

---

# Tranalyzer2

sslDecode



SSL/TLS and OpenVPN

---



Tranalyzer Development Team

Contents

<b>1</b>	<b>sslDecode</b>	<b>1</b>
1.1	Description . . . . .	1
1.2	Dependencies . . . . .	1
1.3	Configuration Flags . . . . .	1
1.4	Flow File Output . . . . .	3
1.5	Plugin Report Output . . . . .	15

## 1 sslDecode

### 1.1 Description

This plugin analyzes SSL/TLS and OpenVPN traffic.

### 1.2 Dependencies

If `SSL_ANALYZE_CERT` is activated, then `libssl` is required.

SSL_ANALYZE_CERT=1		
<b>Ubuntu:</b>	<code>sudo apt-get install</code>	<code>libssl-dev</code>
<b>Arch:</b>	<code>sudo pacman -S</code>	<code>openssl</code>
<b>openSUSE:</b>	<code>sudo zypper install</code>	<code>libopenssl-devel</code>
<b>Red Hat/Fedora<sup>1</sup>:</b>	<code>sudo dnf install</code>	<code>openssl-devel</code>
<b>macOS<sup>2</sup>:</b>	<code>brew install</code>	<code>openssl@1.1</code>

### 1.3 Configuration Flags

The following flags can be used to control the output of the plugin:

Name	Default	Description
<code>SSL_ANALYZE_OVPN</code>	0	Analyze OpenVPN (Experimental)
<code>SSL_ANALYZE_QUIC</code>	0	Analyze TLS 1.3 client/server hello in decrypted QUIC initial packets Requires the <code>quicDecode</code> plugin with <code>QUIC_DECODE_TLS = 1</code>
<code>SSL_REC_VER</code>	1	Output the list and number of record versions
<code>SSL_MAX_REC_VER</code>	3	Maximum number of record versions to store
<code>SSL_HAND_VER</code>	1	Output the list and number of handshake versions
<code>SSL_MAX_HAND_VER</code>	2	Maximum number of handshake versions to store
<code>SSL_EXT_LIST</code>	1	Output the list and number of extensions
<code>SSL_MAX_EXT</code>	20	Maximum number of extensions to store
<code>SSL_SUPP_VER</code>	1	Output the list and number of supported versions
<code>SSL_MAX_SUPP_VER</code>	4	Maximum number of supported versions to store
<code>SSL_SIG_ALG</code>	1	Output the list and number of signature hash algorithms
<code>SSL_MAX_SIG_ALG</code>	15	Maximum number of signature hash algorithms to store
<code>SSL_EC</code>	1	Output the list and number of elliptic curves
<code>SSL_MAX_EC</code>	6	Maximum number of elliptic curves to store
<code>SSL_EC_FORMATS</code>	1	Output the list and number of elliptic curve formats

---

<sup>1</sup>If the `dnf` command could not be found, try with `yum` instead

<sup>2</sup>Brew is a packet manager for macOS that can be found here: <https://brew.sh>

