Vectra Stream: Google SecOps SIEM Integration - User Guide

V 1.0.0

Overview

Vectra Stream Platform

Vectra Stream from Vectra delivers scalable, security-enriched metadata from native cloud, hybrid cloud and enterprise traffic that empowers skilled security analysts and threat hunters to perform conclusive incident investigations.

Google SecOps

Google SecOps is a cybersecurity telemetry platform for threat hunting, and threat intelligence and is part of the Google Cloud Platform. Google SecOps stores log events it receives in two formats: either as the original raw log or structured Unified Data Model (UDM) log. There are two critical elements to consider for parsing, Unified Data Model (UDM) which defines the schema for parsing, and Configuration Based Normalizers (CBN) which describes how log data is transformed to the UDM schema.

Google SecOps Integration for Vectra Stream

This integration enables Google SecOps SIEM to receive the security-enriched cloud and network metadata from the Vectra Stream via Syslog, which enrich threat detection and response capabilities with comprehensive network metadata.

This newly developed Vectra Stream parser is specifically optimized to support logs ingested in JSON format. This allows for accurate parsing and proper mapping of fields to the Unified Data Model (UDM). Although the parser includes fallback support for non-JSON formats (such as KV) to accommodate existing user configurations, these formats are not officially supported going forward. Logs in unsupported formats may result in incomplete parsing, limited field extraction, and degraded dashboard and analytics functionality.

Release Notes

V1.0.0

- Provided the parser that processes data ingested from the Vectra platform and converts it into the Google SecOps UDM data model.

Prerequisites

- Vectra Platform
- Google SecOps

Vectra Google SecOps Forwarder

---

1. Setup Google Security Operations Forwarder
    a. Users must first install and configure the Google Security Operations forwarder for **Vectra Stream** log type in their environment.
    b. Refer to the below guides for detailed setup instructions.
        i. Forwarder Configuration from UI.
        ii. Install and Configure the Forwarder.
2. Configure Vectra for Syslog Forwarding
    a. Once the Google SecOps Syslog forwarder is configured, Vectra Administrator users can enable Vectra to send the host and Account scoring information, detection details, and audit logs over syslog to external collectors for further storage and analysis.

Refer to the Vectra Syslog Guide for step-by-step configuration details.

View Events in Google SecOps

---

1. Log in to Google SecOps:
    a. Open a web browser and navigate to the Google SecOps instance URL. For example: https://test.backstory.chronicle.security/
    b. Replace test with your actual Google SecOps instance name.
2. Access SIEM Search:
    a. From the top left corner of the Google SecOps console, select the "Investigation" option.
    b. Within the Investigation section, choose "SIEM Search".
3. Filter Events by Log Type:
    a. In the SIEM Search interface, locate the "UDM Search" section.
    b. Apply a filter for the metadata field "log_type". Set the filter value to metadata.log_type=VECTRA_STREAM
4. View Vectra Events:
    a. The SIEM Search results will display Vectra events within the "Events" section.

Supported Stream Metadata types

The Vectra Stream parser supports the following metadata types generated by Vectra Stream.

Metadata Type

Metadata Type	
Beacon	NTLM
DCE-RPC	RDP
DNS	Radius
DHCP	SMB Files
HTTP	SMB Mapping
ISession	SMTP
Kerberos	SSH
LDAP	SSL
Match	X509

Field mapping

Common Fields

The following table lists common fields of the STREAM log and their corresponding UDM fields.

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.product_name		This field is set to "Stream"
metadata.vendor_name		This field is set to "Vectra"
metadata.event_timestamp	ts	
metadata.porduct_log_id	uid	
metadata.product_event_type	metadata_type	
network.community_id	community_id	
observer.asset_id	sensor_uid	if "sensor_uid" contains ":" then map to this field else map this field by appending "Vectra:" to the field e.g. "Vectra:{sensor_uid}"
principal.asset_id	orig_huid	if "orig_huid" contains ":" then map to this field else map this field by appending "Vectra:" to the field e.g. "Vectra:{orig_huid}"
principal.hostname	orig_hostname	
principal.ip	id.orig_h	
principal.network.session_id	orig_sluid	
prinicpla.port	id.orig_p	
target.asset_id	resp_huid	if "resp_huid" contains ":" then map to this field else map this field by appending "Vectra:" to the field e.g. "Vectra:{resp_huid}"
target.hostname	resp_hostname	
target.ip	id.resp_h	

target.network.session_id	resp_sluid	
target.port	id.resp_p	
additional.fields [id_ip_ver]	id.ip_ver	
additional.fields [local_orig]	local_orig	
additional.fields [local_resp]	local_resp	

