Advanced Logging Syslog Fields

Advanced logs are generated and sent as syslog messages to the configured destinations for consumption by an external tool. The syslog fields for all advanced logging protocols are detailed here.

- Common fields, on page 1
- CONN protocol fields, on page 2
- DNS protocol fields, on page 3
- FTP fields, on page 4
- HTTP fields, on page 5
- Notice protocol fields, on page 6
- Weird protocol fields, on page 6

Common fields

These fields appear in the syslog message across all protocols:

Table 1: HTTP fields

Field	Description
id.orig_h	The client IP address involved in a connection.
id.orig_p	The client TCP or UDP port used for a connection.
id.resp_h	The server IP address involved in a connection.
id.resp_p	The server TCP or UDP port used for a connection.
pkt_num	The packet number within a network flow.
tenant_id	The identifier for a tenant associated with an event.
ts	The timestamp of the packet that triggered the log record shows when the event occurred.
uid	A unique connection ID that enables you to correlate log records related to the same network flow.

CONN protocol fields

Conn protocol fields capture various aspects of network connections, including state, duration, event history, byte counts, packet counts, transport protocol, and application service information.

Fields available in the CONN protocol for connection analysis:

Field	Description
conn_state	Captures the state of the connection based on the protocol in use. • **UDP**: States include CLT_SRV_UDP_SEEN (packets from both client and server observed), CLT_UDP_SEEN (only client packets observed), and SRV_UDP_SEEN (only server packets observed). • **TCP**: Tracks the client and server states independently using prefixes CLT_ (client) and SRV_ (server), reflecting the TCP STATE machine per RFC standards, with additional states for MID-STREAM activity (TCP_MID_STREAM_SENT, TCP_MID_STREAM_REC) and TCP_STATE_NONE. • **Other traffic**: Indicates non-UDP and non-TCP traffic or error cases.
duration	The duration of the connection, in seconds.

history	A code that indicates the event sequence of the connection.
	Each letter in the history code represents a specific event. Uppercase letters indicate client-side events, and lowercase letters indicate server-side events. Events are recorded only once per direction.
	• For UDP, events include: `d` (packet with payload).
	• For TCP, events include:
	  • `s` (SYN)
	  • `h` (SYN-ACK)
	  • `a` (pure ACK or PUSH)
	  • `d` (packet with payload)
	  • `f` (FIN)
	  • `r` (reset)

orig_bytes	The total number of TCP or UDP payload bytes transmitted by the client during the connection.
orig_pkts	The number of packets sent by the originator.
proto	The transport layer protocol of a connection, for example, IP, ICMP, TCP, or UDP.

Field	Description
resp_bytes	The total number of TCP or UDP payload bytes transmitted by the server during the connection.
resp_pkts	The number of packets sent by the responder.
service	A connection's application protocol. This value indicates the last detected service on the traffic flow.

DNS protocol fields

This reference provides detailed information about DNS protocol fields, including boolean values, resource records, transport protocols, and various identifiers used in DNS transactions.

DNS protocol fields contain specific data elements that define the behavior and content of DNS transactions:

Field	Description
AA	A boolean value that indicates whether this value is an authoritative answer (AA) to a query. For example, T or F.
addl	The list of additional responses. This list contains all the resource records (RR) found in the **addl** section. The resource record types are handled in the same way as for the **answers** field.
answers	A list of resource records that directly answer a DNS query. Each resource record provides specific information about a domain. This may include its IP address, mail server, or other properties, depending on the type of query. All resource records appear in the answers section of a DNS response. The decoding process represents each resource record by summarizing its contents. Each resource record type has specific decoding rules, depending on the type of information it represents. The following resource record types contains type-specific information when decoded—A, AAAA, BIND9 signing, CNAME, DNSKEY, DS, LOC, MX, NS, NSEC, OPT, PTR, RRSIG, SOA, SPF, SRV, SSHFP, TXT. All other resource record types are decoded by the default decoder. If the resource record type is not known or not specifically handled, it is displayed as UNKNOWN followed by the resource record type numeric value.
auth	The list of authoritative responses. This list contains all the resource records found in the **auth** section. The resource record types are handled the same way as those for the **answers** field.
proto	The transport protocol used for the DNS connection, TCP or UDP.
qclass	A 16-bit integer specifying the class of a DNS query.
qclass_name	A descriptive name for the class of a DNS query.
qtype	A 16-bit integer specifying the type of a DNS query.
qtype_name	A descriptive name for the type of a DNS query.