Name	Default	Description
SSL_MAX_EC_FORMATS	6	Maximum number of elliptic curve formats to store
SSL_ALPN_LIST	1	Output the list and number of protocols (ALPN)
SSL_ALPS_LIST	1	Output the list and number of protocols (ALPS)
SSL_NPN_LIST	1	Output the list and number of protocols (NPN)
SSL_MAX_PROTO	6	Maximum number of protocols (ALPN/ALPS/NPN) to store
SSL_PROTO_LEN	16	Maximum number of characters per protocol (ALPN)
SSL_CIPHER_LIST	1	Output the list and number of supported ciphers
SSL_MAX_CIPHER	3	Maximum number of ciphers to store
SSL_ANALYZE_CERT	1	Analyze certificates
If SSL_ANALYZE_CERT > 0, the following flags are available:		
SSL_CERT_SERIAL	1	Print the certificate serial number
SSL_CERT_FINGERPRINT	1	0: no certificate fingerprint, 1: SHA1, 2: MD5
SSL_CERT_VALIDITY	1	Print certificates validity (Valid from/to, lifetime)
SSL_CERT_SIG_ALG	1	Print the certificate signature algorithm
SSL_CERT_PUBKEY_ALG	1	Print the certificate public key algorithm
SSL_CERT_ALG_NAME_LONG	0	Use short (0) or long (1) names for algorithms
SSL_CERT_PUBKEY_TS	1	Print certificates public key type and size
SSL_CERT_SUBJECT	2	0: no info about cert subject, 1: whole subject as one string, 2: selected fields (see below)
SSL_CERT_ISSUER	2	0: no info about cert issuer, 1: whole issuer as one string, 2: selected fields (see below)
SSL_CERT_COMMON_NAME	1	Print the common name of the issuer/subject
SSL_CERT_ORGANIZATION	1	Print the organization name of the issuer/subject
SSL_CERT_ORG_UNIT	1	Print the organizational unit of the issuer/subject
SSL_CERT_LOCALITY	1	Print the locality name of the issuer/subject
SSL_CERT_STATE	1	Print the state/province name of the issuer/subject
SSL_CERT_COUNTRY	1	Print the country of the issuer/subject (iso3166)
SSL_RM_CERTDIR	1	Remove SSL_CERT_PATH before starting
SSL_SAVE_CERT	0	Save certificates
SSL_CERT_NAME_FINDEX	0	Prepend the flowIndex to the certificate name
SSL_DETECT_TOR	0	Detect likely Tor connections
SSL_BLIST	0	Flag blacklisted certificates
SSL_BLIST_LEN	41	Max length for blacklist descriptions

Name	Default	Description
SSL_JA4	1	Output JA4/JA4S fingerprints
SSL_JA4_O	0	Output JA4/JA4S_o fingerprints (original order)
SSL_JA4_R	0	Output JA4/JA4S_r fingerprints (raw)
SSL_JA4_RO	0	Output JA4/JA4S_ro fingerprints (raw, original order)
SSL_JA4_STR_LEN	254	Max length for uncompressed JA4 signatures (JA4/JA4S_r, JA4/JA4S_ro)
SSL_JA4_DLEN	64	Max length for JA4/JA4S descriptions (sslJA4Desc)
SSL_JA3	1	Output JA3 fingerprints (hash and description)
SSL_JA3_STR	0	Also output JA3 fingerprints before hashing
SSL_JA3_DLEN	64	Max length for JA3 descriptions
SSL_JA3_STR_LEN	1024	Max length for uncompressed JA3 signatures (ja3_str)

If `SSL_SAVE_CERT==1` then, certificates are saved under `SSL_CERT_PATH` (default: `"/tmp/TranCerts/"`) with the extension `SSL_CERT_EXT` (default: `".pem"`) and the SHA1 or MD5 fingerprint as filename.

### 1.3.1 Environment Variable Configuration Flags

The following configuration flags can also be configured with environment variables (`ENVCNTRL>0`):

- `SSL_RM_CERTDIR`
- `SSL_CERT_PATH`
- `SSL_CERT_EXT`

## 1.4 Flow File Output

The `sslDecode` plugin outputs the following columns:

Column	Type	Description	Flags
<code>sslStat</code>	H32	Status	
<code>sslProto</code>	H32	Protocol	
<code>ovpnType</code>	H16	OpenVPN message types	<code>SSL_ANALYZE_OVPN=1</code>
<code>ovpnSessionID</code>	U64	OpenVPN session ID	<code>SSL_ANALYZE_OVPN=1</code>
<code>sslFlags</code>	H8	SSL flags	
<code>sslVersion</code>	H16	SSL/TLS Version	
<code>sslNumRecVer</code>	U16	Number of record versions	<code>SSL_REC_VER=1</code>
<code>sslRecVer</code>	R(H16)	List of record versions	<code>SSL_REC_VER=1</code>
<code>sslNumHandVer</code>	U16	Number of handshake versions	<code>SSL_HAND_VER=1</code>
<code>sslHandVer</code>	R(H16)	List of handshake versions	<code>SSL_HAND_VER=1</code>
<code>sslVuln</code>	H8	Vulnerabilities	
<code>sslAlert</code>	H64	Alert type	
<code>sslCipher</code>	H16	Preferred (Client)/Negotiated (Server) cipher	
<code>sslNumExt</code>	U16	Number of extensions	<code>SSL_EXT_LIST=1</code>