Beacon

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.product_log_id	uid	
network.ip_protocol	proto, protoName	
network.received_bytes	resp_ip_bytes	
network.sent_bytes	orig_ip_bytes	
network.session_duration	duration	
network.session_id	beacon_uid	
network.tls.client.ja3	ja3	
security_result.first_discovered_time	first_event_time	
security_result.last_discovered_time	last_event_time	
target.application	service	
target.domain.name	resp_domains	"resp_domains" is an array mapping each values to this field if it is first element
about.domain.name	resp_domains	"resp_domains" is an array mapping each values to this field if it is not the first element
additional.fields[beacon_type]	beacon_type	
additional.fields[session_count]	session_count	

DCE-RPC

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.description	operation	
network.application_protocol		This field is set to "DCERPC"
network.session_duration	rtt	
additional.fields[hostname]	hostname	
principal.user.userid	username	
target.application	endpoint	
target.domain.name	domain	

DNS

UDM Field Name	Raw Log Field Name	Mapping Logic
network.application_protocol		This field is set to "DNS"
network.dns.answers.name	answers	"answers" is an array mapping each values to this field
network.dns.answers.ttl	TTLs	"TTLs" is an array mapping each values to this field
network.dns.authoritative	AA	
network.dns.id	trans_id	
network.dns.questions.class	qclass	
network.dns.questions.name	query	
network.dns.questions.type	qtype	
network.dns.recursion_available	RA	
network.dns.recursion_desired	RD	
network.dns.response	rejected	if "rejected" equals to "false" then set this field to "true" else if "rejected" equals

		to "true" then set this field to "false"
network.dns.response_code	rcode	
network.dns.truncated	TC	
network.ip_protocol	proto	@include
additional.fields[auth]	auth	"auth" is an array mapping each values to auth_{index} key of this field
additional.fields[qclass_name]	qclass_name	
additional.fields[qtype_name]	qtype_name	
additional.fields[rcode_name]	rcode_name	
additional.fields[saw_query]	saw_query	
additional.fields[saw_reply]	saw_reply	
additional.fields[total_answers]	total_answers	
additional.fields[total_replies]	total_replies	

DHCP

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.event_timestamp	ts	
metadata.product_log_id	uid	
network.application_protocol		This field is set to "DHCP"
network.dhcp.chaddr	mac	
network.dhcp.client_hostname	orig_hostname	
network.dhcp.lease_time_seconds	lease_time	
network.dhcp.siaddr	dhcp_server_ip	
network.dhcp.transaction_id	trans_id	
network.dhcp.yiaddr	assigned_ip	
observer.asset_id	sensor_uid	

additional.fields [dns_server_ips]	dns_server_ips	"dns_server_ips" is an array mapping each values to dns_server_ips_{index} key of this field

HTTP

UDM Field Name	Raw Log Field Name	Mapping Logic
intermediary.ip	proxied	"proxied" is an array mapping each values to this field only if the current field value is type of an IPAddress
additional.fields[proxied]	proxied	"proxied" is an array mapping each values to proxied_{index} key of this field if the current field value is not type of an IPAddress
network.application_protocol		"network.application_protocol" => "HTTP"
network.http.referral_url	referrer	
network.http.response_code	status_code	
network.http.user_agent	user_agent	
network.received_bytes	resp_ip_bytes	
network.received_packets	resp_pkts	
network.sent_bytes	orig_ip_bytes	
network.sent_packets	orig_pkts	
principal.ip	host	if "host" is type of an IPAddress then only map to this field
target.file.names	resp_filename	
target.url	uri	

