VECTRA

Vectra AI Platform metadata attributes and descriptions

This document describes the important attributes in all the Network metadata streams supported by Vectra AI Platform.

Common fields in all metadata streams (except DHCP)	
Field	**Description**
id.ip_ver*	IP Version
id.orig_h	Originating endpoint IP address
id.orig_p	Originating endpoint TCP/UDP port
id.resp_h	Responding endpoint IP address
id.resp_p	Responding endpoint TCP/UDP port
local_orig	Boolean indicating if connection was locally originated
local_resp	Boolean indicating if connection was locally responded
orig_hostname*	Originating endpoint hostname
orig_huid*	Unique identifier for the originating host if it is local
orig_sluid*	Unique identifier for the originating host session
resp_hostname*	Responding endpoint hostname
resp_huid*	Unique identifier for the responding host if it is local
resp_sluid*	Unique identifier for the responding host session if it is local
sensor_uid	Unique identifier for Vectra sensor that observed the underlying traffic generating the metadata record
ts	Timestamp when the metadata record is generated. It is in date format (e.g. May 9, 2018, 10:09:25.366)
uid	Unique id of connection

Beacon**	
Field	**Description**
beacon_type	The type of beacon. 'single_resp_multiple_sessions' type indicates a beacon to one destination comprising of multiple sessions
beacon_uid	The unique uid of the beacon
duration	Total duration of the BeaconUid
first_event_time	Timestamp of the first observed session for this beacon_uid
ja3	Ja3 hash of client based on client SSL parameters
last_event_time	Timestamp of the last observed session for this beacon_uid
orig_ip_bytes	Total bytes sent from originator to responder for this beacon_uid
proto	L4 protocol value. 6 is TCP, 17 is UDP
proto_name	L4 protocol name (TCP or UDP)
resp_domains	The responder domains in this event
resp_ip_bytes	Total bytes send from responder to originator for this beacon_uid
service	Service (e.g. "http" or "tls")
session_count	The number of sessions that comprise the beacon_uid
uid	The unique uid of the first connection for the reported beacon event
ts	Timestamp when the metadata record is generated. It is in date format (e.g. May 9, 2018, 10:09:25.366)
uid	Unique id of connection

*Unique to Vectra AI, not in standard Zeek output

**Beacon metadata is uniquely computed by Vectra AI platform, not in standard Zeek output

DCE-RPC

Field	Description
domain*	Domain of the host
endpoint	Endpoint name looked up from the uuid (e.g. IXnRemote, IWbemLoginClientID)
hostname*	Hostname on which the user logged in
operation	Operation seen in the call (e.g. "RemoteCreateInstance")
rtt	Round trip time of request – response
username*	Username or account name that logged in. Names ending in '$' are machine names (not user account names)

DHCP

Field	Description
assigned_ip	Assigned IP in response
dhcp_server_ip*	DHCP server IP address
dns_server_ips*	DNS server ips from DHCP options. DHCP Option 6
lease_time	DHCP lease time. DHCP Option 51
mac	MAC address in request
orig_hostname*	Hostname from DHCP options. DHCP Option 12
trans_id	Transaction id
ts	Timestamp when the metadata record is generated. It is in date format (e.g. May 9, 2018, 10:09:25.366)
uid	Unique id of connection

DNS

Field	Description
AA	Authoritative answer. True if server is authoritative for the query
answers†	List of answers to the query
auth	List of Authoritative responses for the query
proto	Protocol of DNS transaction—6 (for TCP) or 17 (for UDP)
qclass / qclass_name	Value specifying the query class (e.g. 1 / Internet [IN])
qtype / qtype_name	query type value / descriptive name (e.g. A, AAAA, PTR, TXT)
query†	Domain name subject of the query
RA	Recursion available. True if server supports recursive queries
RD	Recursion desired. True if recursive lookup of query requested
rcode / rcode_name	Response code value in the DNS response (e.g. NXDOMAIN, NODATA)
rejected	The DNS query was rejected by the server
saw_query	Whether the full DNS query has been seen
saw_reply	Whether the full DNS reply has been seen
TC	Truncation flag. True if the message was truncated
TTLs	List of TTLs from the answers
total_answers	The total number of resource records in a reply message's answer section
total_replies	The total number of resource records in a reply message's answer, authority, and additional sections
trans_id	16-bit identifier assigned by DNS client

