source port: 12348 Destination IP: 172.16.9.9 Destination port: 53 Source IP: 192.168.1.1 Source port: 12345 Destination IP: 172.16.9.10 Destination port: 389 Source IP: 192.168.1.1 Source port: 12345 Destination IP: 172.16.9.10 Destination port: 389 Source IP: 192.168.1.1 Source port: 12345 Destination IP: 172.16.9.10 Destination port: 389

Aggregation will distill these into the following two events:

Event 1:

Count: 4 Source IP: 192.168.1.1 Source ports: [12345,12346,12347,12348] Destination IP: 172.16.9.9 Destination port: 53

Event 2:

Count 3 Source IP: 192.168.1.1 Source ports: 12345 Destination IP: 172.16.9.10 Destination port: 389

The aggregation period is set to a default range of 10 to 30 seconds, but you can easily modify these defaults in the Redis Aggregations group's Eval Function.

#### 01 | ROUTE AND OPTIMIZE SECURITY DATA

Crowds	strike_Net	work Attach to Ro	ute					
X IIII	# Fu	nction	Description	Filter				
0	) 🗄 8 Comment		GROUP B: Aggregation via Redis					
Ø	9-19 Red	lis Aggregation						
G	iroup Name	e* Redis Aggregatio	n					
D	escription	② Enter description	1					
6	9	Eval	Agg Thresholds Definition	true				
j.	Filter	Filter 🗇						
	tru	true						
	Descri	Description ⑦						
	Agg	Agg Thresholds Definition						
	Final (	Final ⑦  No						
	Evalua	Evaluate Fields ⑦						
		Name ⑦	Value Expression ③					
2	1	Agg_Count_Threshold	100					
	ij /	Agg_Period	60 Change Aggregation period if desired					

*Configuring the interval for Redis aggregation (in a separate Eval Function)* 

#### To enable the Redis aggregation, the steps include:

- Click into the Pack's CrowdStrike\_Network Pipeline.
- Disable GROUP A: Cribl Native Aggregation.
- In GROUP B: Aggregation via Redis, edit all Redis functions to update the Redis URL and user or admin secret.

- Enable GROUP B.
- Save, commit, deploy, and enjoy.

The result will render aggregated events similar to these:

IN OUT	l i	8 ()		nl	Select Fields (4 of 4)
#	Event				
1 × 2021-07-26 08:19:00.000 -07:00	<pre>() [raw:</pre>	97961efabi etworkConi 9.0.0 9.99.2.23 40 er	31784eb nectIP4	58	
2 × 2021-07-26 08:20:00.000 -07:00	<pre>cribl_pipe: 2 items</pre>	 97961efabi etworkConi 9.0.0 9.99.2.23 90 er	81784eb hectIP4	58	

Aggregated events

#### DO YOU REALLY NEED ALL DNS EVENTS?

DNS data is very voluminous, but has huge value for Security Ops teams. However, do you really need to keep DNS records for known addresses? Examples would be records for someone accessing Gmail, Facebook, and other non-sensitive endpoints. You will also have events that show you file transfer activity and process-level information, many of which will be redundant in a different way. The DNS records you truly need are for malicious addresses, not the top 10, 5,000, or even 100,000 addresses.

The CrowdStream FDR Pack ships with a lookup file containing the Majestic Million top 5,000 URLs. The Pack's Crowdstrike\_DNS Pipeline contains several Lookup Functions that work with this CSV. If the last four portions of an event's domain match against this list of common domains, Stream will drop the events. (An example matching domain might be my.docs.google.com.)



DNS lookup preserves unusual events, while dramatically reducing routine data flow
#### VARIATIONS

The CrowdStrike FDR Pack accounts for the most typical use cases, and for the largest-volume event types, to help you maximize value from the combination of Cribl and CrowdStrike. But the Pack is just a starting point. You can modify it and add content to it to suit your needs.

Cribl support and peer advice are readily available on Cribl's online Community resources. And on the Cribl Packs Dispensary, you'll find other Packs to optimize data for other SIEMs like Exabeam (the focus of this book's next chapter), SentinelOne, and Microsoft Sentinel.

#### WHAT'S NEXT?

For details about the resources listed in this chapter, see:

- Cribl Packs Dispensary (packs.cribl.io).
- Cribl CrowdStrike Pack demo video (youtube.com/watch?v=WohpshjFtzE).
- Stream CrowdStrike FDR Source (docs.cribl.io/stream/sources-crowdstrike).
- Stream Amazon S3 Source (docs.cribl.io/stream/sources-s3).
- Cribl Lake (docs.cribl.io/lake).
- Redis server setup, third-party tutorial (flaviocopes.com/redis-installation).
- Redis Commander (npmjs.com/package/redis-commander).
- Other Redis GUI options (redis.com/blog/so-youre-looking-for-the-redis-gui).
- Cribl Redis Lookups demo video (vimeo.com/504479828)
- Aggregations Function (docs.cribl.io/stream/aggregations-function).
- Redis Function (docs.cribl.io/stream/redis-function).

- Eval Function (docs.cribl.io/stream/eval-function).
- Sampling Function (docs.cribl.io/docs/sampling-function).
- Dynamic Sampling Function (docs.cribl.io/stream/dynamic-sampling-function).
- Suppress Function (docs.cribl.io/stream/suppress-function).
- Cribl Stream Data Preview (docs.cribl.io/stream/data-preview).

Cribl Stream's capabilities are always evolving, and you might see some differences from the UI screenshots and instructions provided in this edition. Luckily, Cribl Packs keep evolving, too – expect the Pack illustrated here to be progressively updated with new capabilities. (If you've customized the Pack's Pipelines, Stream ensures that importing an upgraded Pack won't overwrite them. Details at docs.cribl.io/stream/packs/#upgrading.)



## 02 | Optimize High-Value Data for Cybersecurity Platforms

Filtering Options Using Cribl Stream and Cribl Packs

by Jim Apger, Cribl

#### PROBLEM

Security-related inbound data often includes low-value events. These events drain resources that you could redirect to expanded discovery and analysis.

#### SOLUTION

This chapter shows how to filter low-value events out of syslog data ingested from sources like Palo Alto Networks, before you send your data on to common security analytics platforms. (We use Exabeam as our analytics example here.) We'll show how to enable relevant Cribl Stream Functions, integrations, and routing.

#### WHY CONSIDER THIS APPROACH?

By filtering out and dropping low-value data, your security organization can create additional capacity for new data sources, while reducing storage costs. If you track a cybersecurity framework (CSF) like MITRE ATT&CK, these filtering capabilities can help you address visibility gaps and increase your coverage of attack tactics and techniques.

#### HANDS-ON RESOURCES

If you aren't already using Cribl Stream, sign up for a free Cribl.Cloud account at cribl.io/solved to get your data flowing, with or without Packs.

As shown below, you can directly import many of this chapter's techniques from Cribl's Exabeam Pack for Palo Alto. It's a free resource on the Cribl Packs Dispensary at packs.cribl.io.

For details on the Cribl Stream Source and Destination integrations we reference below, see docs.cribl.io/stream.

#### THE END UP: CONFIGURE EXABEAM TO RECEIVE CRIBL STREAM DATA

Let's start on the output side – how Cribl Stream should relay inbound syslog data to Exabeam. As of this writing, we recommend creating a Webhook Cloud Collector in your Exabeam console: Select **Collectors** > **Cloud Collectors** > **New Collector**, and then select the **Webhook** tile.

Name your Webhook Collector, and select **JSON** as the format. This will display an **AUTHENTICATION TOKEN** and a **LINK TO LOGS**, which you will use in Cribl Stream next.

#### OUTPUT: CONFIGURE CRIBL STREAM TO SEND TO EXABEAM

Now, in Cribl Stream, configure a corresponding Webhook Destination to send data to Exabeam.

In your Stream Worker Group, select **Data** > **Destinations** > **Webhook**. Next, select **Add Destination** to open and configure the modal shown below.

We've set the URL field by pasting the **LINK TO LOGS** value from the Exabeam Cloud Collector.

	and the second second second	
Status Cha	irts Live Data Logs Test Notifications	
General Settings	Output ID* ③	
TIS Sattings (Client Side)	Eusbeam	
Tes settings (chefit side)	URL* ③	
Authentication	https://apl2.us-west.exabeam.cloud/cloud-collectors/v1/logs/json	
Processing Settings	OPTIONAL SETTINGS	
Part Processing	Method 🗇	
Post-Processing	POST	
Advanced Settings	Format ③	
	JSON Array	
	Backpressure behavior ①	
	Block	
	Tags 🕥	
	Emer tags	

Stream Webhook Destination, configured for Exabeam – General Settings

On the **Authentication** left tab, we've pasted the **AUTHENTICATION TOKEN** value the Exabeam Cloud Collector.

LADean		
onfigure Status Cha	irts Live Data Logs Test Notifications	
General Settings	Authentication type ③	
TIS Settings (Client Side)	Auth token	
its settings (client side)	Token* ③	
Authentication		
Processing Settings		
Post-Processing		
Advanced Settings		

Stream Webhook Destination, configured for Exabeam – Authentication settings

## TEST OUTBOUND THROUGHPUT (OPTIONAL)

In the above screen capture, note the **Test** tab available at the upper right. You can select this tab to send a few events over to Exabeam, and verify that they appear in the Exabeam console. (If you send a customized event, make sure it's a JSON event wrapped in an array.)

#### INPUT: CONFIGURE CRIBL STREAM TO RECEIVE SYSLOG DATA

In our scenario, we want Cribl Stream to ingest syslog data and then transmit it exactly as Stream received it. To set up the ingest side:

In your same Stream Worker Group, select **Data** > **Sources** > **Syslog**. You can then enable one of the default Syslog Sources, which will select the default TCP and/or UDP port.

## RETAINING \_RAW FOR DATA FIDELITY

We'll need to add one very important custom setting to the Syslog Source, as shown below. In **Fields to keep**, enter <u>\_raw</u>. This way, we'll retain events' original <u>\_raw</u> field as we process the events and send them on to Exabeam.

nfigure Status Cha	rts Live Data Logs Notifications	
General Settings	Input ID* ③	Enabled (
TIS Settings (TCP Only)	Ekabeurr, Sytlag	
tes settings (reir only)	inputId.startsWith('syslog:Exabean_Syslog:')	
Persistent Queue Settings	Address* ②	
Processing Settings	0.0.0.0	
Trees and a compa	UDP port ③	
Fields	9514	
Pre-Processing	TCP port ③	
	9514	
Advanced Settings	V OPTIONAL SETTINGS	
Connected Destinations	Fields to keep (3)	
	iraw ∞	
	Tags ③	
	Ector Lags	

Retaining \_raw to ensure data fidelity

Before we complete connectivity in Stream, we want Cribl's Exabeam Pack for Palo Alto available. The Pack will provide the Stream Pipelines and Functions that optimize our syslog data – so we won't need to write those. We'll plug this Pack into Stream in place of a Pipeline.

If you're working through this scenario as you read, you can download the Pack from the Cribl Packs Dispensary at packs.cribl.io, or install it directly from within Stream by selecting **Processing** > **Packs** > **Add Pack** > **Install from Dispensary**.

#### CONNECT EVERYTHING: ROUTE INBOUND TO OUTBOUND DATA

To complete the circuit, we need to connect our Syslog Source to our Exabeam Webhook Destination. We do this with a Route.

With the focus on your same Worker Group, select **Routing > Data Routes > Add Route**. Then configure your new Route as shown below.

The **Filter** expression says: Where incoming syslog data bears the Syslog Source's Input ID, this indicates syslog data destined for Exabeam. So send the event to the Exabeam Pack, then route it to our corresponding Webhook Destination.



Route configuration for PAN syslog to Exabeam

#### ONE ROUTE OR MANY?

With the **Filter** expression shown above, and with the Pipeline configuration we've imported from the Pack, we're assuming that this Syslog Source will receive only Palo Alto Networks (PAN) data. Configure your PAN sender to relay data to Cribl Stream via syslog. If you had multiple data sources sending events into this Syslog Source, you'd need multiple Routes. Each would have a tighter **Filter** expression than we've shown above, tuned to capture specific events that you want to direct into a corresponding Pipeline (or Pack).

## CHECK INBOUND EVENT STRUCTURE

After you've configured Palo Alto Networks to send syslog events to Cribl Stream, you can confirm the inbound data structure by performing a capture on your Syslog Source in Stream. You should see events looking roughly like the screenshot below – containing only \_raw, \_time, and cribl\_breaker fields. (That last field identifies the Stream Event Breaker that isolated these fields.)

Status	Sample Data Simple Preview ③ Full Preview ③
Sample	data file Pipeline w-network-traffic.log
IN OUT	
#	Event
1 2023-04-08 09:52:20.153 -05:00	<pre>a _raw: <la>Feb 03 10:51:57 SERVER.00MAIN.CON 1,2022/02/03 10:51:57,0009C100744,TRAFTC,stcrt,1,2022/03/03 10:51:57, 10:118.64.58,134.132.48.63,144.04.0.4,134.132.48.63,APC-PROTECT-INET-HTTPS,domain\user123,is51,vsy1,APC-INS IDE, APC-INTENT, ethernet1/24, ethernet1/24, APC-PANDRAM-LOG5,2037/01/23 10:51:57,34523421,1,03305,443,55004,4 43,0x409053,tcp,allow,3349,1661,1608,31,2017/01/23 10:51:40,15,business-and-economy.0,31957071169,0x0,10.0. 0.0-10.255.255.255,US5,US,017,14 Show less _time:160e065540,153 a crib1_breaker: Break on newlines</la></pre>
2 2023-04-08 09:52:20.153 +05:00	<pre>n _row: &lt;14&gt;Sep 21 13:03:56 PA-VW 1,2018/09/20 13:03:56,44A183FC68F5304,TRAFFIC,end,2049,2018/09/20 13:03:58,10.0.0. 103,10.0.0.102,10.0.0.103,10.0.2.65,splunk,,,incomplete,vsys1,trusted,trusted,ethernet1/3 Show more # _time: 1680665540.153 0 cribl_breaker: Break on newlines</pre>
3 2023-04-08 09:52:20.153 -05:00	<pre>n _raw: &lt;14&gt;5ep 20 13:03:58 PA-VW 1,2018/09/20 13:03:58,44A183FC68F5304,TRAFFIC,end,2049,2018/09/20 13:03:58,34.217. 108.226,10.0.0.102,34.217.108.226,10.0.2.65,splunk,,,incomplete,vsys1,untrusted,trusted,e Show more</pre>
4 2023-04-08 09:52:20.153 -05:00	<pre>a _raw: &lt;14&gt;Sep 20 13:03:57 PA-VM 1,2018/09/20 13:03:57,44A183FC66F5304,TRAFFIC,end,2049,2018/09/20 13:03:57,34.216.</pre>

Data preview – inbound syslog Events

#### STREAMLINE DNS DATA

Let's switch gears a bit. So far, we've looked at Palo Alto Networks (PAN) traffic delivered to Cribl Stream via syslog. Let's now look at DNS Security logs from PAN. DNS data is highly valuable within a security operations center for multiple practitioners – including analysts, investigators, threat hunters, and threat intelligence engineers.

