Bro Logs: a selection

These cheat sheets document a subset of the most important logs from Bro release version 2.5. To learn about enterprise solutions from the creators of Bro, visit corelight.com.

conn.log | IP, TCP, UDP, 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 (Orig)
id.orig_p	port	Originating endpoint's TCP/UDP port (or ICMP code)
id.resp_h	addr	Responding endpoint's IP address (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**)
local_orig	bool	Is Orig in Site::local_nets?
local_resp	bool	Is Resp in Site::local_nets?
missed_bytes	count	Number of bytes missing due to content gaps
history	string	Connection state history (see **conn.log** > **history**)
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

conn_state

A summarized state for each connection

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.

history

Orig UPPERCASE, Resp lowercase, uniq-ed

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)
T	Re**t**ransmitted packet
^	Flipped connection

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 client; responses match

BroBox One

Designed by the creators of open source Bro, Corelight's BroBox One is a turn-key appliance optimized for performance and integration.

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	Authoritative 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 and responses
answers	vector	List of resource descriptions in answer to the query
TTLs	vector	Caching intervals of the answers
rejected	bool	Whether DNS query was rejected by 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

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 file 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 file analysis
mime_type	string	File type, as determined by Bro's signatures
filename	string	Filename, if available from 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 the Originator?
seen_bytes	count	Number of bytes provided to file analysis engine
total_bytes	count	Total number of bytes that should comprise the file
missing_bytes	count	Number of bytes in file stream 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 this 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 file contents

[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

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

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

kerberos.log | Kerberos authentication

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_code	count	Error code
error_msg	string	Error message
from	time	Ticket valid from
till	time	Ticket valid until
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	string	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

BRO

Bro is the world's most powerful framework for transforming network traffic into actionable data. Thousands of organizations rely on Bro every day for incident response, forensics, threat hunting, and network traffic analysis.

mysql.log | MySQL

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid & id		Underlying connection info > **See conn.log**
cmd	string	The command that was issued
arg	string	The argument issued to the command
success	bool	Server replies command succeeded?
rows	count	The number of affected rows, if any
response	string	Server message, if any

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

sip.log | SIP analysis

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when the request happened
uid & id		Underlying connection info > **See conn.log**
trans_depth	count	Pipelined depth into request/response transaction connection
method	string	Verb used in the SIP request (INVITE, etc)
uri	string	URI used in the request
date	string	Contents of Date: header from client
request_from	string	Contents of request From: header[1]
request_to	string	Contents of To: header
response_from	string	Contents of response From: header[1]
response_to	string	Contents of response To: header
reply_to	string	Contents of Reply-To: header
call_id	string	Contents of Call-ID: header from client
seq	string	Contents of CSeq: header from client
subject	string	Contents of Subject: header from client
request_path	vector	Client message transmission path, extracted from headers
response_path	vector	Server message transmission path, extracted from headers
user_agent	string	Contents of User-Agent: header from client
status_code	count	Status code returned by the server
status_msg	string	Status message returned by the server
warning	string	Contents of Warning: header
request_body_len	count	Content-Length: header from client
response_body_len	count	Content-Length: header from server
content_type	string	Content-Type: header from server

[1] The tag=value that's usually appended to the sender is stripped off and not logged

smtp.log | SMTP transactions

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when 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 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*

ssh.log | SSH handshakes

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when 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*

For the most recent version of this document, visit:

https://github.com/corelight/bro-cheatsheets

ssl.log | SSL handshakes

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when SSL connection 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 server chose if using ECDH/ ECDHE
server_name	string	Value of 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 server chose using application layer next protocol extension, if seen
established	bool	Was this connection established successfully?
cert_chain[1]	vector	Chain of certificates offered by server
cert_chain_fuids[1]	vector	File UIDs for certs in **cert_chain**
client_cert_chain[1]	vector	Chain of certificates offered by 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 server
issuer[1]	string	Subject of the signer of the server cert
client_subject[1]	string	Subject of the X.509 cert offered by 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]	Cert Notary:: 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*

syslog.log | Syslog messages

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp when syslog message was seen
uid & id		Underlying connection info > **See conn.log**
proto	trans-port_ proto	Protocol over which the message was seen
facility	string	Syslog facility for the message
severity	string	Syslog severity for the message
message	string	The plain text message

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)

Microsoft Logs

Bro version 2.5

Critical business depends on Microsoft protocols, and now you can finally have **visibility** into what's happening at the network layer for these connections.