additional.fields[cookie_vars]	cookie_vars	"cookie_vars" is an array mapping each values to cookie_vars_{index} key of this field
additional.fields[cookie]	cookie	
additional.fields[host_multihomed]	host_multihomed	
additional.fields[is_proxied]	is_proxied	
additional.fields[orig_mime_types]	orig_mime_types	"orig_mime_types" is an array mapping each values to orig_mime_types_{index} key of this field
additional.fields[request_body_len]	request_body_len	
additional.fields[request_cache_control]	request_cache_control	
additional.fields[request_header_count]	request_header_count	
additional.fields[resp_mime_types]	resp_mime_types	"resp_mime_types" is an array mapping each values to resp_mime_types_{index} key of this field
additional.fields[response_body_len]	response_body_len	
additional.fields[response_cache_control]	response_cache_control	
additional.fields[response_content_disposition]	response_content_disposition	
additional.fields[response_expires]	response_expires	
additional.fields[response_header_count]	response_header_count	
additional.fields[status_msg]	status_msg	

ISession Connectivity

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.description	conn_state	if "conn_state" equals to "S0" then set this field to "S0: Connection attempt seen, no reply"

		else if "conn_state" equals to "S1" then set this field to "S1: Connection established, not terminated" else if "conn_state" equals to "S2" then set this field to "S2: Connection established and close attempt by originator seen (but no reply from responder)" else if "conn_state" equals to "S3" then set this field to "S3: Connection established and close attempt by responder seen (but no reply from originator)" else if "conn_state" equals to "SF" then set this field to "SF: Normal SYN/FIN completion" else if "conn_state" equals to "REJ" then set this field to "REJ: Connection attempt rejected" else if "conn_state" equals to "RSTO" then set this field to "RSTO: Connection established, originator aborted (sent a RST)" else if "conn_state" equals to "RSTOS0" then set this field to "RSTOS0: Originator sent a SYN followed by a RST, we never saw a

		SYN-ACK from the responder" else if "conn_state" equals to "RSTOSH" then set this field to "RSTOSH: Responder sent a SYN ACK followed by a RST, we never saw a SYN from the (purported) originator" else if "conn_state" equals to "RSTR" then set this field to "RSTR: Established, responder aborted" else if "conn_state" equals to "RSTRH" then set this field to "RSTRH: Responder sent a SYN ACK followed by a RST, we never saw a SYN from the (purported) originator." else if "conn_state" equals to "SH" then set this field to "SH: Originator sent a SYN followed by a FIN, we never saw a SYN ACK from the responder (hence the connection was 'half' open)" else if "conn_state" equals to "SHR" then set this field to "SHR: Responder sent a SYN ACK followed by a FIN, we never saw a SYN from the originator" else if "conn_state"

		equals to "OTH" then set this field to "OTH: No SYN seen, just midstream traffic (a partial connection that was not later closed)"
network.ip_protocol	proto, protoName	
network.received_bytes	resp_ip_bytes	
network.received_packets	resp_pkts	
network.sent_bytes	orig_ip_bytes	
network.sent_packets	orig_pkts	
network.session_duration	duration	
principal.application	application	
security_result.confidence_score	dir_confidence	
target.application	service	
target.domain.name	resp_domain	
additional.fields[first_orig_resp_data_pkt_time]	first_orig_resp_data_pkt_time	
additional.fields[first_orig_resp_data_pkt]	first_orig_resp_data_pkt	
additional.fields[first_orig_resp_pkt_time]	first_orig_resp_pkt_time	
additional.fields[first_resp_orig_data_pkt]	first_resp_orig_data_pkt	
additional.fields[first_resp_orig_pkt_time]	first_resp_orig_pkt_time	
additional.fields[first_resp_orig_data_pkt_time]	first_resp_orig_data_pkt_time	
additional.fields[orig_vlan_id]	orig_vlan_id	
additional.fields[resp_multihomed]	resp_multihomed	
additional.fields[resp_vlan_id]	resp_vlan_id	
additional.fields[session_start_time]	session_start_time	

Kerberos

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.description	error_code, error_msg	A combination of both values is mapped. e.g. "{error_code} - {error_msg}"
network.ip_protocol	protocol	
network.tls.cipher	rep_cipher	
principal.domain.name	client	
security_result.detection_fields[orig_host_observed_privilege]	orig_host_observed_privilege	
target.application	service	
additional.fields[data_source]	data_source	
additional.fields[reply_timestamp]	reply_timestamp	
additional.fields[req_ciphers]	req_ciphers	"req_ciphers" is an array mapping each values to req_ciphers_{index} key of this field
additional.fields[request_type]	request_type	
additional.fields[service_privilege]	service_privilege	
additional.fields[service_uid]	service_uid	
additional.fields[success]	success	