Column	Type	Description	Flags
<a href="#">sslExtList</a>	R(H16)	List of extensions	SSL_EXT_LIST=1
<a href="#">sslNumSuppVer</a>	U16	Number of supported versions	SSL_SUPP_VER=1
<a href="#">sslSuppVer</a>	R(H16)	List of supported versions (client) (Server: negotiated version)	SSL_SUPP_VER=1
<a href="#">sslNumSigAlg</a>	U16	Number of signature hash algorithms	SSL_SIG_ALG=1
<a href="#">sslSigAlg</a>	R(H16)	List of signature hash algorithms	SSL_SIG_ALG=1
<a href="#">sslNumECPt</a>	U16	Number of elliptic curve points	SSL_EC=1
<a href="#">sslECPt</a>	R(H16)	List of elliptic curve points	SSL_EC=1
<a href="#">sslNumECFormats</a>	U8	Number of EC point formats	SSL_EC_FORMATS=1
<a href="#">sslECFormats</a>	R(H8)	List of EC point formats	SSL_EC_FORMATS=1
<a href="#">sslNumALPN</a>	U16	Number of protocols (ALPN)	SSL_ALPN_LIST=1
<a href="#">sslALPNList</a>	R(S)	List of protocols (ALPN)	SSL_ALPN_LIST=1
<a href="#">sslNumALPS</a>	U16	Number of protocols (ALPS)	SSL_ALPS_LIST=1
<a href="#">sslALPSList</a>	R(S)	List of protocols (ALPS)	SSL_ALPS_LIST=1
<a href="#">sslNumNPN</a>	U16	Number of protocols (NPN)	SSL_NPN_LIST=1
<a href="#">sslNPNList</a>	R(S)	List of protocols (NPN)	SSL_NPN_LIST=1
<a href="#">sslNumCipher</a>	U16	Number of supported ciphers	SSL_CIPHER_LIST=1
<a href="#">sslCipherList</a>	R(H16)	List of supported ciphers	SSL_CIPHER_LIST=1
<a href="#">sslNumCC_</a>	U16_	Number of change_cipher records,	
<a href="#">A_</a>	U16_	Number of alert records,	
<a href="#">H_</a>	U16_	Number of handshake records,	
<a href="#">AD_</a>	U64_	Number of application data records,	
<a href="#">HB</a>	U64	Number of heartbeat records	
<a href="#">sslSessIdLen</a>	U8	Session ID length	
<a href="#">sslGMTTime</a>	R(TS)	GMT Unix Time	
<a href="#">sslServerName</a>	R(S)	server name	

If `SSL_ANALYZE_CERT == 1`, the following columns are output:

<a href="#">sslCertVersion</a>	R(U8)	Certificate version	SSL_CERT_FINGERPRINT=1
<a href="#">sslCertSerial</a>	R(SC)	Certificate serial number	SSL_CERT_SERIAL=1
<a href="#">sslCertShalFP</a>	R(SC)	Certificate SHA1 fingerprint	SSL_CERT_FINGERPRINT=1
<a href="#">sslCertMd5FP</a>	R(SC)	Certificate MD5 fingerprint	SSL_CERT_FINGERPRINT=2
<a href="#">sslCNotValidBefore_</a>	TS_	Certificate validity: not valid before,	SSL_CERT_VALIDITY=1
<a href="#">after_</a>	R( TS_	not valid after,	
<a href="#">lifetime</a>	U64)	lifetime	
<a href="#">sslCSigAlg</a>	RS	Certificate signature algorithm	SSL_CERT_SIG_ALG=1
<a href="#">sslCKeyAlg</a>	RS	Certificate public key algorithm	SSL_CERT_PUBKEY_ALG=1
<a href="#">sslCPKeyType_</a>	SC_	Certificate public key type,	SSL_CERT_PUBKEY_TS=1
<a href="#">Size</a>	U16	Certificate public key size (bits)	

If `SSL_CERT_SUBJECT > 0`, the following columns are output:

<a href="#">sslCSubject</a>	R(S)	Certificate subject	SSL_CERT_SUBJECT=1
<a href="#">sslCSubjectCommonName</a>	R(S)	Certificate subject common name	SSL_CERT_SUBJECT=2