*Unique to Vectra AI, not in standard Zeek output
† Field may contain Base64 encoded data if the metadata contains invalid UTF-8

HTTP

Field	Description
accept	Value of the Accept header in the request, if present, truncated to 256 bytes
accept_encoding	Value of the Accept-Encoding header in the request, if present, truncated to 256 bytes
cookie*	Value of the Cookie header, truncated to 256 bytes
cookie_vars*	The variables in the cookie field, without the values
host	Value of the Host header, truncated to 256 bytes
host_multihomed*	Boolean attribute that indicates whether the address in the host header is observed to be associated with one or multiple IPs
is_proxied*	Boolean value indicative of a proxied request
ja4h	The JA4H fingerprint of the HTTP client
method	HTTP Request Method
orig_ip_bytes*	Bytes sent by originator to responder
orig_mime_types	Content type header in originator request
orig_pkts*	Number of packets sent from originator to responder
post_data	Binary data of the POST request body. Truncated to 2k size
proxied	Value of x-forwarded-for header (e.g. X-FORWARDED-FOR -> 10.10.15.192)
referrer	Value of the Referrer header, truncated to 256 bytes
request_body_len	HTTP payload bytes in request
request_cache_control*	Value of the Cache-Control header in the request, if present, truncated to 256 bytes
request_header_count*	Count of headers in request
resp_filename	The name of the file returned by the server (if any)
resp_ip_bytes*	Bytes send by responder to originator
resp_mime_types	Value of the Content-Type header in response, truncated to 256 bytes
resp_pkts*	Number of packets sent from responder to originator
response_body_len	HTTP payload bytes in response
response_cache_control*	Value of the Cache-Control header in the response, if present, truncated to 256 bytes
response_content_disposition	The value of the Content-Disposition header (specifies names of the files to be downloaded as attachment, e.g. 'attachment; filename="filename.jpg"')
response_expires*	Expires header in response, if present
response_header_count*	Count of headers in response
status_code	The status code in the HTTP response
status_msg	The status message corresponding to the status code
uri	URI used in the request, truncated to 512 bytes
user_agent	Value of the User-Agent header, truncated to 512 bytes

*Unique to Vectra AI, not in standard Zeek output

iSession Connectivity

Field	Description
application	Applications associated with this session
client_luid_proxy	True if the source address of the connection has been learned as a proxy
conn_state	Connection state. Takes values: S0, S1, SF, REJ, S2, S3, RSTO, RSTR, RSTOS0, RSTRH, SH, SHR, or OTH
dir_confidence	Client/server assignment confidence from 0 to 100
duration	Duration of connection in ms
first_orig_resp_data_pkt*	Base64 encoding of the first 16 bytes of the packet from originator to responder, represented as a string
first_orig_resp_data_pkt_time*	Timestamp of first data packet from originator to responder
first_orig_resp_pkt_time*	Timestamp of first packet from originator to responder
first_resp_orig_data_pkt*	Base64 encoding of the first 16 bytes of the packet from responder to originator, represented as a string
first_resp_orig_pkt_time*	Timestamp of first packet from responder to originator
first_resp_orig_data_pkt_time*	Timestamp of first data packet from responder to originator
ja4lc	The JA4LC fingerprint of the client's light distance
ja4ls	The JA4LS fingerprint of the server's light distance
ja4t	The JA4T fingerprint of the client's TCP SYN packet
ja4ts	The JA4TS fingerprint of the server's TCP SYN ACK packet(s)
orig_ip_bytes	Bytes sent from originator to responder
orig_pkts	Number of packets sent from originator to responder
orig_vlan_id*	VLAN_id of originator, if any
proto	L4 protocol value. 6 is TCP, 17 is UDP
proto_name	L4 protocol name (TCP, UDP or ICMP)
proxy_to_internal_dst	True if effective destination after proxy is internal IP
resp_domain*	Calculated from TLS SNI, HTTP Host, or the destination IP name (in this exact order)
resp_ip_bytes	Bytes send from responder to originator
resp_multihomed*	Boolean attribute that indicates whether the domain is observed to be associated with one or multiple IPs
resp_pkts	Number of packets sent from responder to originator
resp_vlan_id*	VLAN id of responder, if any
service	Service (e.g. "smb")
server_luid_proxy	True if the destination address of the connection has been learned as a proxy
session_start_time	Timestamp when session started