LDAP

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.description	error	
network.application_protocol		This field is set to"LDAP"
network.received_bytes	response_bytes	
network.sent_bytes	request_bytes	
network.session_duration	duration	

target.resource.attribute.labels[attributes]	attributes	"attributes" is an array mapping each values to attributes_{index} key of this field
target.resource.attribute.labels[baseObject]	baseObject	
target.resource.attribute.labels[bind_error_count]	bind_error_count	
target.resource.attribute.labels[encrypted_sasl_payload_count]	encrypted_sasl_payload_count	
target.resource.attribute.labels[Is_close]	Is_close	
target.resource.attribute.labels[Is_query]	Is_query	
target.resource.attribute.labels[matched_dn]	matched_dn	
target.resource.attribute.labels[message_id]	message_id	
target.resource.attribute.labels[query_scope]	query_scope	
target.resource.attribute.labels[result_code]	result_code	
target.resource.attribute.labels[result]	result	
target.resource.name	query	
additional.fields[logon_failure_error_count]	logon_failure_error_count	
additional.fields[result_count]	result_count	

Match

UDM Field Name	Raw Log Field Name	Mapping Logic
network.direction	eve_json.direction	if "eve_json.direction" equals to "to_server" then set this field to "OUTBOUND" else if "eve_json.direction" equals to "to_client" then set this field to "INBOUND"
network.ip_protocol	eve_json.proto	
security_result.category_details	eve_json.alert.category	"eve_json.alert.category" is an array mapping each values to this array

		field
security_result.rule_name	eve_json.alert.rule	
security_result.severity	eve_json.alert.severity	if "eve_json.alert.severity" equals to "1" then set this field to "CRITICAL" else if "eve_json.alert.severity" equals to "2" then set this field to "HIGH" else if "eve_json.alert.severity" equals to "3" then set this field to "MEDIUM"
security_result.severity_details	eve_json.alert.severity	
security_result.threat_id	eve_json.alert.signature_id	
security_result.threat_name	eve_json.alert.signature	
security_result.detection_fields[alert_gid]	eve_json.alert.gid	
security_result.detection_fields[alert_metadata_affected_product]	eve_json.alert.metadata.affected_product	
security_result.detection_fields[alert_metadata_attack_target]	eve_json.alert.metadata.attack_target	
security_result.detection_fields[alert_metadata_created_at]	eve_json.alert.metadata.created_at	
security_result.detection_fields[alert_metadata_deployment]	eve_json.alert.metadata.deployment	
security_result.detection_fields[alert_metadata_malware_family]	eve_json.alert.metadata.malware_family	
security_result.detection_fields[alert_metadata_policy]	eve_json.alert.metadata.policy	
security_result.detection_fields[alert_metadata_signature_severity]	eve_json.alert.metadata.signature_severity	
security_result.detection_fields[alert_metadata_tag]	eve_json.alert.metadata.tag	

security_result.detection_fields[alert_metadata_updated_at]	eve_json.alert.metadata.updated_at	
security_result.detection_fields[alert_rev]	eve_json.alert.rev	
security_result.detection_fields[alert_xff]	eve_json.alert.xff	
security_result.detection_fields[packet]	eve_json.packet	
security_result.detection_fields[payload_printable]	eve_json.payload_printable	
security_result.detection_fields[payload]	eve_json.payload	

NTLM

UDM Field Name	Raw Log Field Name	Mapping Logic
extensions.auth.auth_details	status	
security_result.action	success	if "success" equals to "true" then set this field to "ALLOW" else if "success" equals to "false" then set this field to "BLOCK"
security_result.action_details	success	
target.domain.name	domain	
additional.fields[hostname]	hostname	
target.user.userid	username	

RDP

UDM Field Name	Raw Log Field Name	Mapping Logic
principal.asset_id	client_dig_product_id	
additional.fields[client_name]	client_name	
network.application_protocol		This field is set to "RDP"
additional.fields[client_build]	client_build	
additional.fields[cookie]	cookie	
additional.fields[desktop_height]	desktop_height	

additional.fields[desktop_width]	desktop_width	
additional.fields[keyboard_layout]	keyboard_layout	
additional.fields[result]	result	