Unusual DNS requests and responses hold valuable potential as Indicators of Compromise (IOCs), especially during investigations after an incident. However, the challenge lies in the massive log volume that these requests and responses generate for security teams to navigate.

#### LESS DATA, MORE INFORMATION - WITH NO COMPROMISE

Built into Cribl's same Exabeam Pack for Palo Alto are options geared to solving that problem: What we generally care about in the DNS data are the anomalies within the giant dataset. The data we seek is typically in domains seen for the first time, newly registered domains, and domains generated by a Domain Generating Algorithm (DGA).

You can use Cribl Stream to send a drastically reduced DNS dataset to your security analytics platforms. At the same time, you can divert the full-fidelity feed to low-cost object storage or a data lake for long-term retention – whether that's Cribl Lake, or a lake on any other host.

#### UNPACKING THE DNS PIPELINE

To see how to accomplish this massive event reduction, let's look at the DNS Pipeline from the same Exabeam Pack for Palo Alto. If you're following along: From within your Worker Group, select **Processing** > **Packs** > **Exabeam Pack for Palo Alto**. Then, from the Pack's own submenu, select the **Pipelines** tab, where we can select Palo\_Alto\_NGFW\_DNS to examine its configuration, shown below.

۰ چ	Stream 🗏	Home Manage W	tonitoring Settings				Q Search Stream	o 🟸 🕸 💋
Grou	up detault	<ul> <li>Overv</li> </ul>	new Data * Routing *	Processing T Pro	jects Group	Settings	Workers 🛂 2 🔶	95a3306 V Commit & Deptiny
Pack	UL ex	abeam Exabeam	Pack_for_Palo_Alto Rowt	es Pipelines	Knowledg	e Pack Set	ttings	Attached to 0 resources
Pale	Alto_NGF	Show Stats Attac	ched to 1 Route	Add Function	Status	Sample Dota	Simple Preview @ Full Preview @	0
*	-IA []	Function	Filter		Sample	cata file	Pipeline	
1	1.	+++ Important - Be mind	ful of the Exabeam Parsing process as	modifying even.	pan-ng	w-dns.log	✓ → PAG ALC NEFW.	IONS Run -
	Comment ®			Help 🖬	IN OUT	6	영 🚺 🛲 🕍 Sei	ect Fields (4 o' 4)
	*** Import	tant - Be mindful of st be done in accord Parser Optional for DNS events:	the Exabean Parsing process ance with the Exabean parsin tree Pent Drops (do not send for parsing)	i as modifying ng process.	1 0 2022-06-14 20542-15-195 -05:00	Event = ( ,rae: (*Dn d*;* aff) = _viee: 10001 = eribl_breate = eribl_preet	<pre>skesponseType1:121, "IP4Records1:151 (88996465231", "anpr:1222.024.47.3", compager -rect arraydes.pred.do.cop.m 0855.666 r:Brask pn.sev(ings bab_atio_atio_beats_BS</pre>	.104.162.166;","ContextThreadI "CwmLHecords":"geo.prod.do.dsp.tm y.efcrosoft.camj, Show more
•		Unio (5-8) Brop for known -	Domo frillamente "warwd"		2 \$ 2022-06-14 20.54.16.586 -05.02	: : .rset (*Dn d*t* x*t*) : .time: 165525 : cribi_breake	uRoupenseType's"2","1240econds";"52 75827182685"","scp":"84.140.8.197", 8","event_platform":"85s","DualRegu 58355.546 r:Break on newlines	.184.212.181;","ContextThreadD "QueryStatus":"O","InterfaceInde est":"1","EvenDriShowmore
•	Circa	Cleanup - Pass only , raw	leenup - Pass only , row and child pipe (fields. Thet's st stal) trial			: cribl_pipe:F c () raw: ("Dn Diss exag r _time: 10001	Palo_Alto_MorW_DAS SResponselype"1"2","LPARecords":"20 (983815","atp":"222.192.47.5","Came er:met;","QueryStatus":"6","Lite/fe 59353.585	.72.295.199;","ContextThreadId":"B Decords":"settingsfd-geo.trafficm ceindex"."3" Show more
						a aribl_breake a cribl_breake	ribras av sevines Pala_Alta_NGFW_DWS	

Palo Alto DNS Pipeline, from Cribl Pack - Functions and comments

There, the first (yellow or shaded) comment reminds us to proceed with caution, to make sure we don't violate Exabeam parsing conventions. This is why (if you look at the toggles on the left) this Pipeline comes out of the Pack with all Functions disabled by default. This ensures that we're simply passing the original events directly to Exabeam, as a bump in the wire. From this baseline, you can now choose which use cases you'd like to enable for the event drops you need.

#### SAMPLING PIPELINES' EFFECTS

Notice that in the **Simple Preview** pane on the right, we've preloaded a pan-ngfw-dns.log sample data file. By previewing the Pipeline's changes to this sample data, we can quantify individual Pipeline Functions' transformations before we commit any changes into production. Cribl Packs include similar sample data corresponding to every Pipeline. You can also use Cribl Stream's Capture feature to store your own sample data files, to rerun data and test your Pipeline's transformations as you build it out.

## PARSE, DROP, EVAL

In the same Palo\_Alto\_NGFW\_DNS Pipeline shown above, the optional Drop Functions – which drop uninteresting events – will usually be preceded by a Parser Function that captures relevant context from the original event. Also, the Drops will usually be *followed* by an Eval Function, to clean up this context, so that we're left with the original \_raw field.

If you enable the Parser Function, it will extract the request's domain name resolution into a field called **DomainName**.

The first Drop Function drops Web Proxy Auto-Discovery (WPAD)-related events. WPAD is a browser proxy discovery process that most organizations will elect to drop, because it is uninteresting for security investigations. As you can see, to drop these in Stream, we've just configured a Drop Function to match events that reveal DomainName=='wpad'.

The second *group* (collapsed) of four Drop Functions drops *well-known* domains, by rank.

#### VARIATIONS

Many more interesting security-related use cases are built into the Exabeam Pack for Palo Alto, as well as into other Exabeam-specific Packs available in the Cribl Packs Dispensary.

#### WHAT'S NEXT?

Cribl Stream's capabilities are always evolving, and you might see some differences from the UI screenshots and instructions provided in this edition. Luckily, Cribl Packs keep evolving, too – expect the Pack illustrated here to be progressively updated with new capabilities. (If you've customized the Pack's Pipelines, Stream ensures that importing an upgraded Pack won't overwrite them. See docs.cribl.io/stream/packs/#upgrading).





SECTION 02

## Enrich, Observe, Analyze, Respond

# 03 | Better Threat Hunting with Cribl Search Data Enrichment

Crowdsourcing API Endpoints and Lookups to Thwart Evildoers

by Raanan Dagan, Cribl

#### PROBLEM

To detect anomalies most effectively, SIEM (security information and event management) systems need log events to arrive enriched with ample, and up-to-date, contextual information. Maintaining and updating this enrichment data can be cumbersome.

#### SOLUTION

Cribl Search facilitates enrichment by enabling you to uncover and analyze data at rest, directly from its source. Cribl Search can easily reach out and query data already collected in Amazon S3 buckets (or S3-compatible stores), Amazon Security Lake, Azure Blob Storage, Cribl Lake, Google Cloud Storage, and more. Cribl Search's externaldata feature and lookup operators help you keep your lookup table up to date to best serve your needs. This means that you can run concurrent searches, using that same up-to-date lookup table, with minimal effort.

#### WHY CONSIDER THIS APPROACH?

By searching data where it lives, Cribl Search can dramatically speed up your search process by avoiding the need to move data before analyzing it. Search-in-place is useful in keeping threat intelligence current, and can also accelerate incident investigations and response.

#### HANDS-ON RESOURCES

To apply the integration laid out here, you'll need a GreyNoise account and API key.

To access Cribl Search, sign up for a free Cribl.Cloud account at cribl.io/solved. Search will be ready to use immediately – start with its sample datasets or import your own working data. You'll also immediately have free instances of Cribl Lake, Cribl Stream, and Cribl Edge. When you need more search frequency or data capacity, you can readily upgrade to a paid account.

For a guided tutorial on Search basics, take our free **Cribl Search** Sandbox for a spin at sandbox.cribl.io/course/overview-search.

#### SCENARIO

In this example, Cribl Search uses a local lookup to enrich VPC Flow Logs with security classification data from the GreyNoise threat intelligence APIs. Search isolates events and IPs that GreyNoise classified as malicious, and sends them onward to a SIEM and security analytics tool.

α <b>classification</b> Type: <b>string</b>   Uniques: <b>3</b>   Presence: <b>100%</b>			
unknown malicious benign	828 170 2	17.00%	82.80%

Isolating events classified as malicious

#### WHAT WE'LL DO

We'll cover the following techniques:

- Ingest threat intelligence data to Cribl Search from an API endpoint.
- Use Cribl Search operators to isolate potential IOCs (indicators of compromise).
- Export the results to a lookup table.
- Configure Cribl Search to update the lookup table.
- Set the whole ingest/update cycle to repeat on a defined schedule.

#### NOTABLE SEARCH OPERATORS FOR ENRICHMENT

Let's dive into how this threat-hunting approach works. We'll use two main Cribl Search operators for enrichment, and we'll show you how to apply them in the steps below :

- The externaldata operator (docs.cribl.io/search/externaldata) fetches data from HTTP(S) URLs – including, as we'll apply it here, public APIs.
- The lookup operator (docs.cribl.io/search/lookup) retrieves fields from lookup tables, returning all rows by default.

#### VERIFYING GREYNOISE API ACCESS

We can first test the GreyNoise API, independent of Cribl Search. This curl command will validate that all is working:

```
curl --request GET \
--url
'https://api.greynoise.io/v2/experimental/gnql?query='last_
seen:1d'&size=1000 ' \
--header 'accept: application/json' \
--header 'key: greynoise_enterprise_key'
```

#### INGESTING THE ENRICHMENT DATA INTO CRIBL SEARCH

Once API access is verified, we can assemble the Cribl Search query below to pull fresh indicators from GreyNoise threat intelligence APIs, isolate potentially malicious IP addresses, and stage these indicators for lookup.

The first step is to pull in data from GreyNoise threat intelligence APIs, using Cribl Search's externaldata operator. We can accomplish this with an initial query like this:

```
externaldata
[
"https://api.greynoise.io/v2/experimental/gnql?query=last_
seen:1d&size=1000" ]
with( headers='{"accept":"application/json","key":
"greynoise_enterprise_key"}', dataField="data")
```

Running that query gives us results like those shown below. Next, we're interested in the GreyNoise classification field, whose distribution we've zoomed up here from the left nav:

Search History (45)	Seed	165												Sampling 5	0.61	tere   Local	SEARCH
<ul> <li>3 externelidate</li> <li>2 i "https://a</li> <li>3 withi header</li> </ul>	61-8199 8-19700	toise:lo/w2/er	perimental/g	ngligueryrlast.) Neyfi "greynold	séer i bolista e_erterpris	e-1000* ] c.kcy*)*, c	atof Leld-"data										-
Svenis Rola Cha	1													100	0 (1996)/85 (3.4	210 0-0.00	Actions V.
¥.		100															
account Name	1.11	105															
actor 0	1	100															
D bet	1		2023-46-3	Prebelosi com		a	pier beibeibe	X15 V-0	20-00100	politi koristi politi	ine boost	ale f estastas	apprint of allowing	2079-16-11 00 0	100	sind to bring to de-b	
a desidention	2	in all state lifes						1	And a local					Read advert	handers		
D dve	10	O CLASSICICS	reion						theater					TAL SOL	surveys .		
a dutafield		Type: string	Uniques: 2	Presence: 100%					PRT241/14	or Backwares	11/17/43 Del - Me	od sr'i fedri des	"Jacont_been; somethe	-1004 Peri-95-8	L.WCMDC.	Capperson inter	1100.1.44
a dataset		unknown.			471	_	57,12%	**	e respective	t grayerise. 5	s/waylexperim	sursy Mudy Jam	ryslast_sameranistan	*1004 2012-08-12	C.accale.	"application	lites, sec
a anaype	1	mations			325	1	.92%	**	tetpe://ie	rt. grayestur, b	()+2/at per ca	ertal/graph report	eystast_seerindestre		Caccade.	"application,	ipsort, the
and point lane		antiers.	station'	-	M	2913		mulic	e retint/re	ri grupertse.i	n/w2/avgarim	intal/graph/spor	ryslast, size: Mistie	-1008 2021-82-0	("accept"	*application	13400°,"NE
Twit block	541	default	stream	(rener)	- 61	1974	antarnal.meta	<b>FPUTHC</b>	PREMINITA	i Bachelites's	ju2/aspartm	enal/grailique	ryslast_been lokston	-1008 2023-05-8	1"accept"	"Application.	(3107,"NE"
D Perinters	1	default	whier over:	maticious	10	data -	wramalstria	<b>Trilinic</b>	e retain//a	i.graynatia.5	n/w2/wappering	intal/grafiles	ry-Last_sees lieks tee	-1208 2022-04-12	-incoder	Application.	juint, the
a headers aroupt		default	warm	whereart	U	4114	esternalizeta	*stime:	. Peteri/la	rt. preprotos. 5	Unit/sciper tw	enal/grafiter	ry'Last_seer lieks fire	-1000 2021-05-1	L"MORE	Tage Treatment	istort, the
The Res Arts	2	default	universe.	street	£)	data	external data	Tilbio	* ******//*	i preprotie.5	./ +2/1×200-100	ental/graphiese	ry-Last_sees laster	-1006 2022-04-0	("sister"	· application	ison me
headers Uter Agont	1	default	winaw	unknewn	10	4214	external.data	Vallase	. eetpar/ra	rt.greynstae.b	J+2/experte	-tal/gratique	ry Last_seen Sills fae	-1008 2623-07-0	L"actual"	Tep: Nextfor	124007,794
o heat		default	articest)	arkener.	U	data	externaldeta	fallier	+ enter//a	f. preparise. 5	July anger in	-tal/geal/res	ry-Lost_sees likelas	-1008 2021-06-0	- L*scool*	Application,	Seet, Ter
	1000	autorite.	-					millioned		a presenter h	1-11-11-11-11-1	and all time all firms	mail and some it distant			Anna Charles Street	Acres Trees

GreyNoise data ingested via Cribl Search's externaldata operator

## ISOLATING IOCS AND NOURISHING A LOOKUP TABLE

Now that we've examined the raw GreyNoise ingest, we can build out the query to filter IP addresses of interest, and to store them to a lookup table. We'll add the following steps:

- Use the project operator to extract only the IP (ip) and classification fields.
- Use the where operator to consider only malicious IPs.
- Export those results to a local lookup table named badguys.csv.
- Use the fieldMapping option to map fields to lookup table columns.
- Use the mode=overwrite option to update the lookup file.