Connection State Values

State	Description
SO	Connection attempt seen, no reply.
Si	Connection established, not terminated.
SF	Normal establishment and termination. Note that this is the same symbol as for state S1. You can tell the two apart because for S1 there will not be any byte counts in the summary, while for SF there will be.
REJ	Connection attempt rejected
S2	Connection established and close attempt by originator seen (but no reply from responder)
S3	Connection established and close attempt by responder seen (but no reply from originator)
RST0	Connection established, originator aborted (sent a RST)
RSTR	Responder sent a RST
RSTOS0	Originator sent a SYN followed by a RST, we never saw a SYN-ACK from the responder.
RSTRH	Responder sent a SYN ACK followed by a RST, we never saw a SYN from the (purported) originator.
SH	Originator sent a SYN followed by a FIN, we never saw a SYN ACK from the responder (hence the connection was "half" open).
SHR	Responder sent a SYN ACK followed by a FIN, we never saw a SYN from the originator.
OTH	No SYN seen, just midstream traffic (one example of this is a "partial connection" that was not later closed).

*Unique to Vectra AI, not in standard Zeek output

Kerberos

Field	Description
account_privilege	Privilege level of the account. The scores can fall in three categories – Low (1,2) Medium (3,4,5,6,7) and High (8,9)
account_uid	Account unique identifier (principal@REALM format)
as_rep_padata_count	Total PA-DATA entries seen on AS-REP prior to truncation
as_rep_padata_types	PA-DATA type integers from AS-REP messages (12 max)
as_rep_padata_types_string	Human readable PA-DATA type names for AS-REP messages (12 max)
as_req_padata_count	Total PA-DATA entries seen on AS-REQ prior to truncation
as_req_padata_types	PA-DATA type integers from AS-REQ messages (12 max)
as_req_padata_types_string	Human readable PA-DATA type names for AS-REQ messages (12 max)
client	Client name, including realm
data_source	The source of the record, either "network" or "log"
error_code	Error code if not a success
error_msg	Error message if not a success
orig_host_observed_privilege*	The privilege represents the observed privilege based on the activity of an account seen to operate from the host. The scores can fall in three categories – Low (1, 2), Medium (3, 4, 5, 6, 7) and High (8, 9)
protocol*	L4 protocol. 6 (TCP) or 17 (UDP)
rep_cipher	Cipher selected in reply message
reply_timestamp*	Timestamp of reply
req_ciphers	Ordered list of requested ciphers
request_type	Type of request (AS or TGT)
service	Service being requested, including realm
service_privilege	Privilege level of the service. The scores can fall in three categories – Low (1,2) Medium (3,4,5,6,7) and High (8,9)
service_uid	Service unique identifier (principal@REALM format)
success	Whether request was success or not
ticket_cipher	Ticket cipher observed on AS-REP and TGS-REP replies

LDAP*	
Field	**Description**
attributes	A set of attributes to request for inclusion in entries that match the search criteria and are returned
base_object	Base of the subtree in which the search is to be constrained
bind_error_count	If there are bind errors, count of the errors
duration	Duration of the session
encrypted_sasl_payload_count	If sasl encryption is used, the number of encrypted sasl payloads encountered
error	The error message in case of error (e.g. "0000208D: NameErr ...")
logon_failure_error_count	The count of logon errors
is_close	Boolean flag indicating whether the close was observed
is_query	Boolean flag indicating whether the query was observed in the request
matched_dn	The matched distinguished name
message_id	Message id
query	Criteria to use to identify which entries within the scope should be returned
query_scope	The portion of the target subtree that should be considered (e.g. wholeSubtree)
response_bytes	Number of bytes in the response
result	The result of the query in this request
request_bytes	Number of bytes in the request
result_code	The result code (success or failure) in the response
result_count	The count of the entries in the result

