Bro Logs

conn.log

IP, TCP, UDP and ICMP connection details

Field	Type	Description
ts	time	Timestamp of the first packet
uid	string	Unique ID of the connection
id.orig_h	addr	Originating endpoint's IP address (AKA Orig)
id.orig_p	port	Originating endpoint's TCP/UDP port (or ICMP code)
id.resp_h	addr	Responding endpoint's IP address (AKA Resp)
id.resp_p	port	Responding endpoint's TCP/UDP port (or ICMP code)
proto	proto	Transport layer protocol of connection
service	string	Detected application protocol, if any
duration	interval	Connection length
orig_bytes	count	Orig payload bytes; from sequence numbers if TCP
resp_bytes	count	Resp payload bytes; from sequence numbers if TCP
conn_state	string	Connection state (see **conn.log: conn_state** table)
local_orig	bool	Is Orig in Site::local_nets? Unset if local_nets is empty.
local_resp	bool	Is Resp in Site::local_nets? Unset if local_nets is empty.
missed_bytes	count	Number of bytes missing due to content gaps
history	string	Connection state history (see **conn.log: history** table)
orig_pkts	count	Number of Orig packets
orig_ip_bytes	count	Number of Orig IP bytes (via IP total_length header field)
resp_pkts	count	Number of Resp packets
resp_ip_bytes	count	Number of Resp IP bytes (via IP total_length header field)
tunnel_parents	set	If tunneled, connection UID of encapsulating parent(s)
orig_l2_addr	string	Link-layer address of the originator
resp_l2_addr	string	Link-layer address of the responder
vlan	int	The outer VLAN for this connection
inner_vlan	int	The inner VLAN for this connection

dns.log

DNS query/response details

Field	Type	Description
ts	time	Timestamp of the DNS request
uid & id		Underlying connection info - See **conn.log**
proto	proto	Protocol of DNS transaction – TCP or UDP
trans_id	count	16 bit identifier assigned by DNS client; responses match
rtt	interval	Round trip time for the query and response
query	string	Domain name subject of the query
qclass	count	Value specifying the query class
qclass_name	string	Descriptive name of the query class (*e.g. C_INTERNET*)
qtype	count	Value specifying the query type
qtype_name	string	Descriptive name of the query type (*e.g. A, AAAA, PTR*)
rcode	count	Response code value in the DNS response
rcode_name	string	Descriptive name of response code (*e.g. NXDOMAIN, NODATA*)
AA	bool	Authoritateive Answer. T = server is authoritative for the query
TC	bool	Truncation. T = the message was truncated
RD	bool	Recursion Desired. T = recursive lookup of query requested
RA	bool	Recursion Available. T = server supports recursive queries
Z	count	Reserved field, should be zero in all queries & responses
answers	vector	List of resource descriptions in answer to the query
TTLs	vector	Caching intervals of the answers
rejected	bool	Whether the DNS query was rejected by the server
auth[1]	set	Authoritative responses for the query
addl[1]	set	Additional responses for the query

[1] – *If policy/protocols/dns/auth-addl.bro is loaded*

conn.log: conn_state

State	Meaning
S0	Connection attempt seen, no reply
S1	Connection established, not terminated (0 byte counts)
SF	Normal establish & termination (>0 byte counts)
REJ	Connection attempt rejected
S2	Established, Orig attempts close, no reply from RESP.
S3	Established, Resp attempts close, no reply from ORIG.
RSTO	Established, Orig aborted (RST)
RSTR	Established, Resp aborted (RST)
RSTOS0	Orig sent SYN then RST; no Resp SYN-ACK
RSTRH	Resp sent SYN-ACK then RST; no Orig SYN
SH	Orig sent SYN then FIN; no Resp SYN-ACK ("half-open")
SHR	Resp sent SYN-ACK then FIN; no Orig SYN
OTH	No SYN, not closed. Midstream traffic. Partial connection.