Field	Description
query	The domain name that is the subject of a DNS transaction.
RA	A boolean value indicating the availability of recursive query support in a server, for example, T or F.
rcode	A 16-bit integer specifying the response code to a DNS query.
rcode_name	A descriptive name for the response code to a DNS query.
RD	A boolean value indicating whether a client asked the server to pursue the query recursively, for example, T or F.
rejected	A boolean value indicating whether the server responded with an error code and no query, for example, T or F.
TC	A boolean value indicating whether a message was truncated because of UDP PDU size limits, for example, T or F.
trans_id	A 16-bit identifier assigned to a DNS query.
TTLs	The list of caching intervals for the corresponding answers. The values in the list are separated by an empty space.
Z	A 3-bit integer set to 0 unless Domain Name System Security Extensions (DNSSEC) is used.

Values in the list are separated by an empty space.

FTP fields

FTP protocol analysis relies on specific fields that capture essential command parameters, data channel information, file transfers, and server responses. Understanding these fields enables accurate interpretation of FTP activity and troubleshooting.

The following fields are commonly used during FTP protocol analysis:

Field	Description
arg	The parameters associated with an FTP command.
command	The last FTP command seen in a session.
data_channel.orig_h	The IP address of a data channel originator.
data_channel.passive	A boolean value indicating whether passive mode was used for a data channel, for example, true or false.
data_channel.resp_h	The IP address of a data channel receiving point.
data_channel.resp_p	The TCP port of a data channel receiving point.
file_size	The size of a file transferred during an FTP session.

Field	Description
reply_code	The FTP reply code from a server in response to a command.
reply_msg	The FTP reply message from a server.
user	The username used for an FTP session.

HTTP fields

This reference provides details about HTTP protocol fields that are captured and available in application event logs for analysis and monitoring purposes.

Field	Description
host	The host header from an HTTP request, indicating the target server.
info_code	The last informational status code returned by a server.
info_msg	The last informational reason phrase returned by a server.
method	The HTTP method used in a request, for example, GET or POST.
origin	The origin header from a client.
orig_filenames	List of file names sent by a client. Values in the list are separated by empty spaces.
orig_mime_types	List of the content type (MIME type) files sent by a client. Values in the list are separated by empty spaces.
proxied	List of headers associated with proxied requests. Values in a list are separated by empty spaces.
referrer	The referrer header, indicating the URL of a page that is linked to the requested resource.
request_body_len	The length of an HTTP request body (decompressed and normalized).
response_body_len	The length of an HTTP response body (decompressed and normalized).
resp_filenames	List of file names sent by a server. Values in the list are separated by empty spaces.
resp_mime_types	List of the content type files sent by a server. Values in the list are separated by empty spaces.
status_code	The HTTP status code returned by a server, for example, 200 or 404.
status_msg	The HTTP status message returned by a server, for example, OK or Not Found.
trans_depth	The number of request-response pairs seen in an HTTP session.

Field	Description
uri	The Uniform Resource Identifier (URI) from an HTTP request, specifying the resource being requested.
user_agent	The user-agent header from a client, identifying the client software.
version	The HTTP version used in a request.

Notice protocol fields

This reference provides definitions for the protocol fields that appear in notice logs when intrusion policy rules are triggered, helping you understand the information captured in security event logs.

The notice protocol fields provide detailed information about intrusion events. These fields include:

Field	Description
action	The intrusion policy action that was configured for the triggered intrusion policy rule, for example alert, drop, or pass.
gid	The GID of the intrusion rule that triggered the log. This value is collected and displayed even if the rule is disabled.
msg	The message associated with the intrusion rule that triggered the log. This field provides a description of why the flow was logged.
proto	The transport layer protocol associated with the event, for example, IP, ICMP, TCP, or UDP.
rev	The revision number of the intrusion rule that was triggered.
refs	A list of references (URLs) associated with the intrusion rule. These references provide additional information about the specific threat or vulnerability the rule is designed to detect. The references are expanded to full URLs in the log.
sid	The SID of the intrusion rule that triggered the log. This value is collected and displayed even if the rule is disabled.
source	The name of the inspector module that was assigned to process the flow. This identifies the specific component within Snort that detected the anomaly. The name of the module appears similar to that in the Network Analysis Policy.

Weird protocol fields

This reference describes the various protocol fields that can appear in intrusion detection logs, including rule identifiers, messages, transport protocols, and inspector modules.

Field	Description
gid	The GID of the intrusion rule that triggered the log. This value is collected and displayed even if the rule is disabled.
msg	The message associated with the intrusion rule that triggered the log. This field provides a description of why the flow was logged.
proto	The transport layer protocol associated with the event, for example, IP, ICMP, TCP, or UDP.
sid	The SID of the intrusion rule that triggered the log. This value is collected and displayed even if the rule is disabled.
source	The name of the inspector module that was assigned to process the flow. This identifies the specific component within Snort that detected the anomaly. The name of the module appears similar to that in the Network Analysis Policy.