Column	Type	Description	Flags
sslCSubjectOrgName	R(S)	Certificate subject organization name	SSL_CERT_SUBJECT=2
sslCSubjectOrgUnit	R(S)	Certificate subject organizational unit name	SSL_CERT_SUBJECT=2
sslCSubjectLocality	R(S)	Certificate subject locality name	SSL_CERT_SUBJECT=2
sslCSubjectState	R(S)	Certificate subject state or province name	SSL_CERT_SUBJECT=2
sslCSubjectCountry	R(S)	Certificate subject country name	SSL_CERT_SUBJECT=2

If `SSL_CERT_ISSUER > 0`, the following columns are output:

sslCIssuer	R(S)	Certificate issuer	SSL_CERT_ISSUER=1
sslCIssuerCommonName	R(S)	Certificate issuer common name	SSL_CERT_ISSUER=2
sslCIssuerOrgName	R(S)	Certificate issuer organization name	SSL_CERT_ISSUER=2
sslCIssuerOrgUnit	R(S)	Certificate issuer organizational unit name	SSL_CERT_ISSUER=2
sslCIssuerLocality	R(S)	Certificate issuer locality name	SSL_CERT_ISSUER=2
sslCIssuerState	R(S)	Certificate issuer state or province name	SSL_CERT_ISSUER=2
sslCIssuerCountry	R(S)	Certificate issuer country name	SSL_CERT_ISSUER=2
sslBlistCat	R(S)	Blacklisted certificate category	SSL_BLIST=1&& (SSL_SAVE_CERT=1    SSL_CERT_FINGERPRINT=1)
sslJA3Hash	R(SC)	JA3 fingerprint	SSL_JA3=1
sslJA3Desc	R(S)	JA3 description	SSL_JA3=1
sslJA3Str	R(S)	JA3 string	SSL_JA3=1&& SSL_JA3_STR=1
sslJA4	R(SC)	JA4/JA4S fingerprint	SSL_JA4=1
sslJA4Desc	R(S)	JA4/JA4S description	SSL_JA4=1
sslJA4O	R(SC)	JA4/JA4S_o fingerprint (original order)	SSL_JA4_O=1
sslJA4R	R(SC)	JA4/JA4S_r fingerprint (raw)	SSL_JA4_R=1
sslJA4RO	R(SC)	JA4/JA4S_ro fingerprint (raw, original order)	SSL_JA4_RO=1
sslTorFlow	U8	Tor flow	SSL_DETECT_TOR=1

If `SSL_CERT_SUBJECT=2` or `SSL_CERT_ISSUER=2`, then the columns displayed are controlled by the following self-explanatory flags:

- `SSL_CERT_COMMON_NAME`,
- `SSL_CERT_ORGANIZATION`,
- `SSL_CERT_ORG_UNIT`,
- `SSL_CERT_LOCALITY`,
- `SSL_CERT_STATE`,
- `SSL_CERT_COUNTRY`.

### 1.4.1 sslStat

The hex based status variable `sslStat` is defined as follows:

sslStat	Description
0x0000 0001	Message had mismatched version
0x0000 0002	Record was too long (max 16384)
0x0000 0004	Record was malformed, eg, invalid value
0x0000 0008	Certificate had expired
0x0000 0010	Connection was closed due to fatal alert
0x0000 0020	Connection was renegotiated (existed before)
0x0000 0040	Peer not allowed to send heartbeat requests
0x0000 0080	Cipher list truncated. . . increase <code>SSL_MAX_CIPHER</code>
0x0000 0100	Extension list truncated. . . increase <code>SSL_MAX_EXT</code>
0x0000 0200	Protocol (ALPN/NPN/ALPS) list truncated. . . increase <code>SSL_MAX_PROTO</code>
0x0000 0400	Protocol (ALPN/NPN/ALPS) name truncated. . . increase <code>SSL_PROTO_LEN</code>
0x0000 0800	EC or EC formats list truncated. . . increase <code>SSL_MAX_EC</code> or <code>SSL_MAX_EC_FORMATS</code>
0x0000 1000	Certificate is blacklisted
0x0000 2000	Insecure or weak cipher detected (Null, DES, RC4 (RFC7465), ADH, 40/56 bits)
0x0000 4000	Weak protocol detected (SSL 2.0, SSL 3.0)
0x0000 8000	Weak key detected
0x0001 0000	Signature hash algorithms list truncated. . . increase <code>SSL_MAX_SIG_ALG</code>
0x0002 0000	Supported versions list truncated. . . increase <code>SSL_MAX_SUPP_VER</code>
0x0004 0000	Packet snapped, decoding failed
0x0008 0000	Failed to compute JA3 fingerprint. . . increase <code>SSL_JA3_STR_LEN</code>
0x0010 0000	Failed to compute JA4/JA4S fingerprint
0x0020 0000	JA4/JA4S_a successfully computed
0x0040 0000	JA4/JA4S_b successfully computed
0x0080 0000	JA4/JA4S_c successfully computed
0x0100 0000	Insecure cipher (should NEVER be used)
0x0200 0000	Weak cipher (should not be used)
0x0400 0000	Secure cipher
0x0800 0000	Perfect Forward Secrecy (PFS) ciphers
0x1000 0000	JA4/JA4S fingerprint truncated. . . increase <code>SSL_JA4_STR_LEN</code>
0x2000 0000	Record versions list truncated. . . increase <code>SSL_MAX_REC_VER</code>
0x4000 0000	Handshake versions list truncated. . . increase <code>SSL_MAX_HAND_VER</code>
0x8000 0000	—

