

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	vector	File unique IDs from Orig
orig_filenames	vector	File names from Orig
orig_mime_types	vector	File types from Orig
resp_fuids	vector	File unique IDs from Resp
resp_filenames	vector	File names from Resp
resp_mime_types	vector	File types from Resp
client_header_names ¹	vector	The names of HTTP headers sent by Orig
server_header_names ¹	vector	The names of HTTP headers sent by Resp
cookie_vars ²	vector	Variable names extracted from cookies
uri_vars ²	vector	Variable names extracted from the URI

¹If policy/protocols/http/header-names.bro is loaded

²If policy/protocols/http/var-extraction-uri.bro is loaded

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

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

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

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

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

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

¹If *policy/protocols/dns/auth-addl.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_location¹	geo_location	GeoIP data for the "remote" endpoint

¹If policy/protocols/ssh/geo-data.bro is loaded

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
		If this is SMB2, a share type will be included.

modbus.log | PLC requests (ICS)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the PLC request
uid & id		Underlying connection info - See conn.log
func	string	Function message that was sent
exception	string	Exception if there was a failure

syslog.log | Syslog messages

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

capture_loss.log | Packet loss estimate

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

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)

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

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 msgsg
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	vector	File unique IDs seen attached to message
is_webmail¹	bool	If the message was sent via webmail

¹If policy/protocols/smtp/software.bro is loaded

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

¹If policy/protocols/snmp/asn1-base.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

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: header¹
request_to	string	Contents of To: header
response_from	string	Contents of response From: header¹
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

dnp3.log | Distributed Network Protocol (ICS)

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the DNP3 request
uid & id		Underlying connection info - See conn.log
fc_request	string	The name of the request function message
fc_reply	string	The name of the reply function message

¹If policy/protocols/dnp3/asn1-base.bro is loaded

intel.log | Hits on indicators from 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	Name of the node that discovered the match
sources	set	Sources which supplied data for this match

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	A sequence of timestamps for the file's MAC times

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

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
suppressed¹	bool	If the src IP was blocked
note_location¹	geo_location	GeoIP data about the hosts involved

base/files/hash/main.bro is loaded

base/files/iptables/main.bro is loaded

kerberos.log | Kerberos authentication

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp for when activity occurred
uid & id		Underlying connection info - See conn.log

request_type	
client	
service	
success	
error_code	
error_msg	
from	
till	
cipher	
forwardable	
renewable	
client_cert_subj	
client_cert_fuid	
server_cert_subj	
server_cert_fuid	

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the first successful request
host	addr	IP address running the software (for servers)
host_p	port	Port on which the software (for servers)
software_type	Software:Type	Type of software (e.g. HTTP)
name	string	Name of the software
version	Software:Version	Version of the software
unparsed_version	string	The full, unparsed version
url¹	string	Root URL where the software is running

software.log | Software framework

FIELD	TYPE	DESCRIPTION
ts	time	Timestamp of the first successful request
host	addr	IP address running the software (for servers)
host_p	port	Port on which the software (for servers)
software_type	Software:Type	Type of software (e.g. HTTP)
name	string	Name of the software
version	Software:Version	Version of the software
unparsed_version	string	The full, unparsed version
url¹	string	Root URL where the software is running

¹ If policy/protocols/http/detect-webapps.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
irce	string	An identification of the source of the file data
sth	count	Depth of file related to source (e.g., HTTP request depth)
analyzers	set	Set of analyzers attached during file analysis
ne_type	string	File type, as determined by Bro's signatures
name	string	Filename, if available from source analyzer
ration	interval	The duration that the file was analyzed for
al_orig	bool	Did the data originate locally?
xrig	bool	Was the file sent by the Originator?
in_bytes	count	Number of bytes provided to file analysis engine
al_bytes	count	Total number of bytes that should comprise the file
missing_bytes	count	Number of bytes in file stream missed
irflow_bytes	count	Out-of-sequence bytes in the stream due to overflow
redout	bool	If the file analysis timed out at least once
parent_fuid	string	Container file ID this was extracted from
l5sha1	string	MD5/SHA1 hash of the file
racted	string	Local filename of extracted files, if enabled
entropy	double	Information density of the file contents
racted_cutoff	bool	Set to true if the file being extracted was cut off so the whole file was not logged
racted_size	count	The number of bytes extracted to disk