The full Cribl Search query looks like this:



Here's how this looks in the Cribl Search UI:

Search History (60) Sar	ved (6)				Sampling	H C Thom
<ul> <li>i externaldata</li> <li>2 [ "https://api.pr</li> <li>3 with headerse'("</li> <li>4   project classif</li> <li>5   export descript</li> </ul>	reynolis.lo/v2/experiment "accept":"application/jise fication, 1p   where clas Lize="Bod Guys" fieldMapp	al/gnqlhquery=last_seen:ldisize=1000" ) n","sey": "greynoise_enterprise_key")", data sification="malicious" ing="classification:ij_status,ip:source_ndde	Fields"cata") tss" to lackup budgeys mode-overneri	æ		
Events Fields Chart						1 results (2.5
7	4					
IockupFile totalBytesOvt	1 2 1					
EstatéventsCropped	1 pitti-0.0 to to state	data-na ci casa da constanto interna da casa da constanto	and a state of a state of a state	colasias di la quanta da la calenda	and change and an	2012 YO LA SHADE
totaljventsCut	2 Inskupilite	totalByresCut	tistalFventsDrop	pad	totally	tuCetree
			0			
	badgsys.csv	7132	.0		291	

Updating the lookup file

## ONCE IS NOT ENOUGH

Once we're satisfied with our query and its initial results, we can save it by selecting **Actions** > **Save**.

From the resulting **Save Search** modal, we can select the **Schedule** left tab to schedule the query to run automatically on a reasonable interval, like once a day.

Save Search			
Name* ()			
greynoise			
ID: greynaise 🗍			
Description			
Ingest GreyNoise D	finait data, save to lookup table		
Results Access	Private 💽 All Members		
Search String	Run on schedule		
Schedule	Every day at 0 i D Reset	Timezone UTC -	
Notifications	OR		
Advanced	Cron 0.0***		
	Sommand Someble: Tuesday, 9 April 2024 at 6000000 UTC Wedeneday, 10 April 2024 at 600000 UTC Thursday, 11 April 2024 at 60 0000 UTC		
			Cancel Stee

Scheduling the query to run periodically

#### LOOKUP TABLE NOW ISOLATES ONLY THE BADDIES

With all this in place, any query against this lookup file will now find only the malicious IPs reported in the last 24 hours.

To verify the table's contents, we can select **Data** > **Lookups** from Cribl Search's top nav, and then click on the new or updated lookup file. Notice the column names mapped from GreyNoise: ip\_status, source\_address.

Data - lookups badguys.csv			
Riename*	norpops		
Description	Bad Guyd		
Tags 🔿	Enter tags		
Edit Mode Table Text	A set commos		
ŵ		ip_status.	source_address
÷		Politonal .	(TLAN 62154)
8		malicina	48.78.78.124
¥.		Hattyteve:	197.188.68.79
A.		relines.	18117.68.256

Lookup file's structure

#### ENRICHING LOG DATA WITH IOCS

Now that we know we can rely on this periodically refreshed lookup table to identify recently reported malicious IPs, we can enrich VPC Flow Logs data from our badguys.csv lookup.

Why not start with sample data? Cribl Search's out-of-the-box cribl\_search\_sample dataset includes AWS S3 VPC Flow Logs data. This data contains source address IPs (srcaddr field) that might prove to be malicious.

Here's a query that isolates this datatype, and maps its srcaddr field to the lookup table's source\_address column:

```
dataset="cribl_search_sample" datatype="aws_vpcflow" | limit 10000
| lookup badguys on srcaddr=source_address
```

After running this search, we can check the <code>ip\_status</code> field from the left nav for malicious values.

Search History (6	54) Saved	(6)								
<ul><li>1 dataset="</li></ul>	cribl_searc	h_sample" data	type=":	aws_vpc1	flow"   limit	10000   lookup bad	guys on srca	ddr=source,	addres	15
Events Fields	Chart									
7 a account_id a accion	1	1.2% 912.0 608.0 304.0								
a bytes	23	2023-10-10 1	1.50,00	202	3-10-10 11 55.00	2022-10-10 12:00:00	2023-10	10 12:05:00	20	073-10-
a dataset	1	@ account_id	action	bytes	dataSource	dataset	datatype	dstaddr	ds	tport
a dataSource	1	123456789015	REJECT	40	vpcflowlogs	cribl_search_sample	aws_vpcflow	10.0.0.138	10	545
a dstaddr	10	12345678901:	REJECT	40.	vpcflowlogs	cribl_search_sample	aws_vpcflow	10.0.0.138	55	55
a dstport	54	123456789013	REJECT	40	vpcflowlogs	cribl_search_sample	aws_vpcflow	10.0.0.138	85	51
a end	2547	123456789013	REJECT	40	vpcflowlogs	cribl_search_sample	aws_vpcflow	10.0.0.138	30	002
a host	15	123456789013	REJECT	40	vpcflowlogs	cribl_search_sample	aws_vpcflow	10.0.0,138	58	546
a interface_id	4	12345678901:	ACCEPT	1297	vpcflowlogs	cribl_search_sample	aws_vpcflow	54.239.152	.25 44	3
ip_status	1	a la statur							17	96
a log_status	1	Tan status	Halanta						85	69
a packets	13	Type: string	Unique	s: 1   1	resence: 4%				28	545
a protocol	2	malicious				398 3.98%			31	985
a source	25	12345678901:	ACCEPT	62246	vpcflowlogs	cribl_search_sample	aws_vpcflow	10.0.0.89	54	560

Isolating malicious events

To take this a couple of steps further, we can tune the query to return only the malicious hits, and to summarize these by destination. The full query now looks like this:

```
dataset="cribl_search_sample" datatype="aws_vpcflow" | limit 10000
| lookup badguys on srcaddr=source_address | where ip_status=="malicious"
| summarize count() by dstaddr
```

After running this, we can visualize the data's distribution more clearly. On Cribl Search's **Chart** tab, we can click the **Format** (paintbrush) button, then set the **Type** to a **Donut** chart:

同業単	✓ Chart	Reset X
	Туре*	
	Lul Bar	Q
	🗠 Line	1
	and Bar	
	۲ 🗠 Area	
	I# Single	
	Donut	
	Funnel	
	② Gauge	
	> X-Axis	
	> Y-Axis	
	> Interactions	
	> Legend	

Selecting a different chart type



That will enable us to see the distribution of destinations in a format like this:

Malicious hits only, summarized by destination

## FORWARDING THE DATA

Finally, we can send the events and IPs that GreyNoise classified as malicious to our SIEM and security analytics tool. Cribl Search has both narrowed and enriched the data we're sending through. So we're limiting downstream analysis (and license usage) to high-value events only.

The send operator, by default (when invoked with no parameters), forwards Cribl Search results to a Cribl Stream HTTP Source. This gets the data into Stream, where you can route the results to your SIEM service of choice. Here's how to append send to a Search query:

```
dataset="cribl_search_sample" datatype="aws_vpcflow" | limit 10000
| lookup badguys on srcaddr=source_address
| where ip_status=="malicious" | send
```

#### VARIATIONS

We've seen how Cribl Search's externaldata and lookup operators enable you to automatically query a useful API to keep a lookup table of threat indicators up-to-date. This, in turn, enables you to run multiple searches against current log data and indicators, with minimal effort.

The use case we explored here looks up VPC Flow Logs log data against GreyNoise indicators.

This demonstration is representative of many data types and threat intelligence sources, and provides an easy-to-replicate pattern for your own needs.

#### WHAT'S NEXT?

For details on the Cribl Search and other resources we've applied here, see our online documentation:

- externaldata operator (docs.cribl.io/search/externaldata).
- lookup operator (docs.cribl.io/search/lookup).
- project operator (docs.cribl.io/search/project).
- where operator (docs.cribl.io/search/where).
- export operator (docs.cribl.io/search/export).
- send operator (docs.cribl.io/search/send).
- Other operators (docs.cribl.io/search/operators).
- Scheduled Searches (docs.cribl.io/search/scheduled-searches).

To learn more about GreyNoise threat intelligence APIs, see GreyNoise's own documentation at docs.greynoise.io.



# 04 | If You Can't Search It, Does It Even Exist?

Expediting Investigation and Incident Response with Cribl Search

by Alexandria Crusco, Steven Litras, Ana Monroy Zolano, and Aaron Thummel, Cribl Security Team

#### PROBLEM

**TL;DR:** Attacks are increasing, data is increasing, and separating valid attacks from false positives has become exhausting.

The fight against attackers grows in intensity each day for IT/Security incident first responders, defenders, and employees of organizations. The chances of falling victim to an undetected attack are rising, leaving responders to fight what feels like a losing battle.

According to IBM's *Cost of a Data Breach Report 2023*, data breaches' average cost reached an all-time high in that year of U.S. \$4.45 million. This had increased 2.3% from the 2022 average cost, and increased 15.3% from the 2020 average. As Security professionals, our tooling leaves us with alert fatigue and a false sense of security: We have *all this data*, so we must be catching attacks, *right*?

#### SOLUTION

Cribl Search offers a powerful response to mountains of data, relentless attackers, and security tools that do not always fit every incident scenario.

Especially when used in combination with Cribl Stream and Cribl Lake, Search delivers the next generation of incident investigation. Ultimately, it enables defenders to accelerate their incident response processes, and to reduce attacks' potential impact on their organizations.

Through this chapter's incident scenario, we'll highlight applying these Cribl Search techniques:

- 1. How to build Search queries relevant to your environment (i.e., Search + your SIEM == true power).
- How to build meaningful dashboards (i.e., Search == a powerful enrichment tool).
- How to accelerate incident response (i.e., Search == the GOAT investigation tool).
- 4. Bonus: How to use Cribl Stream to accelerate Search's own performance (i.e., Search + Stream == besties together).

#### WHY CONSIDER THIS APPROACH?

Incident responders and/or security engineers can use these techniques to increase the speed at which they detect, respond, communicate, and remediate potential threats. This approach is powerful in combination with your organization's SIEM, or you can use it by itself. Consider this approach if:

- You currently have an internal struggle with massive amounts of data. The data has subsumed what little time your resources have to triage and correlate data to identify potential threats or incidents.
- Incident investigation is not *simple* within the organization's current processes.
- You want to accelerate incident response, to reduce impacts on your organization.

The attacker scenario laid out below is assumed to be *medium* in complexity. Re-engineer this when considering more complex attack use cases.

This approach relies on the MITRE ATT&CK framework (attack.mitre.org), and will be easy to follow by anyone with a basic understanding of investigation methodology.

## WHAT WE'LL DO

This scenario will reveal Cribl Search applications like detection building, custom dashboarding, enrichment, and searching in place. As you can see from this outline, you'll have to read all the way through to find out whether good triumphs over evil!

- 1. The Calm Before the Attack
- 2. The First Clue...
- 3. First Suspect: AWS
- 4. Next Suspect: Okta
- 5. The Final Suspect: Google
- 6. The End

#### THE CALM BEFORE THE ATTACK

"Detective" Steve Searchwell logged onto his trusty computer for another day of riveting security work. To be clear, he referred to *himself* as "Detective," even though no one else did, and was hoping the name caught on as he made a name for himself as the best security investigator in the industry.

Like most Security professionals, "Detective" Searchwell used a SIEM (specifically, Panther), in combination with Cribl Stream, to help ingest all the needed logs from his organization's many active applications and tools. This setup was working pretty well.

However, "Detective" Searchwell was struggling with the massive amount of alerts the SIEM generated each day. Luckily, Searchwell had configured all his needed datasets in Cribl Search the week prior. He was excited to start using Search for investigative work, in the hope of maturing the detections for his organization's environment, as well as potentially reducing alerts.



"Detective" Searchwell, in plainclothes guise

#### WHY DOES THIS MATTER?

Most Security professionals are similarly bogged down in alerts, noise, and potential rabbit holes. In this instance, Searchwell was able to configure datasets in Search to assist him in maturing his investigation process. This powerful combination of his SIEM and Cribl Search allowed Steve to put to the test the phrase "no one knows your environment better than your Security professionals".

And this is an important note for organizations: The Security personnel tasked to protect the environment every day are going to be the quickest to understand root causes during investigative scenarios. They need the tools to get to the root cause quicker.