### 1.4.2 sslProto

The hex based protocol variable `sslProto` is defined as follows:



sslProto	Description
0x0000 0001	HTTP/0.9, HTTP/1.0 or HTTP/1.1 (ALPN starts with http)
0x0000 0002	HTTP/2 (h2 or h2c)
0x0000 0004	HTTP/3 (h3 or HTTP/0.9/1.1 over QUIC (hq))
0x0000 0008	SPDY/1, SPDY/2 or SPDY/3 (ALPN starts with spdy)
0x0000 0010	IMAP
0x0000 0020	POP3
0x0000 0040	FTP
0x0000 0080	XMPP jabber
0x0000 0100	STUN/TURN
0x0000 0200	Apple Push Notification Service (APNS))
0x0000 0400	WebRTC Media and Data or Confidential WebRTC Media and Data
0x0000 0800	Constrained Application Protocol ( <a href="#">CoAP</a> )
0x0000 1000	<a href="#">ManageSieve</a>
0x0000 2000	RTP or RTCP <sup>3</sup>
0x0000 4000	OpenVPN <sup>4</sup>
0x0000 8000	OASIS Message Queuing Telemetry Transport (MQTT)
0x0001 0000	acme-tls/1
0x0002 0000	DICOM
0x0004 0000	NNTP (reading) or NNTP (transit)
0x0008 0000	SIP
0x0010 0000	Tabular Data Stream Protocol (TDS)
0x0020 0000	DNS over Dedicated QUIC Connections (DoQ)
0x0040 0000	DNS-over-TLS (DoT)
0x0080 0000	IRC
0x0100 0000	SMB
0x0200 0000	SUNRPC
0x0400 0000	Network Time Security Key Establishment
0x0800 0000	—
0x1000 0000	—
0x2000 0000	—
0x4000 0000	GREASE value
0x8000 0000	Unknown protocol (ALPN matched none of the above)

### 1.4.3 ovpnType

The ovpnType column is to be interpreted as follows:

<sup>3</sup>Guessed by the presence of the use-srtp hello extension

<sup>4</sup>Guessed by being able to decode the protocol

ovpnType	Description
2 <sup>1</sup> (=0x0002)	P_CONTROL_HARD_RESET_CLIENT_V1
2 <sup>2</sup> (=0x0004)	P_CONTROL_HARD_RESET_SERVER_V1
2 <sup>3</sup> (=0x0008)	P_CONTROL_SOFT_RESET_V1
2 <sup>4</sup> (=0x0010)	P_CONTROL_V1
2 <sup>5</sup> (=0x0020)	P_ACK_V1
2 <sup>6</sup> (=0x0040)	P_DATA_V1
2 <sup>7</sup> (=0x0080)	P_CONTROL_HARD_RESET_CLIENT_V2
2 <sup>8</sup> (=0x0100)	P_CONTROL_HARD_RESET_SERVER_V2
2 <sup>9</sup> (=0x0200)	P_DATA_V2