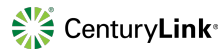
x509.log | SSL certificate details

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

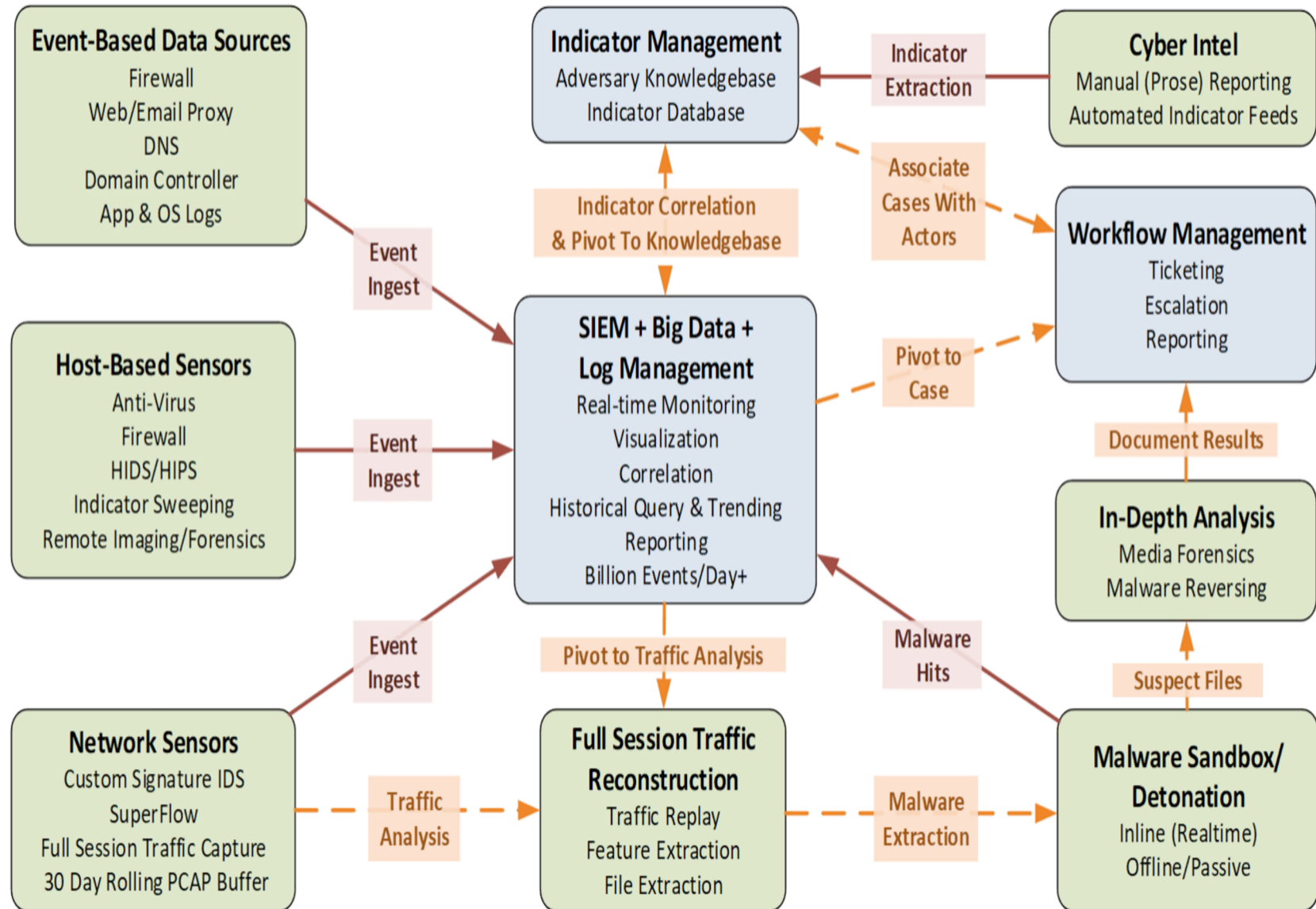