## THE FIRST CLUE...

Most security incidents start with a single notification, or seeing something strange in your environment. Searchwell was looking through the alerts in his SIEM, and saw a notification that could potentially be an indicator of compromise (IOC). It looked like some account was trying to explore the AWS environment with an enumeration effort:

	19 60 - 17 Car	nvas Thread 🖸	
Unexpected AVVS Linganization > mill	INATATION		
Runbook: Tip: Try # F	to search this channel ×	Panther Are 10 days ago	
Investigate CloudTrail for other activ	ities by same User Identity that might indicate	Alert Summary	
enumeration.		Unexpected AWS Organization Enumeration	
Severity: 🛑 High	Status: 👏 Open	Runbook:	
View in Panther Set Assignee	✓ Update Status     ✓	Investigate CloudTrail for other activities by sar	
		User identity that might indicate enumeration.	
		Severity: High	
1 reply 10 days ago		Status: No Open	
Panther APP 17:23		View in Panther Set Assigner 👻	
Alert Summary		Update Status ~	
Unexpected AWS Organization Enur	meration		
Runbook:		Panther 40% 10 days and	
Investigate CloudTrail for other activ	vities by same User Identity that might indicate	Alert Metadata	
enumeration.			
Severity:	Status: JOpen	Alert ID: ac2c514ebf1905079252do4a56cbal70	
Manuela Banthas	N- Dashi Datis	Alert Type: Rule	
View in Panoler Set Advignee	· Mounte spilles ·	Detection Name: AWS Organization Enumeration	
		Activity	
		Log types. miste tobarrott (1)	
1 reply 10 days ago			
Panther APE 1901		Description:	
Panther ARE 19:01 Alert Summary		Description:     Alerts if anyone not sanctioned to enumer     things via AS accomplications does so	
Panther APE 19:02 Alert Summary Okta Rate Limit Event: lowtern oper	ation rate limit violation) by lunknown!	Description: Alerts if anyone not sanctioned to enumern things via AMS organizations does so.	
Panther ARE 1901 Alert Summary Okta Rate Limit Event: [system.oper	ation.rate_limit.violation] by [unknown]	Description: Alerts If anyone not sanctioned to enumer things via AWS organizations does so. Alert Context	
1 reply 10 days ago     Anther ARE 19:01     Alert Summary     Okta Rate Limit Event: (system.open     Runbook: (Not Provided)	ation.rate_limit.violation) by (unknown)	Description: Alerts if anyone not sanctioned to enumeri- things via ABS organizations does so. Alert Context	
1 reply 10 days ago     Anther AM 19:01     Alert Summary     Okta Rate Limit Event: [system.open     Runbook: (Not Provided)     Severity: High	ation.rate_limit.violation] by [unknown] Status: 🎒 Open	Description: Alerts if anyone not sanctioned to enumeri things via ABS organizations does so. Alert Context	
1 reply 10 days ago     Alert Summary     Okta Rate Limit Event: [system.oper.     Runbook: (Not Provided).     Severity: High     View in Panther Set Assignce	ation.rate_limit.violation] by [unknown] Status: 🎒 Open ~ Update Status *	Description: Alerts if anyone not sanctioned to enumeric things via ANS organizations does so. Alert Context { *Account*: *Bealon*: us.east.1*	
1 reply 10 days ago     Panther AM 19/01     Alert Summary     Okta Rate Limit Event: [system.oper     Runbook: (Not Provided)     Severity: High     View in Panther Set Assignce	ation.rate_limit.violation] by [unknown] Status: 🎒 Open 🗸   Update Status 🛛 🗶	Description: Alerts if anyone not sanctioned to enumerations via ARS organizations abore so. Alert Context [ "Account": "Region": "us-east-1", "Session": "Session":	
1 reply 10 days age Panther ### 19:01 Alert Summary Okta Rate Limit Event: [system.open Runbook: (Not Provided) Severity: High View in Panther Set Assignce	ation.rate_limit.violation] by [unknown] Status: 🎒 Open	Description: Alerts if anyone not sanctioned to enumeratings via ABS organizations does so. Alert Context	
1 reply 10 days ago     Anther ARE 19:02     Alert Summary     Okta Rate Limit Event: [system.open     Runbook: (Not Provided)     Severity: High     View in Panther     Set Assignce     1	ation.rate_limit.violation] by (unknown) Status: 🍋 Open V Update Status 🗶	Description: Alerts if anyone not sanctioned to enumeri- trings via AMS organizations does so. Alert Context { { * * * * * * * * * * * * *	
1 reply 10 days ago     Panther #80 19:02     Alert Summary     Okta Rate Limit Event: [system.open     Runbook: (Not Provided)     Severity: High     View in Parsther     Set Assignce     1 reply 10 days ago     t reply 10 days ago	ation.rate_limit.violation] by (unknown) Status: 🍋 Open ~ Update Status *	Description: Alerts if anyone not sanctioned to enumeri- things via AMS organizations does so. Alert Context { { *Account*; *Region*; *us-east-1*, *SesionIdentityJam*; *arc:east:lon::	
1 reply 10 days ago Panther AM 19:01 Alert Summary Okta Rate Limit Event: [system.oper Runbook: (Not Provided) Severity: High View in Paother Set Assignce 1 reply 10 days ago 1 reply 10 days ago	ation.rate_limit.violation] by (unknown) Status: 🍋 Open 👻 Update Status 🗶	Description: Alerts if anyone not sanctioned to enumeric things via ANS organizations does so. Alert Context { "Account": "Account": "Account": "Any any account ": "Any any account ": "Any any account account so account	
1 reply 10 days age Panther #M 19:01 Alert Summary Okta Rate Limit Event: [system.open Runbook: (Not Provided) Severity: High View in Paother Set Assignce I reply 10 days ago Escage Suderts-panthet	ation.rate_limit.violation) by (unknown) Status: 🌏 Open 👻 Update Status 🗶	Description: Alerts if anyone not sanctioned to enumeric thrings via ANS organizations does so. Alert Context { "Account": "Region": "us-east-1", "sessionidentityArm": "arm:ressi Lon: Expressions.compactors.com/us-rest- 2/ANSReservedSSO_ANSAdministratorAccess.com TuserIdentityArm": "arm:ressists: = issued suezifementityArm": "arm:ressists: = issued- rels/ANSReservedSSO_ANSAdministratorAccess.com TuserIdentityArm": "arm:ressists: = issued- rels/ANSReservedSSO_ANSAdministratorAccess.com and and any and any	

Enumeration alerts from Panther SIEM

In this case, Searchwell has a detection on his SIEM that alerts him when someone makes certain AWS Organization calls. These calls could indicate that someone, specifically an admin named Ben Wiff, is trying to explore the Organization's structure – accounts and OUs (organizational units). This type of activity is not a common occurrence within Searchwell and Wiff's environment, so Searchwell puts on his detective hat and decides to investigate further. At this point, all we know is the alert. The alert is a start, but we need to look at other things to understand the context in which the activity occurred. Though Searchwell's company uses Panther as its SIEM, this could really be any SIEM, or even the Cribl Search product itself (assuming the company can build its own detections).

## ELIMINATING FALSE POSITIVES

The true art of investigative work is to understand where the truth is – to pull the thread and ask the questions that reveal the root cause of an investigation. As a Security professional, Searchwell knows that there is a chance that this activity might be legitimate in nature.

Searchwell begins by making a list of things his company does that would make this type of alert "legitimate." A number of processes regularly enumerate the AWS environment; these processes would do things like ensure that tagging is up to standard, maintain an active catalog, do maintenance, or even security scanning.

For example, the enumeration detection has logic that ignores detected enumeration from AWS roles that are expected to enumerate. As you can see at the bottom of the Panther screenshot below, 11 roles are in the "is not in" clause. This simply minimizes the noise, to help Searchwell pay attention only to anomalous behavior. It's entirely possible that Ben simply forgot that he's supposed to use one of those roles for that kind of work. But by having those roles as "exceptions," we minimize the noise that this alert could create, and only have to investigate the ones that deviate from policy.

#### 04 | IF YOU CAN'T SEARCH IT, DOES IT EVEN EXIST?

VS Organization Enumeration Activity 2 MMMAGED - ID AWSOrganization Enumeration Activity の	Enabled (on ) .	Downloa	d Tioni
Configure 🗄 Alerts 🛈 Rule Errors			
For the Following Source			
Log front ANS.Cloudinall ×			•
Detect		Rule Bulider	Text Editor
Filter to only include events:			
Where +			
Match on the following rule logic:			
Where eventName is in list (4 values) 🖽 × and • eventSource is	organizations.amazonaws.com ×		
and userIdentity.sessionContext.sessionIssuer.am is not in list (11 values)	× +		

Panther SIEM logic to ignore expected roles

#### UNDERSTANDING YOUR ENVIRONMENT

At this point, it's important for us to remember that Searchwell is a self-proclaimed "Detective" due to his in-depth understanding of his organization's environment. If you ever find yourself in Searchwell's shoes, it's best to remember the following: To be able to properly triage events in your environment, it's critical that you *understand* your environment.

In Searchwell's case, he could easily have written off this activity as "common." However, Searchwell hones in on this alert, because he's noted that an administrator is not using any of the 11 roles that legitimately make organization enumeration calls.
#### FIRST SUSPECT: AWS

Now that Searchwell has determined things are definitely amok, he must use some of the tools in his belt to glean more information about Ben Wiff's activity. At this point, Searchwell turns to his newly configured Cribl Search tool, and begins to ask himself questions about the enumeration alert. What would he like his first query in Search to be?

Searchwell knows that there are many different reasons why an attacker might want to enumerate an AWS environment. For the sake of being thorough, he begins to write down his hypothesis. If this is an actual attack, an attacker might enumerate an environment to list out OUs and accounts, to get the lay of the land or the attack surface (like our alert example above).

Searchwell continues pondering the enumeration and finds himself wondering, "Now, what would an attacker do next?" This thought leads to Searchwell's next step: After the enumeration, did Ben Wiff's account attempt to connect to each account for the purposes of testing the user's access? This question leads us into Searchwell's first Search query.

#### Sidenote: Start with What You Know

Cribl Search is a powerful tool, and there are many times that understanding what you are looking for can help you get the exact answer you need through your Search query. There are times, however, when you might not have as educated a guess as we are outlining here. In which case, you could simplify your search, by starting with what you know instead of with the hypothesis you want to test. This could be as simple as writing a Search query that goes something like this:

dataset="your\_dataset" "beowiff@cribl.io"

In this instance, the dataset would be where your AWS CloudTrail logs are being stored. (This is assuming that you've added these logs' location to your Cribl Search instance as a dataset. See the end of this chapter for links to Search guides that cover this and other techniques.) Searchwell ran this query, and got a lot of results:

Search Ξ Home Di	ata Dashboards Settings					0 <b>°</b> 19	<b>B</b>
Search History (263) Saved	(32)	Sampling ~ 🕒 2024-03-24 10:50:03 - 2024-03-28 10:50:11 Local			SEARCH		
<pre>③ 1 dataset="cloudtrail_"</pre>	lmo" "beowiff@cribl.io"						
Events Fields Chart		Ran	on: 2024-04-05 08:21:37	Objects discovered:	496683, scanned: 49668	3 721 results Details	Actions v
Image: Second Line     Image: Second Line       Image: Second Line     Image: Second Line       Image: Second Line     Second Line	2111 528 7000 1775		71				
a additi enticationMethod 2 additionesTransferredIn 1	0 2024-03-24-04.03000	pages called on no call Event	1h @ Mar 26, 2	024 09:00:00 (di balia)	2014-00-20-02-02-02	2004-00-0514-00-00	
additio, sTransferredOut 77 additionalE, CipherSuite 2	> 2024-03-26710:04:06.000	"("host":"			","source":"A	WSLogs/o-fq4m8rmcz9/ WSLogs/o-fq4m8rmcz9/	/CL /cL
additionalEvta.MFAUsed	> 2024-03-26710:01:04.000	"("host":"			","source":"A	WSLogs/o-fq4m8rmcz9/ WSLogs/o-fq4m8rmcz9/	/cl
a addition "x-amz-id-2"] 544	2024-03-26710:01:03.000	"("host":"			","source":"A	wSLogs/o-fq4m8rmcz9/	/c1.
o data_raw 715	2024-03-26710:00:56.000	"("host":"			","source":"A	WSLogs/o-fq4m8rmcz9/	/ct
a dataset 1 a datatype 1	> 2024-03-26710:00:11.000	"("host":"			","source":"A	WSLogs/o-fq4m8rmcz9/	/ct
errorCode 6	> 2024-03-26709:59:32.000	.(			","source":"A	WSLogs/o-fq4n8rmcz9/	/ct

Initial search results from CloudTrail logs

#### OH, WE BE QUERYING, WE BE QUERYING

Searchwell started searching across CloudTrail logs to examine the suspected "attacker's" actions on AWS in greater depth. What had the user's account been doing, in what AWS accounts? The easiest way to do this was to summarize the data by a few fields:

- 1. recipientAccountId the account where the action happened.
- 2. eventSource the AWS service invoked.
- 3. eventName the action taken.

To do this, Searchwell appended the following to his query:

| summarize Count=count() by recipientAccountId, eventSource, eventName

And the results were a bit easier to work through:

🧿 Search ☴ Home Data	Dashboards Settings		c* 2 🖷 😒
Search History (261) Saved (32)		Sampling. V	50:03 - 2024-03-28 10:50:11 Local SEARCH
<pre>③ 1 dataset="cloudtrail_ime" ③</pre>	"beswiff@crimi.io"   summarize Countwcount() by reci	sientAccountId, eventSource, eventNam	4
Clarel Clarel 107	Ran bit: 2024-04-05	0.05.29 Objects discovered: 496083, science	s: 499683, 95 results Detain 🖬 Actions -
a'			Course
recipientAccountid	eventSource	eventName	Count
- manufacture -	ec2.anazonaws.com	DescribeRegions	2
and the second s	servicecatalog-appregistry.amazonaws.com	ListApplications	1
- and the second se	securityhub.amazonaws.com	DescribeMub	1
	notifications.amazonaws.com	ListNotificationHubs	2
(Appendix of the second	s3, anazonaws, com	GetBucketPublicAccessBlock	31
-	s3.anazonaws.com	GetBucketAcl	31
	s3.anazonaws.com	GetBucketPolicyStatus	31
and the second s	s3.anazonaws.com	GetAccountPublicAccessBlock	6
	s3,amazonaws,com	ListAccessPoints	81
	s3.amazonawis.com	HeadBucket	1
	s3.anazonaws.con	ListBuckets	2
and the second s	s3.amazonaws.com	GetStorageLensDashboardDataIntern	al 2
	s3.asazonaws.com	GetStorageLensConfiguration	2

Search results after summarization

As Searchwell is reading through the results, he starts to have a suspicious feeling. Ben Wiff is a highly technical employee of his organization, with a very solid understanding of the environment. But Searchwell sees that Ben Wiff was doing some serious clicking through S3, revealed by multiple counts on eventNames (actions) like: GetBucketAcL, GetBucketPublicAccessBlock, and more. Why?

Searchwell's instincts are starting to kick in. This behavior is more typical of an attacker attempting to map an attack surface, than of Ben Wiff's normal day-to-day activity. Alas, Searchwell must not get ahead of himself. He scrolls further down through the results.

As he enumerates the same query, Searchwell notices that the actions taken by beowiff@cribl.io's account were a bit more widespread than just listing S3 buckets' attributes. Our potential attacker had logged into all of the accounts that Ben has access to, and had clearly explored each.

A couple of things jumped out to Searchwell as potential problems: DescribeOrganization and ListRoots are very alarming enumeration methods of the most powerful AWS accounts.