Match	
Field	**Description**
eve_json.alert.category	Category of the Alert Message
eve_json.alert.gid	Unique identifier for group of signatures. Defaults to 1 for most signatures.
eve_json.alert.metadata.affected_product	Specifies details on the affected product
eve_json.alert.metadata.attack_target	Specifies if the attack target is the Client, Server, Both, or Other
eve_json.alert.metadata.created_at	Specifies the date the signature was created
eve_json.alert.metadata.deployment	Specifies where the signature should be deployed
eve_json.alert.metadata.malware_family	Specifies the Malware Family that is associated with the signature
eve_json.alert.metadata.policy	Specifies details on the alert policy
eve_json.alert.metadata.signature_severity	Describes the severity associated with the signature
eve_json.alert.metadata.tag	Specifies any tag information assigned to the signature by the author
eve_json.alert.metadata.updated_at	Specifies the data of the last update to the signature
eve_json.alert.rev	Alert signature revision number indicating if the signature has been updated
eve_json.alert.rule	Specifie the rule that fired the alert
eve_json.alert.severity	Number representing the severity of the alert
eve_json.alert.signature	The rule name. Based on the 'msg' text in the signature
eve_json.alert.signature_id	Alert signature Identifier
eve_json.alert.xff	Value of x-forwarded-for
eve_json.direction	Specifies the traffic direction of the alert
eve_json.packet	Specifies the packet that triggered the signature
eve_json.payload	Provides the Base64 Encoded packet payload information
eve_json.payload_printable	Provides the payload presented in ASCII
eve_json.proto	L4 protocol name

*Unique to Vectra AI, not in standard Zeek output
** Beacon metadata is uniquely computed by Vectra AI platform, not in standard Zeek output
† Field may contain base64 encoded data if the metadata contains invalide UTF-8

NTLM

Field	Description
domain	Domain of the host
hostname	Hostname on which the user logged in
status	Status code in response
success	Whether the request was successful or not
username	Username or account name that logged in

RDP

Field	Description
client_build	RDP client version used by client machine. Will be "unknown" if encrypted
client_dig_product_id	Product ID of the client machine
client_name	Name of the client machine
cookie	Cookie value used by client machine (username)
desktop_height	Desktop height of client machine. 0 if encrypted
desktop_width	Desktop width of client machine. 0 if encrypted
keyboard_layout	Keyboard layout (language) of client machine (e.g. "US" "Encrypted Keyboard Layout")
result	If encrypted, result value is "encrypted" otherwise it will be empty

Radius

Field	Description
account_authentic	Identifies how the user was authenticated
account_delay_time	Identifies how long the sender has been trying to send the message for
account_input_gigawords	Identifies how many times the Acct-Input counter has rolled over for input
account_input_octets	How many bytes have been received
account_input_packets	How many packets the system has received
account_output_gigawords	Identifies how many times the Acct-Input counter has rolled over for output
account_output_octets	How many bytes have been set
account_output_packets	How many packets the system has sent
account_session_id	This is a unique ID that identifies the RADIUS Accounting Session which is sent in a separate packet.
account_session_time	Duration of service received by user
calling_station_id	This is the identifier of the calling station
connect_info	Identify the speed of the connection or other connection related information
delegated_ipv6_prefix	IPv6 Pool from which the IPv6 address was assigned
dst_display_name	DNS Name of the Destination
dst_host_luid	This is the ID of the destination host with host ID
dst_luid	The LUID of the RADIUS Server
dst_luid_external	Value is True if the destination is external
event_timestamp	Similar to ts but is the timestamp from the device, not from Vectra
filter_id	This identifies any ACL that is in use
framed_address	This field is available in the request that identifies the endpoint requesting authentication
framed_interface	Identifies the interface used when the user connects to the system
framed_ip_address	IP address of the endpoint device connecting to the system
framed_ipv6_prefix	Indicates the framed IPv6 prefix for the user
framed_protocol	Identifies the Framed Protocol used when the user connects to the system