mysql.log | MySQL

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

Some Companies/Insts. Using Zeek

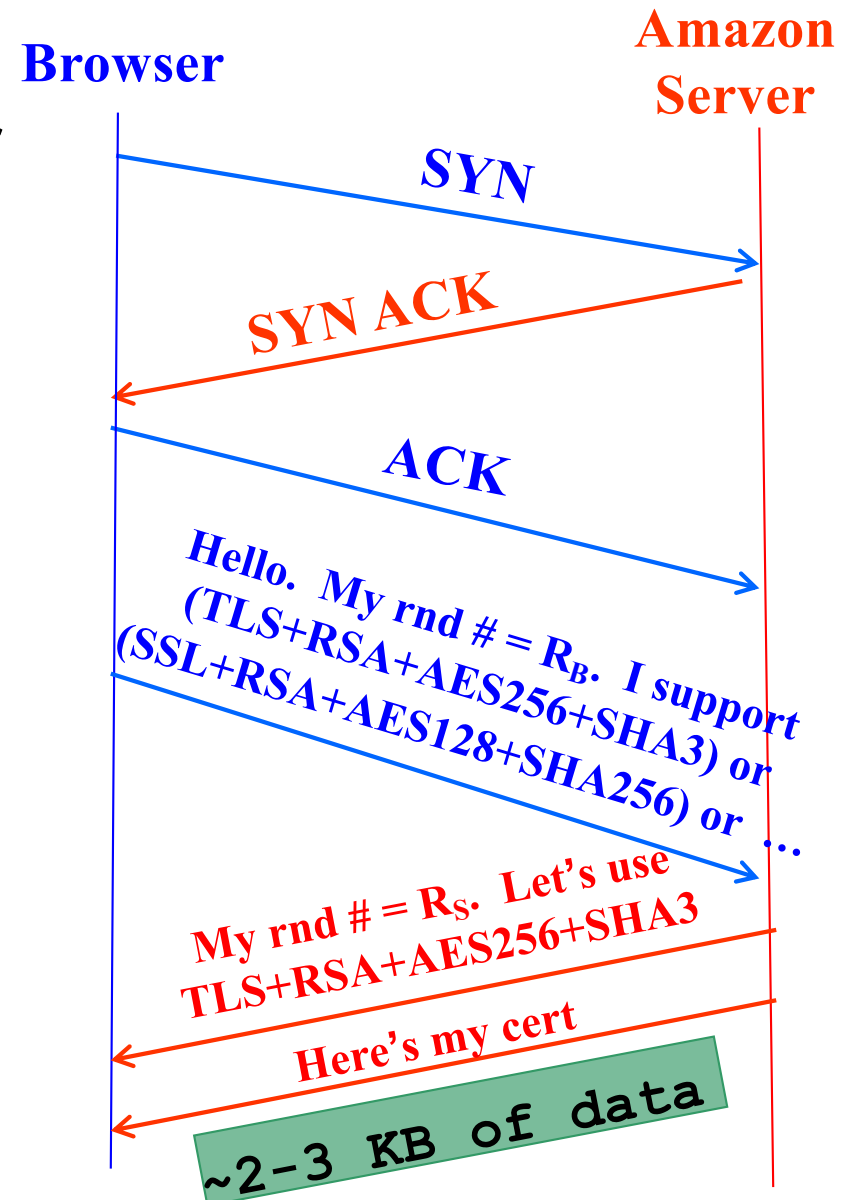


SOC operations overview (Microsoft)