💽 Search 🗏 Home Data Dashboards Setting	5		6 P 🖯 🥵
Search History (257) Saved (32)		Sampling ~	00 2024-03-24 10:50:03 - 2024-03-28 10:50:11 Local SEARCH
<pre>\$ 1 dataset="cloudtrail_Ins" "beowiffgcribl, in 0</pre>	*   summarize Countercount() by recipie	entAccountId, eventSource, eventName	
Events Fields Chart		Ramore: 2024-04-05 10:05:29 Objects discovere	d 496683, scarned 496683 95 results Details 🖬 Actions 🤉
			- : #
21			
			E Dour?
© recipientAccountid	eventSource	eventName	Count
ALCONOMIC .	ce.amazonaws.com	GetAnomalyMonitors	1
and the second s	ce.anazorians.com	GetCostCategories	4
ALCONO DES	ce.anazonaes.com	GetReservationUtilization	4
August Street.	ite-anazonaws-com	GetAnomalies	5
attact the	ce.amazonaws.com	Gettage	4.
ACCOUNT OF A	organizations.amazonaws.com	DescribeOrganization	+
Annual Annual Contraction of Contrac	organizations.amazonams.com	ListRoots	4
and the second s	arganizations, seazonews, con	115TOrganizationalUnitsformare	1
	organizations.anazonaws.com	ListCreateAccountStatus	4
ALL	organizations.amazonaws.com	ListAccountsforParant	3
And the second s	organizations.anazonaws.com	DescribeAccount	1
warman (1955)	organizations.amazonees.com	DescribeResourcePolicy	1
Manual Man	organizations.amazonaws.com	ListPolicies	1
account on the	organizations.andzonawt.com	DescribeRolicy	

Summarized search results, showing DescribeOrganization and ListRoots actions

Searchwell pauses. It is clear that he must go figure out if there was foul play on Ben Wiff's account, so he heads to his next suspect to build his case.

#### NEXT SUSPECT: OKTA

Searchwell clearly now understands the magnitude of the investigation he is analyzing. Most likely, an attacker was able to compromise the user's account. Upon compromise, the attacker then went through and identified valuable information about the environment. Searchwell knows that he now must be able to confirm the attacker's point of entrance, so he turns his attention to his organization's identity provider, Okta.

There are many ways to confirm whether a user's Okta account has been compromised. However, the first thing that Searchwell decides to do is look for failed login attempts. The first query he writes in Cribl Search looks like this:

```
dataset="your_dataset_okta_logs" actor.displayName="Ben Wiff"
```

Searchwell sets the appropriate time range in Search's time picker, and hits **Search**. This returns a lot of information. Searchwell combs through the 53 results, which look something like this:

Search History (175) Save	ed (32)	Sampling ~	2024-03-25 09-28:36 - 2024-03-29 09:28:39 Local SEARCH
③ 1 dataset="51 ③	, actor.displayNam	e="Ben Wiff"	
Events Fields Chart		Objects discovered: 49	927, scanned: 49927 53 results (43.30s) Details 🛂 Actions 🗸
a ["detail-type"]	36 37 1 20 10		· · · · · · · · · · · · · · · · · · ·
• actor	1 B	2024-03-05 13	20 00 2024-03-26 14 02:00
actor.alternateld     actor.detailEntry	1 © Time Local ∨ Event 1 > 2024-03-26T14:32:56.398		',"source":"aws.partner/okta
a actor.id	2024-03-26714:32:40.903		',"source":"aws.partner/okta
a actor.type	<pre>&gt; 2024-03-26T14:32:39.830 c &gt; 2024-03-26T13:11:22.402</pre>		',"source":"aws.partner/okta ',"source":"aws.partner/okta
authenticnticationProvider	2 > 2024-03-26T13:10:59.609		',"source":"aws.partner/okta
# authentic thenticationStep	1 2024-03-26713:10:58.597		',"source":"aws.partner/okta-
authentic edentialProvider     authenticaticredentialType	3 > 2024-03-26T13:10:56.351 2 > 2024-03-26T13:10:55.109		',"source":"aws.partner/okta '."source":"aws.partner/okta
a authenticatternalSessionId	2 > 2024-03-26713:10:44.835		',"source":"aws.partner/okta

Initially retrieved Okta logs

As he is reviewing the results, he identifies his next point of interest: the target display name field. Searchwell feels like he is on to something, and starts to refine his Search query so it looks something like this:

```
dataset="your_dataset_okta_logs" actor.displayName="Ben Wiff"
target.0.displayName="Password"
```

After running this search with the same time range, Searchwell begins to see where things went awry. This search narrows down his results to 11, and Searchwell finally finds his sought-after needle in the haystack. He hones in on the field outcome reason, and refines his search once more so that it looks something like this:

```
dataset="your_dataset_okta_logs" actor.displayName="Ben Wiff"
target.0.displayName="Password" outcome.reason="INVALID_CREDENTIALS"
```

*Yes!* Searchwell finally is able to see the seven invalid logins from the "attacker," which are then followed by a successful authentication.



Failed login event

This now confirms for Searchwell that the attacker most likely gained access via Okta, which leads to the next step in his investigation. He must see what other things the attacker did upon gaining access to this user's account, and then build a foolproof case against the initial access point.

# Sidenote: A Viable Attack Path, or Someone Who Can't Type Their Password?

There are plenty of reasons for seven unsuccessful login attempts. Maybe someone hasn't had their coffee yet, and types their password wrong seven times. That's possible. However, when we are looking at the context of the scenario we are running through, this is a very interesting thread to pull on.

Incident investigation is about finding *viable* threads to investigate further. As the "best" Detective, Searchwell would not just stop at his discovery of the failed attempts on AWS followed by a successful authentication, and call the investigation "closed." He would now take these things and look at the next step – more clues to confirm or dismiss his hypothesis.

# THE HOLISTIC VIEW - OKTA

With a good lead to follow, Searchwell now wants to save some of his more in-depth queries into a dashboard. This dashboard should help solve the case and determine the root cause of this investigation.

After some organizing and analysis of Okta logs, Searchwell will end up with this dashboard. (We'll show you below how he put it together.) It displays a map and three tables.

Okta investigation										XT
Uper		1								
and a local design of the										
Vito										
								and the second		Ĩ
theorem and location						,				1
Lane .	arter alternation	there user Apres	case.	card propter	time page.	tart populate	-	belagCarrier amogCata ankayory		
2228-22.22728141.437	Annal Property A. St.	17 commission database	me. current	Same file	New Dorlands	Bitting Thinks		the last initiation within, he belied walling, his thermitter, his higher	WITTEL ON CAMPACINETING, but	
	Secultrair (st. ta	I'm menter karris'	-	Derses 63.01	Aug Derster	striked blates	dini!	the investigation willing, we bering follow, an it-willing, her high-	CONTRACT, WWW CONCERPTING AND	
3454-89-24158144-257	sentitional.m	(fesetserkpert)	THE OWNER	San Referi	datificenta	artitud States	-14011	they destructed annual Allen. we become Abilitial, we completing, we state	widdling, we controchilation, we	
2834-82-26738-82-882	manarif/parcal.va	Penettartgart	· ····································	Jarany City	Tim Jacsoy	Strikes Skines	- attact	for ten-initia-POLITIN, we be im POLITIN, he IP-POLITIN, we State-	SHITTH, We Darty Mintten, We	
1024-02-241201411962	securities lat, to	("riseliser Agent."	THE CHINE	Jervey City	Ame Detary	Britley States	#2.54 <sup>1</sup>	(New ServiceLation PROSTER), New Device POSITIVE, New Diversities, New States	SHITLE, WE CONTRACTOR, WE	
2624-42-211261411982	beow1778cr181.10-	("ravitainkant"	THE ENTITE	Jersey City	New Jerste	Brited States	17211	New Ser-Lacation-MSEPIN. We Device-MSEPINE New CP-PISITER. New State-	POSITINE, New COUNTY-MEDALINE, New	
2824-10-217361121092	beent transfer.tol.tol	(*reducing)	-	Series City	Nor Deraity	B-Har States	-11111	(New Tele-Leventhin-MOLIPIPE, New Derice ACRIPINE, New DirPOLIPIPE, New State-	NULTUR, New Country-MEGATIVE, You	
2884-02-20106-02-202	level therisking	Provident April 1	THE EXTERN	Jerses City	Sun Jarary	arrived depices -	47947	Das for seation-Milling, we bering Milling, tes D-Milling, we have	PERITER, we Courty-MEGATINE, W	
21024-02-091001412022	desertmental.ra	(frammer spect	The series	HEREY LTD.	NOR JUSTIC	arrise insets	4700	the And-Lacation-Multin, we percertainly the impointer, we state	enlifet, es lartry-musine, ye	
Google Drive Loga										
al line	atter,	Isan	event.	mental. re				id, applicate	charat ipadetes	
2028-02-20120140123.40	tot beyond	Mparties, no.	Life.	actesia (1)	Sumit, "getting	weater, "needed.o	PLINET,	(*isse*s*b(thatte*,*bis)value*skise),(*isse*s*secer_is_bised_drive_drive	06.210.28.303	
2024-02-20120145100-10	nit beneti	Merrill. In	ster	mores 17	same" /"ge lines y	week', "instable	ritreet.	("need") "Littlatte", "sortaater") (roe), ("need") "sever., "s, shared, inter at two	26.2+0.25.400	
3424-43-2418/45-94 M	tiz benefit	marial in	1789	action 10	similar princip	wiene", "the Trial of	direasi.	("name") "bollahla", "moolaalae") trans"; ("mame"; "name"; "a, shared, erias, derias	Tel. 5+0.25.385	
2424-62-21718141-62.00	times 10	et parallela	wim	action IC	age inprinces	availt, "Includes	distai),	("same"s "Billabla", "Box Tallas" (Eran), ("same"s "serier, sa, shared, drive deton	06.110.13.10	
2824-92-24728193-25.85	UK UK UK	Mart Spir 14	edit:	-M1440 II.	Sami' / printry	unerch disettation	(*) (#344)	("new"""httlabte","postkalor")(rise), ("new") "sene", is shared able derive	294,2+8,53,587	
2629-43-26126155123.00	tread 11	P##cr101.10	100	access 11	Sume". Portieury	event","becaralo	rt: talkes)	("name") "billiable", "boolastie":true), ("name": "swmer, til shared driv drive	101.210.20.20	
2824-83-247881681982.48	str birwit	Mark SK. No.	inter.	acress 11	haisenin' i fajoritatura	property, Property,	e*(trob)	("same">"(()Latter, "minimal.ae"(Krse), ("same")"server_16_Attred_server.de")ve	106. 24d.54.581	
101-10-22146140-23.34	67 berwit	Mierbiche	view	access 10	sum's printry	alahfand", "mainala	e <sup>a</sup> strait),	("none"s"billable"s "hostinal or"stron) ("none"s "nover_in_harmed, drive drive	96.2+6.65.88s	
2024-02-24 Tantage 26. 24	DEDR'T	rtikirior9	1789	score 10	ma,buimuh	went", "berrelu	PTI SELET.	("name"1"b11Labla", "Bob1haLae"1c11ae), ("name"1"bener, "to_shared, ortive-driver	06.244.13.303	
3(24-97-24714) Minhel 40	21 heavil	Married via	stes.	actess IC	aune" ("princip	provint Phanking in	rtitrie),	("naar") "In [Eable", "Realistice") ("naar") "narar", 'n Jahreef, drive de he	BL. (4.5, 11, 18)	
Application signings										
Setu:			arger, 5, slipts	ly filme				digity/Mecoge	activ_alteriated	
2009-02-20100102-002			RLA DALVICA	rá				eren, syndyn aldte om 25 meter	searintecrist. la	
2026-01-21200 M (167			NT. LAW LAWY	FTTS Cartan				concretely relation as the	header ( fight - this . In	
2894-49-33,081-69-943			Arts Southeast	ed.				liber bingle cige as to app.	Seper Piller 18,5, 54	
3834-43-28137-14-082			anter .					Sime: alight eight an An age	letoel *Apo: (85, 54	

"Detective" Searchwell's very helpful dashboard

At a glance, Searchwell is able to use this Cribl Search dashboard to paint a picture of the initial compromise through the map. He sees simultaneous logins in different states, better known as "geographically improbable" alerts.

Simultaneously, Searchwell wants the option to disable whichever other applications the "attacker" browsed to after the initial compromise. The second table below the map shows events on Google Drive, which we'll get to below in (spoiler alert:) "The Final Suspect: Google."

The last table shows that the "attacker" looked across the Okta Dashboard, the AWS Identity Center, and Looker – in all, actions better known as "application sign-ons." All of this information is very valuable to Searchwell for a few reasons:

- Searchwell can continue to narrow down his searches to focus on pertinent information about what actions the attacker might have taken.
- Searchwell can communicate with other members of his team, or his superiors, about the scope of the incident.

# GEOGRAPHIC QUERY RECIPE

But first, let's take a deeper dive into how to write a "geographically improbable" query and then save it as a map, as displayed above. The query looks like this:

```
dataset="your_dataset_okta_logs"| where actor.alternateId == "beowiff@
cribl.io" and eventType=="user.session.start"| summarize Count=count() by
client.geographicalContext.state
```

This query essentially looks for the actor ID beowiff@cribl.io, and for sessions' event type, while summarizing the count for different geographic regions – specifically, by state. Put simply: Tell us how many times a session was started in different states. This allows Searchwell to understand whether there is any anomalous activity.

This is especially interesting for Searchwell, as it confirms that this user's account was compromised. The several failed login attempts were followed by a session starting in a location that is not typical for this user. When you hit **Search** on this query, with the correct time range input, you get results like those shown below.



Suspiciously bicoastal logins

#### DEFINING A MAP

On his way to the dashboard we showed above: Now that Searchwell has his query, he wants to save this to a dashboard. But before doing this, Searchwell thinks to himself, "How nice would it be to visualize this as the U.S. map?" Well, Searchwell is in luck. To do this, he clicks on the **Format** (paintbrush) button in the chart's top right corner, and then opens the **Type** drop-down to get these choices:



Reformatting a bar chart to a different visualization type

This allows Searchwell to select a **Map** chart type for his Search – and, eventually, for his dashboard. So now he has a map, but it's actually rendering as the entire globe, which is not as relevant for Searchwell. So he selects the **USA Map** subtype, and his chart is now perfect.



U.S. map, where something is rotten in the state of abnormal

# FROM VISUALIZATION TO DASHBOARD

Now that he has the visualization he wants, Searchwell will add it to his existing **Okta Investigation** dashboard. In Cribl Search's top right corner, he selects the **Actions** menu and then **Add to Dashboard**. Next, in this modal, Searchwell can title his map and select its select its destination dashboard.



Adding map visualization to a dashboard

He clicks **Add & Go to Dashboard**. Yay! For each of the subsequent queries we'll explore below, Searchwell will follow this same process to add his relevant queries to his **Okta Investigation** dashboard.