conn.log: history

Orig UPPERCASE, Resp lowercase, uniq-ed

Letter	Meaning
S	a **S**YN without the ACK bit set
H	a SYN-ACK ("**h**andshake")
A	a pure **A**CK
D	packet with payload ("**d**ata")
F	packet with **F**IN bit set
R	packet with **R**ST bit set
C	packet with a bad **c**hecksum
I	**i**nconsistent packet (Both SYN & RST)
Q	multi-flag packet (SYN & FIN or SYN + RST)

capture_loss.log

Estimate of packet loss

Field	Type	Description
ts	time	Timestamp of the end of the measurement
ts_delta	interval	Time difference from previous measurement
peer	string	Name of the Bro instance reporting loss
gaps	count	ACKs seen without seeing the data being ACKed
acks	count	Total number of TCP ACKs
percent_loss	double	Estimate of loss: gaps/acks

irc.log

IRC communication details

Field	Type	Description
ts	time	Timestamp of the IRC command
uid & id		Underlying connection info - See conn.log
nick	string	Nickname given for this connection
user	string	Username given for this connection
command	string	Command given by the client
value	string	Value for the command given by the client
addl	string	Any additional data for the command
fuid[1]	string	File unique ID

[1] – *If base/protocols/irc/files/bro is loaded*

Note: *base/protocols/irc/dcc-send.bro adds several DCC-related fields*

Bro Logs

files.log

File analysis results

Field	Type	Description
ts	time	Timestamp when file was first seen
fuid	string	Unique identifier for a single file
tx_hosts	set	Host(s) that sourced the data
rx_hosts	set	Host(s) that received the data
conn_uids	set	Connection UID(s) over which the file was transferred
source	string	An identification of the source of the file data
depth	count	Depth of file related to source (*e.g. HTTP request depth*)
analyzers	set	Set of analyzers attached during the file analysis
mime_type	string	The file type, as determined by Bro's signatures
filename	string	The filename, if available from the source analyzer
duration	interval	The duration that the file was analyzed for
local_orig	bool	Did the data originate locally?
is_orig	bool	Was the file sent by Orig?
seen_bytes	count	Number of bytes provided to the file analysis engine
total_bytes	count	Total number of bytes that should comprise the file
missing_bytes	count	Number of bytes in the file stream that were missed
overflow_bytes	count	Out-of-sequence bytes in the stream due to overflow
timedout	bool	If the file analysis timed out at least once
parent_fuid	string	Container file ID that this one was extracted from
md5/sha1/sha256[1]	string	MD5/SHA1/SHA256 hash of the file
extracted[2]	string	Local filename of extracted files, if enabled
entropy	double	Information density of the contents of the file

[1] – *If base/files/hash/main.bro is loaded*
[2] – *If base/files/extract/main.bro is loaded*

ftp.log

FTP request/reply details

Field	Type	Description
ts	time	Timestamp of the FTP command
uid & id		Underlying connection info - See **conn.log**
user	string	Username for the FTP session
password	string	Password for the FTP session
command	string	Command issued by the client
arg	string	Any command arguments
mime_type	string	File type if there's a file transfer
file_size	count	Size of transferred file
reply_code	count	Reply code from server in response to the command
reply_msg	string	Reply message from server in response to the command
data_channel	record	Information about the data channel (orig, resp, is passive)
fuid[1]	string	File unique ID