Radius

UDM Field Name	Raw Log Field Name	Mapping Logic
extensions.auth.auth_details	account_authentic	
intermediary.ip	tunnel_client	
intermediary.asset.product_object_id	calling_station_id	
metadata.description	radius_type	
network.ip_protocol	framed_protocol	
network.received_bytes	account_input_octets	
network.received_packets	account_input_packets	
network.sent_bytes	account_output_octets	
network.sent_packets	account_output_packets	
network.session_duration	account_session_time	
network.session_id	account_session_id	
principal.ip	framed_address	
principal.ip	framed_ip_address	
security_result.action	result	if "result" regex matches to "success" (case insensitive) then set this field to "ALLOW" else if "result" regex matches to "failed" (case insensitive) then set this field to "FAIL"
security_result.action_details	result	
target.asset_id	dst_host_luid	
target.ip	nas_ip_address	
target.mac	mac	
target.port	nas_port	

target.user.attribute.labels[nas_identifier]	nas_identifier	
target.user.userid	username	
additional.fields[account_delay_time]	account_delay_time	
additional.fields[account_input_gigawords]	account_input_gigawords	
additional.fields[account_output_gigawords]	account_output_gigawords	
additional.fields[connect_info]	connect_info	
additional.fields[delegated_ipv6_prefix]	delegated_ipv6_prefix	
additional.fields[dst_display_name]	dst_display_name	
additional.fields[dst_luid_external]	dst_luid_external	
additional.fields[dst_luid]	dst_luid	
additional.fields[event_timestamp]	event_timestamp	
additional.fields[filter_id]	filter_id	
additional.fields[framed_interface]	framed_interface	
additional.fields[framed_ipv6_prefix]	framed_ipv6_prefix	
additional.fields[idle_timeout]	idle_timeout	
additional.fields[logged]	logged	
additional.fields[nas_port_id]	nas_port_id	
additional.fields[nas_port_type]	nas_port_type	
additional.fields[password_seen]	password_seen	
additional.fields[reply_msg]	reply_msg	
additional.fields[reply_timestamp]	reply_timestamp	
additional.fields[service_type]	service_type	
additional.fields[session_timeout]	session_timeout	
additional.fields[src_display_name]	src_display_name	
additional.fields[src_host_luid]	src_host_luid	
additional.fields[src_luid_external]	src_luid_external	
additional.fields[src_luid]	src_luid	
additional.fields[ttl]	ttl	

SMB Files

UDM Field Name	Raw Log Field Name	Mapping Logic
metadata.description	action, name	A combination of both values is mapped. e.g. "action: {action} on: {name}"
metadata.event_type		if "action" is equal to "SMB::FILE_READ" then map "metadata.event_type" to "FILE_READ" else if "action" is equal to "SMB::FILE_WRITE" then map "metadata.event_type" to "FILE_MODIFICATION" else if "action" is equal to "SMB::FILE_OPEN" then map "metadata.event_type" to "FILE_OPEN" else if "action" is equal to "SMB::FILE_CLOSE" then map "metadata.event_type" to "FILE_UNCATEGORIZED" else if "action" is equal to "SMB::FILE_DELETE" then map "metadata.event_type" to "FILE_DELETION" else if "action" is equal to "SMB::FILE_RENAME" then map "metadata.event_type"

		to "FILE_MOVE" else if "action" is equal to "SMB::FILE_SET_ATTRIBUTE" then map "metadata.event_type" to "FILE_UNCATEGORIZED" else map "metadata.event_type" to "FILE_UNCATEGORIZED"
network.application_protocol_version	version	
src.file.names	prev_name	
target.application	service	
target.file.full_path	path	
target.file.names	name	
additional.fields[delete_on_close]	delete_on_close	

SMB Mapping

UDM Field Name	Raw Log Field Name	Mapping Logic
network.application_protocol_version	version	
principal.asset_id	username	If “username” does not contains “$” then map to this field.
principal.user.userid	username	If “username” contains “$” then map to this field.
target.application	service	
target.domain.name	domain	
target.file.full_path	path	
additional.fields[hostname]	hostname	