#### MOAR GEOGRAPHIC DETAILS, PLEASE

The next query that Searchwell runs is to render more detailed information about the geographically improbable logs. Diving a bit deeper, Searchwell wants to understand the following parameters: time, actor ID, user agent, browser, city, state, postal code, and any debug behaviors. He uses a query like this:

```
dataset="your_dataset_okta_logs"| where actor.alternateId == "beowiff@
cribl.io" and eventType=="user.session.start"| project time, actor.
alternateId, client.userAgent,
client.userAgent.browser, client.geographicalContext.city,
client.ge ographicalContext.state, client.geographicalContext.country,
client geographicalContext postalCode,
debugConte xt.debugData.behaviors
```

And this query renders results like those shown below. Searchwell is fine with a table for this visualization, so he again selects **Actions > Add to Dashboard**, and moves on.

1 defaset="60 client.geogram	AbcalContext.state, cl	"   sherr attar.altern iant.geographicalicotext.cr	<pre>cabd == "be==:Flg::::0.un" untry,client.geographicalC</pre>	and eventType="star.posts entert.postslCode.debugConte	incider[*] drupert time, i st.debugbeta.behaviors	actor-atternate54 , illusi	landt Ajertas I bertann Aje	ot.braker, (taet.geographic	alCartest-Lity,
livering. Justice others							Diving discount \$9825.1	LANKER STREET. Freedom (FE-MAG	Details 🖬 Arminia 1
-									π. •
	beg streamt	SLU PLUZ	nurt-articulat	LCS-578-6-47 \$14.03	70.0.2	C140-C	94-00 month	1010 001040	
0 Dave	actor above and	spear interdiers	specification potents	tions Building gauged tak	tran Bolkets Crease and	conclusion reaction	span halor i harryone	enhagteneest. A behaviors	
Sale of the second second	Server of the server of the	Louising, strengt, in	- AL COURT	(an april)	A R I PROVIDE	format reason	page 2	dense and the prostant sector of the sector	
5900-09-06106-071405	permit color part (a	Finance Locality in	-en Dates	Second Plats	Non Servery	Forder System	8(34)	thes see weather Pisciple.	
Saba na Superatives	Server (Spr. Sp. 14	Frankriss against, " an	The Context	-purper Costs	tics Servicy	Berned Shates-	87967	Spin Ken-Territor-Merilan	
2004-09-00750-01-002	Served Affect Web., Tor	Transa Contra .	-an Drive	Servey Dilly	Res Servey	Deling Thates.	07567	(Rest Rest Land Int POSTTOR,	
2628-63-28738-61/582	tamandepertuit. sa	I DIMANT "CHERE", IN	THE CHERE	Servey crisy	Res Jersey	annes lastes	87.86.7	(now non-constructed states)	
2009-03-08133-01-092	heartfpreisi.iz	Interest "Concert, or	THE DRIFT.	Servey city	Res Jarany	Annual Chatter.	87367	(New Kall-Lanakian-Polizities,	
2404-83-04756-82-992	beautification to	Distance "Control", in	THE CHOIC	Jersey Cray	Num Derkey	Drived States.	8(267	the too-costion estimate	
2024-03-08758-40.222	Semilifperial_re	Linear Science, and	The Deller	Jersey Diny	Row Dormey	Record Status	80397	the do-tase in-relifier,	
2404-43-DITo6.411292	bowilffaorial, ta	Editoriary Period to an	- online	Serves City	tim Server	Rental States	107367	low too-usuation-Mid211vt.	

Search returns detailed time, geographic, and technical data

Searchwell now feels confident that he has collected all the information he would need to confirm and report the attacker's initial access point in this scenario. Interestingly, while he was going through these logs, he discovered his final suspect: Google.

#### Sidenote: How to Build a Dashboard?

Search is an exploratory tool. As you start searching, you usually end up evolving your searches as you go – adding here a clause, there a clause (everywhere a clause clause). Building a dashboard is as simple as the **Add to Dashboard** flow you saw above. If you don't have an existing dashboard, the modal offers a **Create new** button to create and name your first one.

Even better, when you want to have a dynamic dashboard – like the ones we're using – you can edit the dashboard and select **Add Input** for each desired enrichment. For our AWS dashboard, we added a user text input, and a time picker. Then we edited each of the panel's searches to use those inputs.

For details on how to do this, see Cribl's *Editing a Dashboard* documentation, linked at the end of this chapter.

04 | IF YOU CAN'T SEARCH IT, DOES IT EVEN EXIST?

#### THE FINAL SUSPECT: GOOGLE

While exploring Okta logs, Searchwell noticed that the attacker had enumerated a number of applications within the compromised user's Okta Dashboard. Prior to picking which application Searchwell would explore next, he wanted to be able to see **all** the logged-in applications during the defined period. So Searchwell wrote a query of this form and added it to his Search dashboard:

dataset="your\_dataset\_okta\_logs" displayMessage="User single sign on to app" and actor.alternateId == "beowiff@cribl.io" | project time,target.0.displayName, displayMessage, actor.alternateId

B 1 putateseter	attalagenager"ner tirste sige er te sar" ein atter	alternatelå er "annelfigelikklad" i arabeit dien target.Autoa	dayhame, displayMessage, activiatiermeteld
trent held that			Classific scored (202), are the 1962 - Hynnaks (M-Sec Sec B. Albert -
			1.5
entre mire m	nany markan makan makan a	the patter sound balls make	make make with miles with
		43	
D Etma.	terpet 5. Biglie/fame	displayMattage	actor, attentioned
2424-43-247281821488	Sktu-Benfedoard	pres spelle syle on to real	term(//gprtsL)/
2454-43-23996-88-282	null the Silicotty Carbon	disertatingte stige on to laps	Decevit/Approxil_16
204-0-2096-0622	Skina Bandidmand	liger single tigs of the app-	Search Percentel, 19
2504 48-26821-42-303	Lasker	liter single sign on things	New Percent, 10
3924-43-39727(39)-492	Stationer Store	liver strate star to telean	Depart 1793ur 1934, Sa
2424-43-24727-31-312	Actional are Claud	liver single sign on to see	Second 1990-1991-19
3804-43-04T11-89-107	Lyndate	User single sign as in app.	Second Chargers 1981. Tax
280-48-04717-69-692	Kaute	liver single silps of talles.	Second Page 191-10
2424-43-24717181-347	Balar:	court stright ange on its age	Second Placemail. Yo
2020-03-20120-02-022	Sall Dat Incoming Conter-	case single ange on to app	Same (Peperta)_is
2407-43-04724-122-282	Google Welhause	lawr single arge on ta sea	Search 19960 (201., to
2027-03-00124-05-062	the Life Linearity Contar	ther single sign or to sea	Developmentation of
345+-43-56F54(32-182	dougle workspore	diver single sign on in spa	beautifigerial.to
2424-48-24734-44-242	Diru bestiere	iter sight sign on the spa	beautificantia.to
2124-10-2015-0-44-202	desals meriapate	liver visible star on to ass	benet Algorital. In
The at strictly	tiets instituted	over wingle star of to any	benef (fijoria). ta

This query rendered the following *interesting* information:

Search returns suspect account's application behavior

In this table, Searchwell hones in on the Google Workspace application authentications, as there is a potential for "sensitive" company data to be accessed there. Searchwell does not panic. He has gotten this far with Cribl Search, and is going to finish this job with one final query. What exactly did the attacker access or have access to in Google Drive? Searchwell writes a query like this:

dataset="your\_dataset\_google\_logs" | where actor.email ==
"beowiff@cribl.io" | project id.time, actor email, events.0.name,
events.0.type, events.0.parameters, id.applicationName, ipAddress

This query returns the following results:

	68 (32)			Sampling	7 days ago Local S	SEARCH
13 1 dataset=" un"	<pre>&gt;" juncre actor.email == "t</pre>	Deow1750crio1.io"] project is	s.time,actor.email,events.0.eam	e,events.0.type,10.application#	lane, LpAddress	
Eventa Fields Chart				Objects discovered: 1, scanned: 0	0 10 results (218.75s) Details 🚺	Actions v
4,995.1						н 🕸
1004/0	1.0117 - 1014-00, 1.7077 - 1014	LIN TABLE INTAGE THAT IN	100 -1 MIZ 200-45 1-4027	on silent meiline kannt meilig	N AMER WINNI AMMI	
TTAKE	1000	any name motor many ma		en såen moåns kannt modag	n khapi mikan shiri	
0 E H, time	score and story out	un sam mun sut mu eees,0,0ams	0.000 (0.000 (0.000 (0.000)) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000	nt stämt motern känst moter käjeppikationNerne är ive	n khatt misken skorre påddress bå dar 15 185	<b>1</b> ,000
1994-03-20116-55-13.7972	scoreman sectoreman securificaritat.so securificaritat.so	events, graame events, graame adit	eens, 9, type eens, 9, type excess access	nn Saland molarm 8,8017 molard MiljeppicationName Artive Crive	påddress 94.5m 13.185 94.5m 13.185	
0 M Jine 2024-03-30156-56, 25, 8137 2024-03-2616-55, 21, 7872 2024-03-2616, 55 21, 7872	score and score	evens guara evens guara adir edit vice	evens, 8, type evens, 8, type evens, 8, type evens, 8, type evens, 9, type	nt saint mban saint mban d'asplationNene érise erise erise	plotes 96.141.15.145 96.241.15.145 96.241.15.145	
0 Muline 2024-01-30106-155-8132 2024-03-20106-155-11,7872 2024-03-20106-156-31,7872 2024-03-20106-09-33,1472	scorpoial scorpoial securificental.in securificental.in securificental.in securificental.in	ener same amant stare and events grane ante edit vise Vise	erris appendi hazar men erris appendi hazar men erris appendi erris a eccess eccess eccess erros	ni tahu muan kasir mua U kepikationiene Arise Arise Frise	plotes e6.5a15.5145 96.246.55.145 96.246.55.145 96.246.15.145 96.246.15.145	
C NJ.ine     Ditate - 1:-22110:55:51:14:22     2014-01:-20110:55:51:11,7112     2014-01:-20110:55:51:11,7112     2014-01:-20110:50:21;1122     2014-01:-20110:50:21;1122     2014-01:-20110:51:11,1122	Autor and Aren's and addr.chail Securif/Schill.in Securif/Schill.in Securif/Schill.in Securif/Schill.in Securif/Schill.in	eneral grants that the eneral grants date efft view view view	and what oppen hand man all ma encode Appe across across across across across across	nt salet mban santh mban MagginatioNene drive drive drive drive drive drive	p 3.5447 ergalant Salaver <b>pAddress</b> 66.248-15.145 66.248-15.145 66.249.15.145 66.249.15.145 66.249.15.145	
100.40 0. M. Jine 103.4-01-021141-50, 10, 41.01 204.4-02-02114-50, 201.4-02 204.4-02-02114-50, 201.4-02 204.4-02-02114-50, 201.4-02 204.4-02-02114-50, 201.4-02 204.4-02-02114-50, 201.4-02 204.4-03-02114-50, 201.4-02 204.4-03-0214-50, 201.4-02 204.4-03-02 204.4-0	Valle weeks a //er/ min actor_enall SeverTrgs:N3.16 SeverTrgs:N3.16 SeverTrgs:N3.16 SeverTrgs:N3.16 SeverTrgs:N3.16 SeverTrgs:N3.16	eens plane motor may me eens plane afte eft vies vies vies vies	evens, bype evens, bype excess eccess eccess eccess eccess eccess eccess	nt saint mtans saint mtan utjappisatontene artise drive artise artise artise artise artise	plature start platures 64. Jan. 13. 145 96. Jan. 15. 145	
Image: Section (Section (	Verrat worken a freed with actor, enable Second Paper Stal. (In Second Paper Stal. (In	eens bleet motor maar ma eens glaane eets eets vree vree vree vree vree vree vree	ANT ANT ODEAD FACT THE EVEN ACOUS ACOUS ACOUS ACOUS ACOUS ACOUS ACOUS ACOUS ACOUS ACOUS ACOUS	at taket means smith mean drive drive drive drive srive drive drive drive drive	påderses 94.5447 stylene 5.445 94.544 151.145 94.544 151.145 94.544 151.145 94.544 151.145 94.544 151.145 94.544 151.145	
C      C      M.2444     Source     Sou	Anter and Arrive and addr.chail Second (Specifia), in Second (Specifia), in	event bland motor motor motor ante ante vice vice vice vice vice vice vice vic	eres s	na salat mban sanif mban MagginatioNene Price Price Price Price Price Price Price Price Price Price	p 2 5 427 version 5 8007 <b>pb/dess</b> 66.246.15.145 66.246.15.145 66.249.15.145 66.249.15.145 66.249.15.145 66.249.15.145 66.249.15.145 66.249.15.145	
Implies           Implies <td< td=""><td>Vatta Webb 3,7677 MM actor_enall WwwFFQPV33,10 accordfgev33,10 accordfgev33,10 Accordfgev33,10 beaut/fgev33,10 beaut/fgev33,10 beaut/fgev33,10 beaut/fgev33,10 beaut/fgev33,10</td><td>eess plane moot young mo eess plane effs effs view view view view view view view view</td><td>erria, 0, type erria, 0, type arross access access erross access access access access access access access access</td><td>nt sales mbans sanst mban al applicationterie : active drive drive drive drive drive drive drive drive drive</td><td>p 3.5.447 errolen 5.6497 pb0fres 06.1.841.15.185 06.2041.02.15.485 06.2041.02.15.485 06.2041.02.15.485 06.2041.02.185 06.2041.02.185 06.2041.02.185 06.2041.02.185</td><td>•</td></td<>	Vatta Webb 3,7677 MM actor_enall WwwFFQPV33,10 accordfgev33,10 accordfgev33,10 Accordfgev33,10 beaut/fgev33,10 beaut/fgev33,10 beaut/fgev33,10 beaut/fgev33,10 beaut/fgev33,10	eess plane moot young mo eess plane effs effs view view view view view view view view	erria, 0, type erria, 0, type arross access access erross access access access access access access access access	nt sales mbans sanst mban al applicationterie : active drive drive drive drive drive drive drive drive drive	p 3.5.447 errolen 5.6497 pb0fres 06.1.841.15.185 06.2041.02.15.485 06.2041.02.15.485 06.2041.02.15.485 06.2041.02.185 06.2041.02.185 06.2041.02.185 06.2041.02.185	•

Search displays suspect accounts actions on Google Drive

Searchwell is able to see that the attacker was able to view and edit several different items within the organization's Google Drive. Searchwell continues to enumerate the log files and grabs the files' IDs.