#### 1.4.4 sslFlags

The sslFlags is defined as follows:

sslFlags	Description
0x01	request is SSLv2
0x02	SSLv3 version on 'request' layer different than on 'record' layer
0x04	gmt_unix_time is small (less than 1 year since epoch, probably seconds since boot)
0x08	gmt_unix_time is more than 5 years in the future (probably random)
0x10	random data (28 bytes) is not random
0x20	compression (deflate) is enabled

#### 1.4.5 sslVersion, sslRecVer, sslHandVer and sslSuppVer

The hex based version variables sslVersion, sslRecVer, sslHandVer and sslSuppVer are defined as follows:

sslVersion	Description
0x0200	SSL 2.0
0x0300	SSL 3.0
0x0301	TLS 1.0
0x0302	TLS 1.1
0x0303	TLS 1.2
0x0304	TLS 1.3
0x0a0a	GREASE value
0x1a1a	GREASE value
0x2a2a	GREASE value
0x3a3a	GREASE value
0x4a4a	GREASE value
0x5a5a	GREASE value
0x6a6a	GREASE value
0x7a7a	GREASE value
0x7f0e	TLS 1.3 (draft 14)
0x7f0f	TLS 1.3 (draft 15)
0x7f10	TLS 1.3 (draft 16)
0x7f11	TLS 1.3 (draft 17)
0x7f12	TLS 1.3 (draft 18)
0x7f13	TLS 1.3 (draft 19)
0x7f14	TLS 1.3 (draft 20)

sslVersion	Description
0x7f15	TLS 1.3 (draft 21)
0x7f16	TLS 1.3 (draft 22)
0x7f17	TLS 1.3 (draft 23)
0x7f18	TLS 1.3 (draft 24)
0x7f19	TLS 1.3 (draft 25)
0x7f1a	TLS 1.3 (draft 26)
0x7f1b	TLS 1.3 (draft 27)
0x7f1c	TLS 1.3 (draft 28)
0x8a8a	GREASE value
0x9a9a	GREASE value
0xaaaa	GREASE value
0xbaba	GREASE value
0xcaca	GREASE value
0xdada	GREASE value
0xeaea	GREASE value
0xfafa	GREASE value
0xfb17	TLS 1.3 (Facebook draft 23)
0xfb1a	TLS 1.3 (Facebook draft 26)
0xfefc	DTLS 1.3
0xfefd	DTLS 1.2
0xfeff	DTLS 1.0

#### 1.4.6 sslVuln

The hex based vulnerability variable `sslVuln` is defined as follows:

sslVuln	Description
0x01	vulnerable to BEAST
0x02	vulnerable to BREACH
0x04	vulnerable to CRIME
0x08	vulnerable to FREAK
0x10	vulnerable to POODLE
0x20	HEARTBLEED attack attempted
0x40	HEARTBLEED attack successful (Not implemented)
0x80	—

### 1.4.7 sslAlert

The hex based alert variable `sslAlert` is defined as follows (**red** is fatal):