[1] – *If base/protocols/ftp/files.bro is loaded*

dhcp.log

DHCP lease activity

Field	Type	Description
ts	time	Timestamp of the DHCP lease request
uid & id		Underlying connection info - See conn.log
mac	string	Client's hardware address
assigned_ip	addr	Client's actual assigned IP address
lease_time	interval	IP address lease time
trans_id	count	Identifier assigned by the client; responses match

http.log

HTTP request/reply details

Field	Type	Description
ts	time	Timestamp of the HTTP request
uid & id		Underlying connection info - See **conn.log**
trans_depth	count	Pipelined depth into the connection
method	string	HTTP Request verb: GET, POST, HEAD, etc.
host	string	Value of the Host header
uri	string	URI used in the request
referrer	string	Value of the "Referer" header
user_agent	string	Value of the User-Agent header
request_body_len	count	Uncompressed content size of Orig data
response_body_len	count	Uncompressed content size of Resp data
status_code	count	Status code returned by the server
status_msg	string	Status message returned by the server
info_code	count	Last seen 1xx info reply code by server
info_msg	string	Last seen 1xx info reply message by server
tags	set	Indicators of various attributes discovered
username	string	Username if basic-auth is performed
password	string	Password if basic-auth is performed
proxied	set	Headers indicative of a proxied request
orig_fuids[1]	vector	File unique IDs from Orig
orig_filenames	vector	File names from Orig
orig_mime_types[1]	vector	File types from Orig
resp_fuids[1]	vector	File unique IDs from Resp
resp_filenames	vector	File names from Resp
resp_mime_types[1]	vector	File types from Resp
client_header_names[2]	vector	The names of HTTP headers sent by Orig
server_header_names[2]	vector	The names of HTTP headers sent by Resp
cookie_vars[3]	vector	Variable names extracted from cookies
uri_vars[3]	vector	Variable names extracted from the URI

[1] – *If base/protocols/http/entities.bro is loaded*
[2] – *If policy/protocols/http/header-names.bro is loaded*
[3] – *If policy/protocols/http/var-extraction-uri.bro is loaded*

intel.log

Hits on indicators from the intel framework

Field	Type	Description
ts	time	Timestamp of the intelligence hit
uid & id		Underlying connection info - See **conn.log**
fuid	string	The UID for a file associated with this hit, if any
file_mime_type	string	A mime type if the hit is related to a file
file_desc	string	Additional context for file, if available
seen.indicator	string	The intelligence indicator
seen.indicator_type	string	The type of data the indicator represents
seen.where	string	Where the data was discovered
seen.node	string	The name of the node that discovered the match
sources	set	Sources which supplied data for this match

tunnel.log

Details of encapsulating tunnels

Field	Type	Description
ts	time	Timestamp tunnel was detected
uid & id		Underlying connection info - See conn.log
tunnel_type	string	The type of tunnel (e.g. Teredo, IP)
action	string	The activity that occurred (discovered, closed)

Bro Logs

notice.log

Logged notices

Field	Type	Description
ts	time	Timestamp of the notice
uid & id		Underlying connection info - See **conn.log**
fuid	string	File unique ID, if this notice relates to a file
file_mime_type	string	File type, as determined by Bro's signatures
file_desc	string	Additional context for the file, if available
proto	proto	Transport protocol
note	string	The type of the notice (*e.g. SSL::Weak_Key*)
msg	string	Human readable message for the notice
sub	string	Sub-message for the notice
src	addr	Source address
dst	addr	Destination address
p	port	Associated port, if any
n	count	Associated count or status code
peer_descr	string	Name of the node that raised this notice
actions	set	Actions applied to this notice
suppress_for	interval	Length of time dupes should be suppressed
dropped[1]	bool	If the src IP was blocked
remote_location[2]	geo_location	GeoIP data about the hosts involved

[1] – *If base/frameworks/notice/actions/drop.bro is loaded*

[2] – *If base/frameworks/notice/actions/add-geodata.bro is loaded*

radius.log

RADIUS authentication attempts

Field	Type	Description
ts	time	Timestamp of the authentication attempt
uid & id		Underlying connection info - See **conn.log**
username	string	The username of the user attempting to authenticate
mac	string	The MAC address of the client (e.g. for wireless)
remote_ip	addr	The IP address of the client (e.g. for VPN)
connect_info	string	Additional connect information, if available
result	string	Whether the attempt succeeded or failed