Radius

Field	Description
idle_timeout	Amount of time a session can be idle before it is disconnected
logged	The boolean attribute indicates if the request was previously logged
mac	MAC Address if observed as a field in the Radius message
nas_identifier	Identifies the role the authenticating client is requesting
nas_ip_address	This is an IP Address format, it can be the IP of the Device, the Endpoint, or Intermediate system, depending on implementation
nas_port	Physical Port Number of the Device Authenticating the User
nas_port_id	Text string identifying the port provided by the client
nas_port_type	This is the type of medium of the port (e.g. Ethernet, Wifi &c.)
password_seen	Boolean attribute indicating password was seen
radius_type	The value indicates if it is an access or accounting request
reply_msg	Reply message from the server challenge. This is frequently shown to the user authenticating.
reply_timestamp	Timestamp when the reply message was received
result	Success or Failed Authentication
service_type	Type of service the user has requested
session_timeout	This is the maximum session length
src_display_name	DNS Name of the Source
src_host_luid	This is the ID of the Src with Host ID
src_luid	The LUID of the RADIUS Client
src_luid_external	Value is True if the source is external
ttl	The duration between the first request and either the "Access-Accept" message or an error. If the field is empty, it means that either the request or response was not seen.
tunnel_client	Address (IPv4, IPv6, or FQDN) of the initiator end of the tunnel, if present. This is collected from the Tunnel-Client-Endpoint attribute.
username	This is the username if observed in the Radius message

*Unique to Vectra AI, not in standard Zeek output
** Beacon metadata is uniquely computed by Vectra AI platform, not in standard Zeek output
† Field may contain base64 encoded data if the metadata contains invalide UTF-8

SMB Files

Field	Description
action	Action taken on file
delete_on_close*	Flag indicating if the delete_on_close attribute is enabled. If enabled, a file close action may delete the file if it is the last close on the file
domain*	Domain of the SMB server
hostname*	Hostname of the SMB client
path	Path pulled from the tree this file was transferred to or from
prev_name	If the rename action was seen, this will be the file's previous name
name	Filename if one was seen
username*	Username or account name that logged in. Names ending in '$' are machine names (not user account names)
version	SMB version (SMBv1 or SMBv2)

SMB Mapping

Field	Description
domain*	Domain of the SMB server
hostname*	Hostname of the SMB client
path	Name of the tree path
service	Type of re-originator of the tree
username*	Username or account name that logged in. Names ending in '$' are machine names (not user account names)
version	SMB version (SMBv1 or SMBv2)

SMTP	
Field	**Description**
cc	Contents of the CC header, formatted as a comma separated list
date	Contents of the Date header
dkim_status	pass/fail/none. Based on the 'Authentication-results' header
dmarc_status	pass/fail/none. Based on the 'Authentication-results' header
first_received	Contents of the first Received header, which signifies the first SMTP server to receive this message, (i.e. sending server)
from	Contents of the From header
helo	Contents of the Helo header
in_reply_to	Contents of the In-Reply-To header
mail_from	Email addresses found in the From header
msgid	Contents of the MsgID header
rcpt_to	Email addresses found in the Rcpt header, formatted as a comma separated list
reply_to	Contents of the ReplyTo header
second_received	Contents of the second Received header, which signifies the second SMTP server to receive this message
subject	Contents of the Subject header
spf_helo_status	Based on the 'Received-SPF' header in smtp. This header specifies the SPF status (Sender Policy Framework) One of pass/fail/neutral/softfail/none/temperror/permerror See: https://tools.ietf.org/html/rfc7208#section-9.1
spf_mailfrom_status	One of pass/fail/neutral/softfail/none/temperror/permerror
tls	Indicates that the connection has switched to using TLS
to	Contents of the To header, formatted as a comma separated list
user_agent	Value of the User-Agent header from the client
x_originating_ip	Contents of the X-Originating-IP header