With these IDs, he uses the Google Admin tool (or GAM) to look up the files. This shows Searchwell that the attacker has added an external email to files titled IPO Plans and 1Password Password and AWS Passwords.

Result? The team disables the compromised user's Okta account, and begins cleanup work to ensure that there are no backdoors or remaining entry points for the attacker.

#### THE END

Searchwell's boss names him the best "Detective" in the SOC and commends him for using Cribl Search to identify this attacker. We must allow Searchwell to monologue his excellency:

My comrades, today, we were able to fight back against our adversary with a tool unlike any other. Search is the investigative tool of the future. If we had not set up Search last week, we might have missed the clues to solve this crime. The initial access via Okta, the AWS enumeration, and the access and exfiltration of sensitive company data was a doozy of an attack. However, we were able to mitigate this crime through our own investigative superpowers.

Tomorrow, my friends, there will be a new adversary. But as Theodore Roosevelt once said, "It is not the critic who counts; not the man who points out how the strong man stumbles, or where the doer of deeds could have done them better. The credit belongs to the man who is actually in the arena, whose face is marred by dust and sweat and blood; who strives valiantly; who errs, who comes short again and again, because there is no effort without error and shortcoming; but who does actually strive to do the deeds; who knows great enthusiasms, the great devotions; who spends himself in a worthy cause; who at the best knows in the end the triumph of high achievement, and who at the worst, if he fails, at least fails while daring greatly, so that his place shall never be with those cold and timid souls who neither know victory nor defeat."

Through this incident, Searchwell wielded the power of Search in combination with alerts from his SIEM. Searchwell learned how to write powerful queries, often starting with what he knew and diving deeper, and finally discovering how to make dashboards to show a holistic picture of an attack.



Searchwell's honor, well-earned

#### VARIATION

Search is a powerful tool, and there are many ways to configure it and use it to investigate potential incidents. There might be times, however, when Search needs some extra juice. Let's see how to use the Parquet object storage format to increase Search speeds. 04 | IF YOU CAN'T SEARCH IT, DOES IT EVEN EXIST?

#### Put Some Juice in That Search — How to Increase Search Speed

Search does its searching across S3 buckets, using a partitioning scheme that's defined when writing to S3. Step One in making Search perform well is to get the partitioning scheme and data format right.

We use Cribl Stream to populate our S3 buckets, using an Amazon S3 Data Lake Destination that defines a partitioning scheme. In the corresponding Cribl Search dataset, we use this matching scheme as our **Bucket Path**:

```
<bucket name>/${*}-${_time:%Y}/${*}-${_time:%m}/${*}-${_time:%d}/${*}-${_time:%H}
```

And because CloudTrail is an *immense* dataset for an AWS environment of any reasonable size, we also want to optimize the data format. Using JSON is not performant on a dataset of this size. So when configuring our S3 Data Lake Destination in Stream, we instead set the output **Data format** to Parquet, which is significantly more performant.

Destinations + Data Lakes A searchwell	nazon 53	0>
Configure Status Charts	Live Data Logs Test Notifications	
General Settings	Output ID* ②	
Authentication	- Alternational Astronomy and Astronomy and Astronomy and Astronomy and Astronomy and Astronomy and Astronomy a	
	53 bucket name* (1)	
Assume Role	'searchwell-cloudtrail'	50
Processing Settings	Data format* (2)	
	page	
Post-Processing	ISON	
Advanced Settings	Rhw	
	Parquet	
	Backpressure behavior ①	
	Block	
	Tags ()	
	Concept New York Concep	

Sending data to S3 in Parquet format, for faster Search retrieval

Cribl Search recognizes the Parquet format, and efficiently retrieves the structured data.

#### WHAT'S NEXT?

For a guided, live walk-through on configuring Cribl Search for yourself, take our free **Cribl Search** Sandbox for a spin at sandbox.cribl.io/course/overview-search.

If you aren't already using Cribl Search or other Cribl products, sign up for a free Cribl.Cloud account at cribl.io/solved. You'll be ready to start your first search immediately.

For details on the Cribl Search and other resources we've applied here, see our online documentation:

- Datasets and Dataset Providers (docs.cribl.io/search/#datasets).
- KQL/Kusto Query Language (docs.cribl.io/search/your-first-query).
- Editing a Dashboard (docs.cribl.io/search/editing-dashboards).
- Cribl Stream > S3 Data Lakes Destination (docs.cribl.io/stream/destinations-data-lake-s3).
- Cribl Lake (docs.cribl.io/lake).



# 5 | Streamline Windows Monitoring with Cribl Edge

Use Fleets to Centralize Management with Fewer Agents

by Ahmed Kira, Sri Kotikelapudi, and Paul Dugas, Cribl

#### PROBLEM

Monitoring and administering a large number of servers and clients can require installing multiple agents. Parsing through individual log files per machine is laborious. Managing the agents, themselves, becomes a burden – requiring shelling into individual machines to update and restart agents.

This all adds up to redundant, tedious work. Practitioners find this burden manageable up to a few dozen or even hundreds of machines, but a fleet of thousands rapidly becomes almost impossible for a team to manage.

# SOLUTION

Cribl Edge eliminates these tedious processes by providing a centralized, comprehensive and reliable platform for monitoring and analysis across your Windows machines. Edge collects and processes logs, metrics, and other data from Windows Servers (2016, 2019, or 2022). Edge Nodes can also run on "always-on" workstations with current Windows versions, sleep/hibernate options disabled, and an uninterrupted network connection. By using Edge to route this data to your favorite system of analysis or visualization, you can identify and investigate issues faster, and monitor applications more effectively.

#### WHY CONSIDER THIS APPROACH?

Notably, by running Cribl Edge on Windows domain controllers, you can collect security-related Windows Event Logs events from Windows Nodes – such as logins/logouts, failed logins, and password changes. Collecting this data centrally on the domain controller removes the need to instrument a fleet of laptops or other clients. Some other agents provide this facility, but Cribl Edge handles it particularly effectively.

Beyond the Security realm, Edge offers the same centralized collection, processing and routing for Windows performance metrics, and for logs and metrics generated by specific applications running on your Windows Servers. This allows you to troubleshoot application-specific issues and ensure their smooth operation.

#### HANDS-ON RESOURCES

Although Edge Nodes (instances) run on Windows hosts, the Leader that coordinates them is a Linux-only application. The fastest way to launch a Cribl Leader is to let Cribl run it for you: Sign up for a free Cribl.Cloud account at cribl.io/solved.

This will give you a Leader from which you can deploy Edge Nodes on your own systems. You'll also get free instances of Cribl Stream, Cribl Search, and Cribl Lake. When you need more capacity, you can readily upgrade to a paid account.

If you prefer to run a Leader on your own infrastructure, you can download an Edge (and Stream) Linux package at cribl.io/download. This chapter shows you how to install Edge onto Windows machines via script, but if you'd rather download a Windows .msi installer to run yourself, the same download page provides that option.

At the end of this chapter, you'll find links to individual topics of Cribl Edge online guidance. The starting point for all Edge documentation is docs.cribl.io/edge.

#### SCENARIO

This chapter presents a simple but realistic overview of a whole Cribl Edge life cycle for Windows monitoring: Deploy a Node, assign it to a Fleet, examine a host's state and logs, configure specific inbound integrations to monitor files and metrics, route data to downstream services, and execute and process PowerShell commands.

#### WHAT WE'LL DO

We'll walk you through the following techniques:

- Create a Windows-specific Fleet and Subfleet, to centrally monitor multiple machines that share the same configuration.
- Deploy Cribl Edge Nodes on Windows machines, by automatically generating and running a bootstrap script.
- Create Mapping Rulesets to set and switch Nodes' Fleet assignments.
- Explore a Windows Node to snapshot the host system's current state.
- Locate and collect custom Windows event logs and send them to your chosen visualization or analysis tool (or to any supported destination).
- Collect Windows metrics and send them to any supported destination.
- Collect application logs using the File Monitor Source.

As a bonus, we'll show how to use Cribl's Exec Source to execute PowerShell commands and process them in Cribl.

#### DEPLOY CRIBL EDGE ON WINDOWS

This section covers creating a Windows-specific Fleet, creating a logsspecific Subfleet, deploying ("bootstrapping") Windows Nodes, and assigning the Nodes to the Fleet and Subfleet. It's a snapshot of a broader, very powerful Cribl Edge feature that facilitates managing as many as several thousand hosts. To learn more about this feature, see "Variation: Using Fleets to Manage Many Diverse Hosts" below.

#### CREATE A WINDOWS-SPECIFIC FLEET

Fleets and their corresponding Subfleets allow you to author and manage shared configuration settings for a particular set of Edge Nodes. At a minimum, Cribl recommends separating Linux versus Windows machines into different Fleets per OS.

The first step is to configure a Fleet and name it Windows. (We've chosen this literal name to keep the demonstration simple. Fleet names must be unique across your deployment, so in production, select Fleet names that will convey each Fleet's purpose. For better practices on designing and naming your Fleets, see the references at the end of this chapter.)

To set up our mapping illustration later, let's also create a Subfleet within the Windows Fleet. We're naming this Subfleet windows\_logs.

Subfleets New Subfleet	X
Fleet name* windows_logs Description ⑦ Windows Event Logs Enable teleporting to Nodes ⑦ Yes Inherited configuration ⑦ Windows	Subfleets and their Edge Nodes inherit configurations from their parent Fleets. Optionally, select an existing Fleet from which your new Fleet will inherit configurations.
	Cancel Save

Creating a Subfleet

# USE THE BOOTSTRAP SCRIPT TO ADD AN EDGE NODE

The Fleet provides logical structure on your Cribl Leader, whether running in Cribl.Cloud or your own infrastructure. The next step is to deploy Cribl Edge software on individual Windows machines.

From the Edge UI, you can readily generate a script to deploy these Edge Nodes. Below is a composite screenshot of generating this script for Windows.

	Add/Update Edge Node	× 4	dd Windows Node	
Add Updat	Docker Kubernetes Linux Windows	2 2 4	Acyos foi and mostly theirs Balos, they'l populate the Sorat Ib each hastacane (# 10) and the solar port () and taken () (and at taken () () () () () () () () () () () () () (	as an the right. To add your new Edge Holds, angly this sorp and resource it an the Edge Hold's command line. Learnment, Sorp CO mit form: // https://citik.or/ib/is//citik.or/i
			State 1	

Generating a bootstrap script for a Windows Edge Node

To add the Edge Node, paste the script into your Windows Command Prompt and execute it. (You must run the Command Prompt as an administrator for this to work.)



Pasting the script onto the command line

You'll need to do this on each target Windows machine, but only for initial setup there.

After deployment, you'll have the option to edit each target machine's instance.yml file to reassign the Edge Node to a different Fleet, to modify TLS settings on its communications with the Leader, or to customize other options. (Reference at the end of this chapter.)

# USE A MAPPING RULESET TO SWITCH FLEETS

Next, we want to assign the newly deployed Windows Edge Node to the windows\_logs Subfleet that we created for it. We can do this efficiently within Edge's Mapping Ruleset – where we can set one rule for all matching Nodes.

First, we add a new platform-specific rule in the existing default Ruleset:

😂 Edge 😑 Hor	me Manage Set	tings			
Fleets Edge Nodes	Mappings N	lotifications Logs			
← Editing Active R	uleset: default			Add R	ule 🗗
# Rule Name		Filter	Fleet	© All	
II > 1 Default Map	pings	!cribl.group	default_fleet	On	
li ∨ 2 Windows		platform==win32	windows_logs	On	
Rule Name 🗇 Wind	ows				đ
Filter 🗇 plat	form==win32				R
Fleet* winde	ows_logs				×.
					-
				Cancel	Save

Mapping to the SubFleet

An important note: Only one Mapping Ruleset can be active at a time – although a ruleset can contain multiple rules. So it's important to keep your Fleet assignments within one ruleset. This example uses the default ruleset.

We want to reorder the rules, so that the new rule is first:

😂 Edge	e ≡	Home	e Manage	Settings			
Fleets	Edge I	Nodes	Mappings	Notifications	Logs		
←	Editing A	ctive Rul	eset: default				Add Rule
#	Rule	Name		Filter		Fleet	⊚ All ▼
∥ > 1	Wind	lows		platform==	win32	windows_logs	<u>On</u>
> 2	Defa	ult Mappi	ngs	!cribl.grou	up	default_fleet	<u>on )</u>
							Cancel Save

Reordering the rulesets

After we resave the Ruleset, Edge Nodes that match the platform value will now be assigned to the new SubFleet.

2 Noder	1/4 Active F	eets	2 Alive	Ontrea	imy-	1 Saltwire	Versions	1 Cont Venzione	
leet Ali	-10	a kini na sine		С				Add/Update Edge	Node
		Host	Agent	Fieet	Last Time	Start Time	Config Version	Cribl Version	Msg
3fe84a8p-9385-	4499-8cc7-7827	ECZAMAZ-BRZFN1D	🖬 aliv	windows logs	2023-04-18 1	2023-04-171	365cb4f	A 1.1-RC1-83916317	ά

#### Viewing Edge Nodes

#### EXPLORE THE WINDOWS NODE

On your Leader, from the Fleet's **Explore** menu, you can select the **System State** tab to see snapshots of the host system's current state. The time intervals are configurable at the upper right.

Reet oilast, feet	* Overview	Depute Callect Heat	Mare * Rest Settings		♦ 0817996 - C
Explore		_			
Node to suplore EC2AMA21/Am4Q82	3-03242563387				
Processes 17	tes. System State				
Host Info	Host Info at 2023-03-14-11:16:06	2020-01-01-12-05			1 httar ági
Disks	Key			Value	
DNS	host			16274442-0445823	
File Systemi	timetarip.			1678806625.076	
Groups	ariti sertine			4.5.07034-30994-084	
Longto Die	utili modit			managed-edge	
Looper Lan	o'thi group-			detail: Fee	
implaces	unbl config, version			0424293	
Routes	-to-jectr			864	
Using.	as (ps.) court			2	

Edge System State tab

You can specify which collectors are enabled, and the polling interval, by configuring the System State Source (see the reference at the end of this chapter).

#### CONFIGURE THE WINDOWS EVENT LOGS SOURCE ON CRIBL EDGE

Next, we'll look at how to get Windows Event Logs into Edge, and on to your chosen destination. We must first identify the logs we want, and then configure Edge's input (Source) to match.