smtp.log

SMTP transactions

Field	Type	Description
ts	time	Timestamp when the message was first seen
uid & id		Underlying connection info - See **conn.log**
trans_depth	count	Transaction depth if there are multiple msgs
helo	string	Contents of the HELO header
mailfrom	string	Contents of the MAIL FROM header
rcptto	set	Contents of the RCPT TO header
date	string	Contents of the DATE header
from	string	Contents of the FROM header
to	set	Contents of the TO header
cc	set	Contents of the CC header
reply_to	string	Contents of the ReplyTo header
msg_id	string	Contents of the MsgID header
in_reply_to	string	Contents of the In-Reply-To header
subject	string	Contents of the Subject header
x_originating_ip	addr	Contents of the X-Originating-IP header
first_received	string	Contents of the first Received header
second_received	string	Contents of the second Received header
last_reply	string	Last server to client message
path	vector	Message transmission path, from headers
user_agent	string	Value of the client User-Agent header
tls	bool	Indicates the connection switched to TLS
fuids[1]	vector	File unique IDs seen attached to this message
is_webmail[2]	bool	If the message was sent via webmail

[1] – *If base/protocols/smtp/files.bro is loaded*

[2] – *If policy/protocols/smtp/software.bro is loaded*

weird.log

Anomalies and protocol violations

Field	Type	Description
ts	time	Timestamp of message
uid & id		Underlying connection info - See **conn.log**
name	string	The name of the weird that occurred
addl	string	Additional information accompanying the weird, if any
notice	bool	Indicate if this weird was also turned into a notice
peer	string	The peer that generated this weird

snmp.log

SNMP messages

Field	Type	Description
ts	time	Timestamp when the message was first seen
uid & id		Underlying connection info - See conn.log
duration	interval	Time between the first and last seen packet
version	string	SNMP version (v1, v2c, v3)
community	string	The community string of the first SNMP packet
get_requests	count	Number of GetRequest/GetNextRequest packets
get_bulk_requests	count	Number of GetBulkRequest packets
get_responses	count	Number of GetResponse/Response packets
set_requests	count	Number of SetRequest packets
display_string	string	A system description of Resp
up_since	time	Timestamp that Resp has been up since

socks.log

SOCKS proxy requests

Field	Type	Description
ts	time	Timestamp of the SOCKS proxy request
uid & id		Underlying connection info - See **conn.log**
version	count	SOCKS protocol version
user	string	Username for proxy auth, if available
password	string	Password for proxy auth, if available
status	string	Server status for the proxy request
request.host	addr	Client requested address
request.name	string	Client requested name
request_p	port	Client requested port
bound.host	addr	Server bound address
bound.name	string	Server bound name
bound_p	port	Server bound port

software.log

Software identified by the software framework

Field	Type	Description
ts	time	Timestamp of the first software detection
host	addr	IP address running the software
host_p	port	Port on which the software is running (for servers)
software_type	Software::Type	Type of software (e.g. HTTP::SERVER)
name	string	Name of the software
version	Software::Version	Version of the software
unparsed_version	string	The full, unparsed version of the software
url[1]	string	Root URL where the software was found

[1] – *If policy/protocols/http/detect-webapps.bro is loaded*

Bro Logs

ssh.log

SSH handshakes

Field	Type	Description
ts	time	Timestamp when the SSH conn was detected
uid & id		Underlying connection info - See **conn.log**
version	count	SSH major version (1 or 2)
auth_success	bool	Did the auth succeed? Unset if undetermined
direction	Direction	Inbound or outbound connection
client	string	Software string from the client
server	string	Software string from the server
cipher_alg	string	The negotiated encryption algorithm
mac_alg	string	The negotiated MAC (signing) algorithm
compression_alg	string	The negotiated compression algorithm
kex_alg	string	The negotiated key exchange algorithm
host_key_alg	string	The server's host key algorithm
host_key	string	The server's host key fingerprint
remote_location[1]	geo_location	GeoIP data for the "remote" endpoint