sslAlert	Description	sslAlert	Description
0x00000000 00000001	close notify	0x00000001 00000000	unknown PSK identity (fatal)
0x00000000 00000002	unexpected message (fatal)	0x00000002 00000000	no application protocol (fatal)
0x00000000 00000004	bad record MAC (fatal)	0x00000004 00000000	—
0x00000000 00000008	decryption failed	0x00000008 00000000	—
0x00000000 00000010	record overflow	0x00000010 00000000	—
0x00000000 00000020	decompression failed (fatal)	0x00000020 00000000	—
0x00000000 00000040	handshake failed (fatal)	0x00000040 00000000	—
0x00000000 00000080	no certificate	0x00000080 00000000	—
0x00000000 00000100	bad certificate	0x00000100 00000000	—
0x00000000 00000200	unsupported certificate	0x00000200 00000000	—
0x00000000 00000400	certificate revoked	0x00000400 00000000	—
0x00000000 00000800	certificate expired	0x00000800 00000000	—
0x00000000 00001000	certificate unknown	0x00001000 00000000	—
0x00000000 00002000	illegal parameter (fatal)	0x00002000 00000000	—
0x00000000 00004000	unknown CA (fatal)	0x00004000 00000000	—
0x00000000 00008000	access denied (fatal)	0x00008000 00000000	—
0x00000000 00010000	decode error (fatal)	0x00010000 00000000	—
0x00000000 00020000	decrypt error	0x00020000 00000000	—
0x00000000 00040000	export restriction (fatal)	0x00040000 00000000	—
0x00000000 00080000	protocol version (fatal)	0x00080000 00000000	—
0x00000000 00100000	insufficient security (fatal)	0x00100000 00000000	—
0x00000000 00200000	internal error (fatal)	0x00200000 00000000	—
0x00000000 00400000	user canceled	0x00400000 00000000	—
0x00000000 00800000	no renegotiation	0x00800000 00000000	—
0x00000000 01000000	unsupported extension	0x01000000 00000000	—
0x00000000 02000000	inappropriate fallback (fatal)	0x02000000 00000000	—
0x00000000 04000000	certificate unobtainable	0x04000000 00000000	—
0x00000000 08000000	unrecognized name	0x08000000 00000000	—
0x00000000 10000000	bad certificate status response	0x10000000 00000000	—
0x00000000 20000000	bad certificate hash value	0x20000000 00000000	—
0x00000000 40000000	unknown PSK identity (fatal)	0x40000000 00000000	—
0x00000000 80000000	no application protocol (fatal)	0x80000000 00000000	Fatal

### 1.4.8 sslCipher

The `sslCipher` variable represents the preferred cipher for the client and the negotiated cipher for the server. The corresponding name can be found in the `src/sslCipher.h` file. All values following the `0x[0-9a-f]a[0-9a-f]a` pattern are GREASE values.

### 1.4.9 sslNumCC\_A\_H\_AD\_HB

The number of message variable `sslNumCC_A_H_AD_HB` decomposed as follows:

<code>sslNumCC_A_H_AD_HB</code>	Description
<code>sslNumCC</code>	number of change cipher records
<code>sslNumA</code>	number of alerts records
<code>sslNumH</code>	number of handshake records
<code>sslNumAD</code>	number of application data records
<code>sslNumHB</code>	number of heartbeat records

### 1.4.10 sslExtList

The list of extensions is to be interpreted as follows:

sslExt	Description
0x0000	Server name
0x0001	Max fragment length
0x0002	Client certificate URL
0x0003	Trusted CA keys
0x0004	Truncated HMAC
0x0005	Status request
0x0006	User mapping
0x0007	Client authz
0x0008	Server authz
0x0009	Cert type
0x000a	Supported groups (elliptic curves)
0x000b	EC point formats
0x000c	SRP
0x000d	Signature hash algorithms
0x000e	Use SRTP
0x000f	Heartbeat
0x0010	ALPN
0x0011	Status request v2
0x0012	Signed certificate timestamp
0x0013	Client certificate type
0x0014	Server certificate type
0x0015	Padding
0x0016	Encrypt then MAC
0x0017	Extended master secret
0x0018	Token binding
0x0019	Cached info
0x001a	TLS LTS
0x001b	Compress certificate
0x001c	Record size limit
0x001d	Pwd protect
0x001e	Pwd clear
0x001f	Password salt
0x0020	Ticket pinning
0x0021	TLS cert with extern PSK
0x0022	Delegated credential
0x0023	Session ticket
0x0024	TLMSP
0x0025	TLMSP proxying
0x0026	TLMSP delegate
0x0027	Supported EKT ciphers
0x0028	Extended random
0x0029	Pre-Shared Key (PSK)
0x002a	Early data

sslExt	Description
0x002b	Supported versions
0x002c	Cookie
0x002d	PSK key exchange modes
0x002f	Certificate authorities
0x0030	OID filters
0x0031	Post handshake auth
0x0032	Signature algorithms cert
0x0033	Key Share
0x0034	Transparency info
0x0035	Connection ID (deprecated)
0x0036	Connection ID
0x0037	External ID hash
0x0038	External session ID
0x0039	QUIC transport parameters
0x0040	Ticket request
0x0041	DNSSEC chain
0x0042	Sequence number encryption algo. (DTLS)
0x0043	Return Routability Check (RRC) for DTLS
0x0a0a	GREASE value
0x1a1a	GREASE value
0x2a2a	GREASE value
0x3374	NPN
0x3377	Origin bound cert
0x337c	Encrypted client cert
0x3a3a	GREASE value
0x4469	Application settings (ALPS)
0x4a4a	GREASE value
0x5a5a	GREASE value
0x6a6a	GREASE value
0x754f	Channel ID old
0x7550	Channel ID
0x7a7a	GREASE value
0x8a8a	GREASE value
0x9a9a	GREASE value
0xaaaa	GREASE value
0xbaba	GREASE value
0xcaca	GREASE value
0xdada	GREASE value
0xeaea	GREASE value
0xfafa	GREASE value
0xfd00	Encrypted Client Hello outer extensions
0xfe0d	Encrypted Client Hello
0xff01	Renegotiation info