In version 2.5, Bro has a completely rewritten analyzer for SMB and related protocols. This page collects the most critical Microsoft and SMB related logs for quick reference.

DCE RPC

Distributed Computing Environment/Remote Procedure Calls: this log shows Windows systems using other Windows systems to perform tasks such as user management, remote task execution, and general system management.

NTLM

NT Lan Manager: this log shows authentication attempts over SMB and several other protocols.

RDP

Remote Desktop Protocol: this log shows information about RDP connections. If the session is over an unencrypted connection, you will see more detailed information like keyboard layout and screen resolution.

SMB FILES

This log indicates that Bro saw the presence of a file in a SMB connection and contains metadata about the file such as time-stamps and size. Transferred files will be recorded in **files.log**.

SMB MAPPING

This log contains details of shares that are mapped over SMB. This can include user drive or other administrative share mapping and includes details like share type and service.

dce_rpc.log | Details on DCE/RPC messages

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
rtt	interval	Round trip time from the request to the response (if either the request or response wasn't seen, this will be null)
named_pipe	string	Remote pipe name
endpoint	string	Endpoint name looked up from the uuid
operation	string	Operation seen in the call

ntlm.log | NT LAN Manager (NTLM)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
username	string	Username given by the client
hostname	string	Hostname given by the client
domainname	string	Domainname given by the client
success	bool	Indicate whether or not the authentication was successful
status	string	String representation of status code returned in response to authentication attempt
done	bool	Internally used field to indicate if the login attempt has already been logged

rdp.log | Remote Desktop Protocol (RDP)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when the event happened
uid	string	Unique ID for the connection
id	conn_id	The connection's 4-tuple of endpoint addresses/ports
cookie	string	Cookie value used by client machine (username)
result	string	Status result for the connection. It's a mix between RDP negotiation failure messages and GCC server create response messages.
security_protocol	string	Security protocol chosen by server
keyboard_layout	string	Keyboard layout (language) of client machine
client_build	string	RDP client version used by client machine
client_name	string	Name of client machine

client_dig_ product_id	string	Product ID of client machine
desktop_width	count	Desktop width of client machine
desktop_height	count	Desktop height of client machine
requested_color_ depth	string	The color depth requested by the client
cert_type	string	If the connection is being encrypted with native RDP encryption, this is the type of cert being used
cert_count	count	The number of certs seen: X.509 can transfer an entire certificate chain
cert_permanent	bool	Indicates if the provided certificate or certificate chain is permanent or temporary
encryption_level	string	Encryption level of the connection
encryption_ method	string	Encryption method of the connection
analyzer_id	count	The analyzer ID used for the analyzer instance attached to each connection. Not used for logging since it's an arbitrary number.
done	bool	Track status of logging RDP connections
ssl[1]	bool	Flag the connection if it was seen over SSL

[1] *Present if policy/protocols/rdp/indicate_ssl.bro is loaded*

smb_files.log | Details on SMB files

FIELD	TYPE	DESCRIPTION
ts	time	Time when the file was first discovered
uid	string	Unique ID of the connection the file was sent over
id	conn_id	ID of the connection the file was sent over
fuid	string	Unique ID of the file
action	SMB:: Action	Action this log record represents
path	string	Path pulled from the tree this file was transferred to or from
name	string	Filename if one was seen
size	count	Total size of the file
prev_name	string	If the rename action was seen, this will be the file's previous name
times	SMB:: MACTimes	Last time this file was modified
fid	count	ID referencing this file
uuid	string	UUID referencing this file if DCE/RPC

smb_mapping.log | SMB mappings

FIELD	TYPE	DESCRIPTION
ts	time	Time when the tree was mapped
uid	string	Unique ID of the connection the tree was mapped over
id	conn_id	ID of the connection the tree was mapped over
path	string	Name of the tree path
service	string	The type of resource of the tree (disk share, printer share, named pipe, etc)
native_file_system	string	File system of the tree
share_type	string	If this is SMB2, a share type will be included. For SMB1, the type of share will be deduced and included as well.

ILLUMINATE YOUR NETWORK

Contact:

info@corelight.com

510-281-0760

corelight.com

bro.org