[1] – *If policy/protocols/ssh/geo-data.bro is loaded*

ssl.log

SSL handshakes

Field	Type	Description
ts	time	Timestamp when the SSL connection was detected
uid & id		Underlying connection info - See **conn.log**
version	string	SSL version that the server offered
cipher	string	SSL cipher suite that the server chose
curve	string	Elliptic curve the server chose if using ECDH/ECDHE
server_name	string	Value of the Server Name Indicator SSL extension
session_id	string	Session ID offered by client for session resumption
resumed	bool	Flag that indicates the session was resumed
last_alert	string	Last alert that was seen during the connection
next_protocol	string	Next protocol the server chose using the application layer next protocol extension, if seen.
established	bool	Was this connection established successfully?
cert_chain[1]	vector	Chain of certificates offered by the server
cert_chain_fuids[1]	vector	File UIDs for certs in **cert_chain**.
client_cert_chain[1]	vector	Chain of certificates offered by the client
client_cert_chain _fuids[1]	vector	File UIDs for certs in **client_cert_chain**.
subject[1]	string	Subject of the X.509 cert offered by the server
issuer[1]	string	Subject of the signer of the server cert
client_subject[1]	string	Subject of the X.509 cert offered by the client
client_issuer[1]	string	Subject of the signer of the client cert
validation_status[2]	string	Certificate validation result for this handshake
ocsp_status[2]	string	OCSP validation result for this handshake
ocsp_response[2]	string	OCSP response as a string
notary[3]	CertNotary:: Response	A response from the ICSI certificate notary.

[1] – *If base/protocols/ssl/files.bro is loaded*

[2] – *If policy/protocols/ssl/validate-certs.bro is loaded*

[3] – *If policy/protocols/ssl/notary.bro is loaded*

kerberos.log

Kerberos authentication activity

Field	Type	Description
ts	time	Timestamp for when activity occurred
uid & id		Underlying connection info - See **conn.log**
request_type	string	Authentication Service or Ticket Granting Service
client	string	Client
service	string	Service
success	bool	Request result
error_msg	string	Error message
from	time	Ticket valid from
till	time	Ticket valid till
cipher	string	Ticket encryption type
forwardable	bool	Forwardable ticket requested
renewable	bool	Renewable ticket requested
client_cert_subject	string	Subject of X.509 cert offered by client for PKINIT
client_cert_fuid	srting	File UID for X.509 client cert for PKINIT auth
server_cert_subject	string	Subject of X.509 cert offered by server for PKINIT
server_cert_fuid	string	File UID for X.509 server cert for PKINIT auth

x509.log

SSL certificate details

Field	Type	Description
ts	time	Time when the cert was seen
id	string	File unique ID
certificate.version	count	Cert version number
certificate.serial	string	Cert serial number
certificate.subject	string	Cert subject
certificate.issuer	string	Cert issuer
certificate.not_valid_before	time	Time the cert is valid from
certificate.not_valid_after	time	Time the cert is valid until
certificate.key_alg	string	Name of the key algorithm
certificate.sig_alg	string	Name of the signature algorithm
certificate.key_type	string	Key type (RSA, DSA or EC)
certificate.key_length	count	Key length, in bits
certificate.exponent	string	Exponent, if RSA
certificate.curve	string	Curve, if EC
san.dns	string_vec	List of DNS entries in Subject Alternative Name (SAN)
san.uri	string_vec	List of URI entries in SAN
san.email	string_vec	List of email entries in SAN
san.ip	addr_vec	List of IP entries in SAN
basic_constraints.ca	bool	CA flag set?
basic_constraints.path_len	count	Maximum path length

Other Logs

The remaining log files may be found at:
www.bro.org/sphinx-git/script-reference/log-files.html

Attribution

This work was originally created by Broala, LLC.
It has been modified by NCSA.