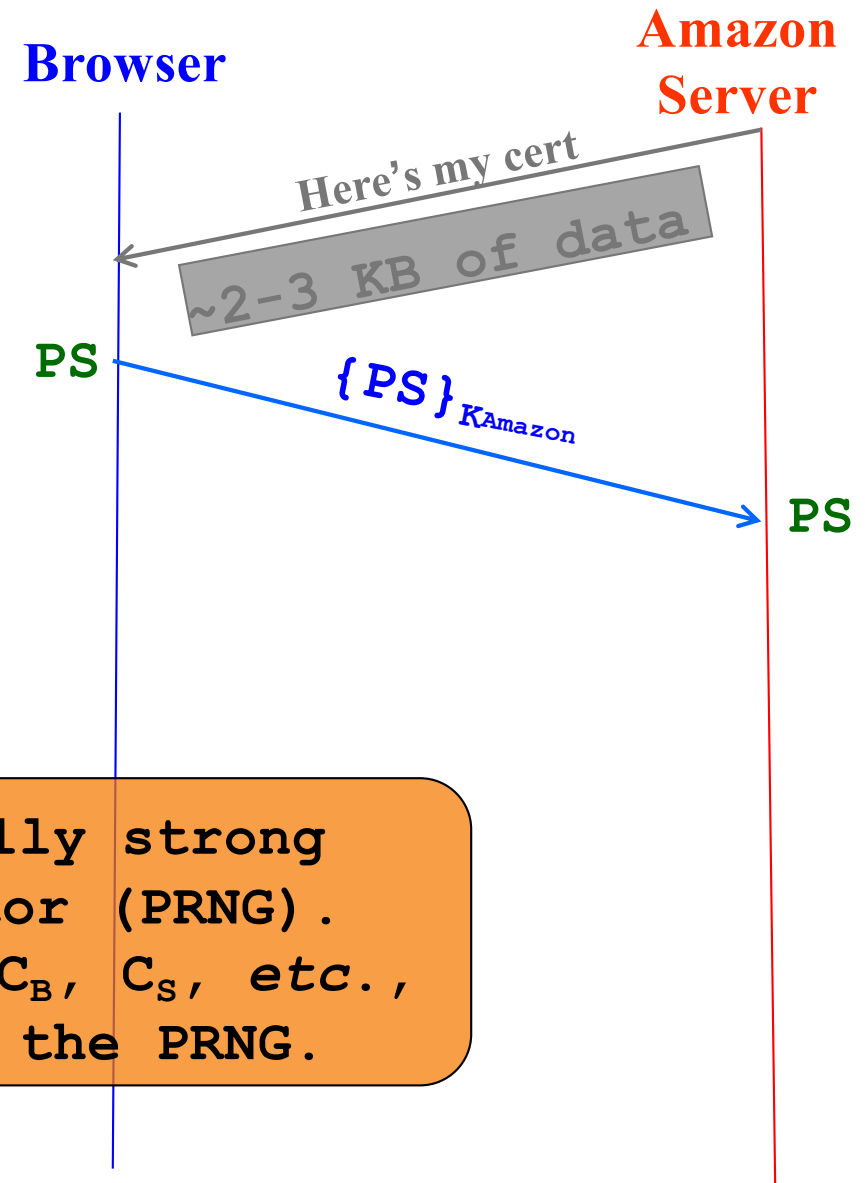
HTTPS Connection (SSL / TLS)

- Browser (client) connects via TCP to Amazon's **HTTPS** server
- Client picks 224-bit random number R_B , sends over list of crypto protocols it supports
- Server picks 224-bit random number R_S , selects protocols to use for this session
- Server sends over its certificate
- (all of this is in the clear)
- ***Client now validates cert***




HTTPS Connection (SSL / TLS), con't

- For RSA, browser constructs long (368 bits) “Premaster Secret” **PS**
- Browser sends **PS** encrypted using Amazon’s public RSA key K_{Amazon}
- Using **PS**, R_B , and R_S , browser & server derive symm. cipher keys (C_B , C_S) & MAC integrity keys (I_B , I_S)
 - One pair to use in each direction



These seed a cryptographically strong pseudo-random number generator (PRNG). Then browser & server produce C_B , C_S , etc., by making repeated calls to the PRNG.

HTTPS Connection (SSL / TLS), con't

- For RSA, browser constructs long (368 bits) “Premaster Secret” **PS**
- Browser sends PS encrypted using Amazon’s public RSA key K_{Amazon}
- Using PS, R_B , and R_S , browser & server derive symm. *cipher keys* (C_B , C_S) & MAC *integrity keys* (I_B , I_S)
 - One pair to use in each direction
- Browser & server exchange MACs computed over entire dialog so far
- If good MAC, Browser displays 
- All subsequent communication encrypted w/ symmetric cipher (e.g., **AES256**) cipher keys, MACs
 - Messages also numbered to thwart **replay attacks**