#### LOCATE WINDOWS EVENT LOGS ON THE SERVER

Event logs in Windows Servers are classified into the following broad categories:

- **System**: Events related to the system and its components. Failure to load the boot-start driver is an example of a system-level event.
- Applications and Services: Events related to software, or to an application hosted on a Windows machine, get logged. For example, if a user encounters a problem in loading the app, it will be logged.
- **Security**: Events related to the safety of the system. The event gets recorded via the Windows auditing process. Examples include failed authentications and valid logins.
- **Setup**: Events that occur during the installation of the Windows operating system. On domain controllers, this log will also record events related to Active Directory.
- Forwarded Events: Contains event logs forwarded from other machines in the same network.

To find out what other event logs might be of interest, examine your Windows Server's Event Viewer.

Event Viewer
File Action View Help
🗢 🔿 🗾 🖬 📓 🖬
Event Viewer (Local)
🗸 📑 Custom Views
Server Roles
Administrative Events
🗸 📑 Windows Logs
F Application
Security
Setup
🛃 System
Forwarded Events
<ul> <li>Applications and Services Logs</li> </ul>
Active Directory Web Services
DFS Replication
📄 Directory Service
DNS Server
F Hardware Events
👔 Internet Explorer
📔 Key Management Service
> Microsoft
> 🦰 OpenSSH
Windows PowerShell
Caracteristic Subscriptions

Examining Windows Event Viewer

You can get each event log's accurate name by right-clicking and selecting **Properties**.

Þ 🔿 📩	? 📻	
Event Viewer (Lo	call)	
Custom View	15	
> Server Ro	les	
T Administ	rative Events	
🗸 🚺 Windows Log	gs	
Application	on	
Security		
Setup	Open Saved Log	
👔 System	Create Custom View	
Forwai	Import Custom View	
Applicatio		
Active	Clear Log	
DFS Re	Filter Current Log	
	Properties	
E DNS S	Find	
	Save All Events As	
👔 Key M	Attach a Task To this Log	1
> 📫 Micros > 🚞 OpenS	View	>
Windo	Refresh	
Subscripti	Help	>

Viewing log properties

05 | STREAMLINE WINDOWS MONITORING WITH CRIBL EDGE

#### GET THE LOG'S FULL NAME

From the **Properties** modal, we'll copy the **Full Name**. When we configure Edge's **Windows Event Logs** Source below, we'll paste this into the **Event Logs** field.

Log Properties - Sec	urity (Type: Administrative)	×
General		
Full Name:	Security	
Log path:	%SystemRoot%\System32\Winevt\Logs\Security.evtx	
Log size:	128.00 MB(134,221,824 bytes)	

Copying the full name from the Event Viewer UI

Alternatively, you can execute the following PowerShell command to list a particular log's name. In this example, we're using a \* wildcard:

powershell Get-WinEvent -ListLog Hardware\*



Log name from PowerShell

From the PowerShell output, the **LogName** field's value is what you'd copy to paste into the Edge UI. From the above example, you'd enter Active Directory Web Services in the Edge UI.

#### CONFIGURE THE WINDOWS EVENT LOGS SOURCE

Once we have the accurate names of the event logs we want to collect, we can configure and enable the Windows Event Logs Source in Cribl Edge. An example configuration modal is below.

onfigure	Status Charts	Live Data Logs Notifications			
General Se	ettings	Input ID* ③			
		in_win_event_logs			
Processing	Settings	inputId=='win_event_logs:in_win_event_logs' 🗍			
Fields		Event Logs* ⑦			
		II Security			
Pre-Pro	ocessing	Add Log			
Advanced	Settings	V OPTIONAL SETTINGS			
Connected	Destinations	Read Mode 🗇			
		Entire log From last entry			
		Polling Interval ①			
		10			
		Batch Size ③			
		500			
		Tags ③			
		Ether tags			

Configuring Windows Event Logs

After we've configured the required fields (and any desired optional settings), we'll save, commit, and deploy the changes. Before moving on to the next step, we should check the Source's **Live Data** tab to make sure the logs are generated.



Verifying log generation via Source's Live Data tab

We can now send the event logs to a visualization tool of our choosing, or to any of Cribl Edge's other supported Destinations. We also have the option to route the logs through Cribl Stream for further processing on Stream Worker Groups with greater capacity.

#### CONFIGURE THE WINDOWS METRICS SOURCE

We'll also take a quick look configuring and enabling Edge's Windows Metrics Source.

The key requirement here is to decide the level of granularity for the metrics you want to collect. On this Source's **Host Metrics** left tab, we recommend selecting the **Custom** button, and then setting the following granularity options for three of the configurable types:

- CPU: Enable Per CPU metrics.
- Network: Enable Per Interface metrics.
- Disk: Enable Per Volume metrics.



Configuring a Windows Metrics Source

# CONFIGURE THE FILE MONITOR SOURCE

To round out this walk-through, we'll take a quick look at enabling the File Monitor Source to gather application logs. The example configuration below monitors all files that end with variations on .log, as well as .journal files.

Sources - File Monitor in_file_auto		© ×
Configure Status Charts	Live Data Logs Notifications	
General Settings	Input ID* ③	Enabled 🕐 🐖 🔾
Processing Settings	_inputId=="file.in_file_auto"[]	
Event Breakers Fields	Discovery mode © Auto ① Manual ①	
Pre-Processing	✓ OPTIONAL SETTINGS Polling interval ⊙	
Advanced Settings	1 Filename allowiist 🗇	
	Projecumal X Profailing X Phasting X Phasing S Program Angel X Phasting X Pha	
	Check File modification times () ()	
	Yegs U	

File Monitor Source in Auto mode

The Leader deploys with two File Monitor Sources preconfigured, respectively to operate in **Auto** and **Manual** modes. **Manual** enables you to specify a single path from which Edge will collect log files, along with a **Max depth** setting where you can limit collection to a specified depth of subdirectories.

windows_files		0	
onfigure Status Cha	rts Live Data Logs Notifications		
General Settings	Input ID* 🗇	Enabled ()	
Processing Settings	inputIder'file:windows_files/ []		
Event Breakers	Discovery mode ③		
Fields	Search path* ()	Max depth 🕥	
Advanced Settings	C:UBersVAdministrators/Downloads/FDR		
Connected Destinations	Polling interval ① 10		
	Filename allowlist ① FDR-CSPack-d* ×		
	Max age duration 🗇		
	Check file modification times (3)		
	Collect from end ()		
	Tags (2)		

File Monitor configuration in Manual mode

# VARIATION: EXECUTING POWERSHELL COMMANDS

Here's the bonus topic. Edge's flexible Exec Source enables you to periodically execute a command and collect its stdout output. You'd typically configure and enable this in cases where Cribl Edge cannot accomplish collection with native Sources.

The Exec Source command is actually executed from a Windows command prompt, not from a PowerShell prompt. (PowerShell's execution is resource-intensive, so use it with caution. This applies especially to limited-resource systems, such as domain controllers, because the Source itself uses the host's CPU.)
The main thing to keep in mind when you configure Cribl's Exec Source is that your entry in the **Command** field will be slightly different from the corresponding PowerShell command.

PowerShell Prompt Command	Cribl Exec Source Configuration
Get-Process	powershell Get-Process
<pre># cd c:\mydir # .\myps-script.ps1</pre>	powershell c:\mydir\myps-script.ps1

## POWERSHELL EXAMPLE

To process the PowerShell command for a script like this:

Get-CimInstance win32\_process | select ProcessId, Name, commandline, HandleCount, WorkingSetSize, VirtualSize

...you would enter the following into the Exec Source's **Command** field. (Add the -command parameter if your PowerShell version requires it:)

powershell [-command] "Get-CimInstance win32\_process | select ProcessId, Name, commandline, HandleCount, WorkingSetSize, VirtualSize"



Populating Exec Source's Command field

# EXAMPLE OUTPUT

The output of the command, without using an Edge Event Breaker, will look something like this:

Filter Expression* 💿		
inputId=='exec:gcim'		
Fields 🗍 🛛 All None		
✓ _raw ✓ _time ✓ cribl_breaker ✓ source		Event
	1 2023-04-30 12:46:34.327 -07:00	<pre>d _rew: ProcessId : 0 # _time: 1682883994.327 0 cribl_breaker: fallback n source: stdout</pre>
	2 2023-04-30 12:46:34.327 -07:00	n _raw:Name : System Idle Process # _time:1682883994.327 0 cribl_breaker:fallback 0 source:stdout
	3 2023-04-30 12:46:34.327 -07:00	<pre>n _raw: commandline : # _time: 1682883994.327 n cribl_breaker: fallback n source: stdout</pre>
	4 2023-04-30 12:46:34.327 -07:00	<pre>q_raw:HandleCount : 0 #_time:1682883994.327 p_cribl_breaker:fallback n_source:stdout</pre>
	5 2023-04-30 12:46:34,327 -07:00	a _raw:WorkingSetSize : 8192 # _time:1682883994.327 n cribl_breaker:fallback g source:stdout
	6 2023-04-30 12:46:34.327 -07:00	n _raw:VirtualSize : 8192 W _time:1682883994.327 g cribl_breaker:fallback g source:stdout
	7 2023-04-30 12:46:34.327	u _raw:ProcessId ; 4 # _time:1682883994.327 u cribl_breaker:fallback

Output with unbroken events

#### With an Event Breaker, the output will look something like this:

Rules		
Filter Condition* ③ true V EVENT BREAKER SETTINGS Enabled ④ (***) Event Breaker Type* ③ Regex Event Breaker* ④ / Event Breaker* ④ / Event Bytes ④ 512000	In Out ProcessId : 2460 Name : Sxxhost.exp Consandling : C:\Windows\System32\sxcho MorkingStize : 244546 VirtualSize : 2203402817536 ProcessId : 2468 Name : 2xxhost.exp commandling : C:\Windows\System32\sxcho p -5 SysMain HandleCount : 137 WorkingStesIze : 2252800 VirtualSize : 2207683502080	Upload Sample File Remote File st.exe -k NetSxcs -p -s iphlexxc. st.exe -k LocalSystemNetworkRestricted -
V TIMESTAMP SETTINGS Timestamp Anchor* ③ / * Timestamp Format* ③ Autotimestamp Scan Depth ③ Manual Format ③ ④ Current Time ③	ProcessId : 2484 Name : sxchosl.cxc commantline : C:Windows\system32\sxcho NanleCount : 213 WorkingSetSize : 220339261153 ProcessId : 2500 Name : sxchosl.cxg commantline : C:Windows\system32\sycho	<u>st.exe</u> -k LocalService -s W32Time. s <u>t.exe</u> -k NetworkService -p -s <u>CryptSyc</u> .

Output with an Event Breaker

## VARIATION: USING FLEETS TO MANAGE MANY DIVERSE HOSTS

Cribl Edge's hierarchical Fleets support heterogeneous deployments to monitor different types of hosts, in large numbers. You can use a top-level Fleet to set common configurations (such as Edge Destinations and processing Pipelines). Then you can set more-specific configs (like Edge Sources) in nested, descendant Subfleets.

For example, let's imagine (and analyze) a hierarchy of Fleet and Subfleets like this:

#### - **XYZ\_Corp** - top-level Fleet for the entire company

- SiteA separate Subfleets for each data center
- SiteB
  - SiteB\_Linux separate Subfleets for different platforms
  - SiteB\_Windows
    - SiteB\_Windows\_ISS separate Subfleets for different apps
       SiteB\_Windows\_SQL

Here's how your organization might configure these levels:

The top-level **XYZ\_Corp** Fleet could have Destinations common to all Edge instances, such as the company's central Cribl Stream cluster or notification systems. Routes and pre-processing Pipelines could be defined in common here for these Destinations.

The **Site** Subfleets could have additional local Destinations, or overrides of the parent Fleet's companywide instances. You could adjust the preprocessing Pipelines here to add site-specific properties useful to these local Destinations. You could also add local Routes to configure local logs and/or metrics storage.

On the platform-specific Subfleets, you'd enable Sources to collect platformspecific data. Examples would be File Monitor Sources for /var/log on Linux; Windows Event Logs Sources on Windows; and System Metrics or System State. To these Subfleets, you'd map Cribl Edge instances that run on generic hosts (those without application-specific monitoring needs).

The bottom-level, application-specific Subfleets are where you'd add Sources configured to collect logs and metrics from individual applications. To these Subfleets, you'd map Edge instances running on hosts that run each corresponding application. This is just one illustration of how Edge's hierarchical Fleets provide a framework for managing large heterogeneous deployments. Fleets and Subfleets enable you to collect logs and metrics broadly, without having to independently manage each Node. This, in turn, facilitates managing as many as thousands of hosts.

## WHAT'S NEXT?

You can explore Cribl Edge hands-on in our free **Introducing Cribl Edge** Sandbox (sandbox.cribl.io/course/edge). You'll find a hosted, fully functional instance of Cribl Edge, with source data set up for you to ingest and experiment with. The Sandbox walks you through deploying Edge Nodes, managing Fleets, configuring and managing Sources and Destinations, and exploring the Edge UI to discover insights into all the monitored data.

For further details on the Cribl Edge techniques outlined above, check out our online docs:

- Exploring Cribl Edge on Windows (docs.cribl.io/edge/explore-edge-win).
- Monitoring your Infrastructure with Cribl Edge (docs.cribl.io/edge/usecase-infrastructure-monitoring).
- Windows Observability Using Cribl Edge online counterpart to this chapter (docs.cribl.io/edge/usecase-windows-observability).
- Installing Cribl Edge on Windows (docs.cribl.io/edge/deploy-windows).
- Fleets (docs.cribl.io/edge/fleets).
- Mapping Edge Nodes to Fleets (docs.cribl.io/edge/fleets#mapping).
- Fleets Hierarchy and Design (docs.cribl.io/edge/fleets-design).
- Setting Up Leader and Edge Nodes (docs.cribl.io/edge/setting-up-leader-and-edge-nodes).

- instance.yml file reference (docs.cribl.io/edge/instanceyml).
- Windows Event Logs Source (docs.cribl.io/edge/sources-windows-event-logs).
- Windows Metrics Source (docs.cribl.io/edge/sources-windows-metrics).
- File Monitor Source (docs.cribl.io/edge/sources-file-monitor).
- System State Source (docs.cribl.io/edge/sources-system-state).
- Exec Source (docs.cribl.io/edge/sources-exec).
- Cribl Edge to Cribl Stream (docs.cribl.io/edge/usecase-edge-stream).
- Event Breakers (docs.cribl.io/edge/event-breakers).
- Microsoft's Start Command Prompt as an Administrator topic.