SMTP

UDM Field Name	Raw Log Field Name	Mapping Logic
network.application_protocol		This field is set to "SMTP"
network.email.cc	cc	
network.email.from	from	
network.email.mail_id	msgid	
network.email.reply_to	reply_to	
network.email.subject	subject	
network.email.to	to	
network.smtp.helo	helo	
network.smtp.is_tls	tls	
network.smtp.mail_from	mail_from	
network.smtp.rcpt_to	rcpt_to	
principal.ip	x_originating_ip	
additional.fields[date]	date	

additional.fields[dkim_status]	dkim_status	
additional.fields[dmarc_status]	dmarc_status	
additional.fields[first_received]	first_received	
additional.fields[in_reply_to]	in_reply_to	
additional.fields[second_received]	second_received	
additional.fields[spf_helo_status]	spf_helo_status	
additional.fields[spf_mailfrom_status]	spf_mailfrom_status	
additional.fields[user_agent]	user_agent	

SSH

UDM Field Name	Raw Log Field Name	Mapping Logic
network.application_protocol_version	version	
network.application_protocol		This field is set to "SSH"
network.tls.cipher	cipher_alg	
principal.application	client	
target.application	server	
additional.fields[compression_alg]	compression_alg	
additional.fields[hassh]	hassh	
additional.fields[hasshServer]	hasshServer	
additional.fields[host_key_alg]	host_key_alg	
additional.fields[host_key]	host_key	
additional.fields[kex_alg]	kex_alg	
additional.fields[mac_alg]	mac_alg	

SSL

UDM Field Name	Raw Log Field Name	Mapping Logic
network.application_protocol_version	version, version_num	
network.application_protocol		This field is set to "SSL"
network.tls.cipher	cipher	

network.tls.client.issuer	client_issuer	
network.tls.client.ja3	ja3	
network.tls.client.server_name	server_name	
network.tls.client.subject	client_subject	
network.tls.client.certificate.version	client_version, client_version_num	A combination of both values is mapped. e.g. "{client_version_num}” - “{client_version}"
network.tls.curve	curve	
network.tls.established	established	
network.tls.next_protocol	next_protocol	
network.tls.server.certificate.issuer	issuer	
network.tls.server.ja3s	ja3s	
network.tls.server.certificate.subject	subject	
network.tls.server.certificate.version	version, version_num	A combination of both values is mapped. e.g. "{version_num}” - “{version}"
network.tls.version_protocol	client_version_num	
target.application	application	
additional.fields[client_curve_num]	client_curve_num	“client_curve_num” is an array mapping each values to client_curve_num_{index} key
additional.fields[client_ec_point_format]	client_ec_point_format	“client_ec_point_format” is an array mapping each values to client_ec_point_format_{index} key
additional.fields[client_extension]	client_extension	“client_extension” is an array mapping each values to client_extension_{index}

		key
additional.fields[server_extensions]	server_extensions	“server_extensions” is an array mapping each values to server_extensions_{index} key

X509

UDM Field Name	Raw Log Field Name	Mapping Logic
network.tls.client.certificate.issuer	certificate.issuer	
network.tls.client.certificate.not_after	certificate.not_valid_after	
network.tls.client.certificate.not_before	certificate.not_valid_before	
network.tls.client.certificate.serial	certificate.serial	
network.tls.client.certificate.subject	certificate.subject	
network.tls.client.certificate.version	certificate.version	
network.tls.curve	certificate.curve	
target.application	application	
about.ip	san.ip	
about.url	san.uri	
about.user.email_addresses	san.email	
additional.fields[basic_constraints_ca]	basic_constraints.ca	
additional.fields[basic_constraints_path_len]	basic_constraints.path_len	
additional.fields[certificate_cn]	certificate.cn	
additional.fields[certificate_exponent]	certificate.exponent	
additional.fields[certificate_key_alg]	certificate.key_alg	
additional.fields[certificate_key_length]	certificate.key_length	
additional.fields[certificate_key_type]	certificate.key_type	
additional.fields[certificate_self_issued]	certificate.self_issued	
additional.fields[certificate_sig_alg]	certificate.sig_alg	

additional.fields[san_dns]	san.dns	“san.dns” is an array mapping each values to san_dns_{index} key
additional.fields[san_other_fields]	san.other_fields	

References

- Vectra Stream: Network metadata with an opinion