SSH	
Field	**Description**
client	The client's version string
cipher_alg	The encryption algorithm in use
compression_alg	The compression algorithm in use
hassh	hassh hash of client based on client SSH parameters
hassh_server	haashServer hash of server based on client SSH parameters
host_key	The server's key fingerprint
host_key_alg	The server host key's algorithm
kex_alg	The key exchange algorithm in use
mac_alg	The signing (MAC) algorithm in use
server	The server's version string
version	SSH major version (1 or 2)

SSL

Field	Description
application	Applications associated with this session
cipher	SSL/TLS cipher suite chosen from server
client_curve_num*	Elliptical curve number sent by the client
client_ec_point_format*	Elliptical curve point format offered by the client
client_extension*	Client extensions
client_issuer	Client cert issuer
client_luid_proxy	True if the source address of the connection has been learned as a proxy
client_subject	Client cert subject
client_version*	SSL version string sent by the client
client_version_num*	SSL version number sent by the client
curve	Elliptical curve number for ECDHE
established	Flag to indicate if this ssl session has been established successfully, or if it was aborted during the handshake
issuer	Server cert issuer
ja3	JA3 hash of client based on client SSL parameters
ja3s	JA3S hash of server based on server SSL parameters
ja4	The JA4 fingerprint of the TLS client
ja4s	The JA4S fingerprint of the TLS server response
next_protocol	Next protocol the server chose using the application layer next protocol extension, if present
proxy_to_internal_dst	True if effective destination after proxy is internal ip
server_extensions	Server extensions
server_luid_proxy	True if the destination address of the connection has been learned as a proxy
server_name	SNI value
subject	Server cert subject
version	SSL/TLS version that the server chose
version_num	Numeric SSL/TLS version that the server chose
version/version_num	SSL version number

*Unique to Vectra AI, not in standard Zeek output

For more information please contact us: Email: info@vectra.ai | vectra.ai

© 2026 Vectra AI, Inc. All rights reserved. Vectra, the Vectra AI logo, and Security that thinks are registered trademarks and the Vectra Threat Labs, Threat Certainty Index and Attack Signal Intelligence are trademarks of Vectra AI. Other brand, product and service names are trademarks, registered trademarks or service marks of their respective holders.Version: 021926

X509

Field	Description
application	Applications associated with this session
basic_constraints.ca	Flag indicating whether the subject of the certificate is a CA
basic_constraints.path_len	Maximum depth of valid certification paths that include this certificate
certificate.cn	Common name that identifies the host name of the certificate
certificate.curve	Curve, if EC-certificate
certificate.exponent	Key exponent
certificate.issuer	Combination of country, organizations, common name, issuer, URI
certificate.key_alg	Name of the public key algorithm that is used in data transmission, e.g. RSA encryption
certificate.key_length	Number of bits used in the encryption, e.g. 2,048-bit encryption
certificate.key_type	Three key types, depending upon the key algorithm
certificate.not_valid_after	Time after the certificate is invalid
certificate.not_valid_before	Time before the certificate is invalid
certificate.self_issued	Boolean flag indicating whether the certificate is self-issued or backed by a CA
certificate.serial	Unique serial number given by certificate authority or certificate signed authority. Usually 40 hexadecimal characters
certificate.sig_alg	Name of the signature algorithm
certificate.subject	Owner of the certificate (distinguished name)
certificate.version	Version of the server certificate (SSL V3, TLS 1.0, TLS 1.1, etc.)
client_luid_proxy	True if the source address of the connection has been learned as a proxy
ja4x	The JA4X fingerprint of the X.509 TLS certificate
proxy_to_internal_dst	True if effective destination after proxy is internal ip
san.dns	Specifying a list of additional host names for a single certificate along with DNS names that are associated with SAN (Subject Alternative Name)
san.email	Email address associated with the SAN
san.ip	IP address of the SAN in the digital certificate
san.other_fields	Other fields in the SAN
san.uri	URL name associated with SAN
version/version_num	SSL version number
server_luid_proxy	True if the destination address of the connection has been learned as a proxy