#### 1.4.11 sslSigAlg

The list of signature hash algorithms is to be interpreted as follows:

sslSigAlg	Description
0x0201	rsa_pkcs1_sha1
0x0203	ecdsa_sha1
0x0401	rsa_pkcs1_sha256
0x0403	ecdsa_secp256r1_sha256
0x0420	rsa_pkcs1_sha256_legacy
0x0501	rsa_pkcs1_sha384
0x0503	ecdsa_secp384r1_sha384
0x0520	rsa_pkcs1_sha384_legacy
0x0601	rsa_pkcs1_sha512
0x0620	rsa_pkcs1_sha512_legacy
0x0603	ecdsa_secp521r1_sha512
0x0708	sm2sig_sm3
0x0709	gostr34102012_256a
0x070a	gostr34102012_256b
0x070b	gostr34102012_256c
0x070c	gostr34102012_256d
0x070d	gostr34102012_512a
0x070e	gostr34102012_512b
0x070f	gostr34102012_512c
0x0804	rsa_pss_rsae_sha256
0x0805	rsa_pss_rsae_sha384
0x0806	rsa_pss_rsae_sha512
0x0807	ed25519
0x0808	ed448
0x0809	rsa_pss_pss_sha256
0x080a	rsa_pss_pss_sha384
0x080b	rsa_pss_pss_sha512
0x081a	ecdsa_brainpoolP256r1tls13_sha256
0x081b	ecdsa_brainpoolP384r1tls13_sha384
0x081c	ecdsa_brainpoolP512r1tls13_sha512
0x0a0a	GREASE value
0x1a1a	GREASE value
0x2a2a	GREASE value
0x3a3a	GREASE value
0x4a4a	GREASE value
0x5a5a	GREASE value
0x6a6a	GREASE value
0x7a7a	GREASE value
0x8a8a	GREASE value
0x9a9a	GREASE value
0xaaaa	GREASE value
0xbaba	GREASE value
0xcaca	GREASE value

sslSigAlg	Description
0xdada	GREASE value
0xeaea	GREASE value
0xfafa	GREASE value
0xfea0	dilithium2
0xfea1	p256_dilithium2
0xfea2	rsa3072_dilithium2
0xfea3	dilithium3
0xfea4	p384_dilithium3
0xfea5	dilithium5
0xfea6	p521_dilithium5
0xfea7	dilithium2_aes
0xfea8	p256_dilithium2_aes
0xfea9	rsa3072_dilithium2_aes
0xfeaa	dilithium3_aes
0xfeab	p384_dilithium3_aes
0xfeac	dilithium5_aes
0xfead	p521_dilithium5_aes
0xfe0b	falcon512
0xfe0c	p256_falcon512
0xfe0d	rsa3072_falcon512
0xfe0e	falcon1024
0xfe0f	p521_falcon1024
0xfe96	picnic1full
0xfe97	p256_picnic1full
0xfe98	rsa3072_picnic1full
0xfe1b	picnic311
0xfe1c	p256_picnic311
0xfe1d	rsa3072_picnic311
0xfe27	rainbowIclassic
0xfe28	p256_rainbowIclassic
0xfe29	rsa3072_rainbowIclassic
0xfe3c	rainbowVclassic
0xfe3d	p521_rainbowVclassic
0xfe42	sphincsharaka128frobust
0xfe43	p256_sphincsharaka128frobust
0xfe44	rsa3072_sphincsharaka128frobust
0xfe5e	sphincssha256128frobust
0xfe5f	p256_sphincssha256128frobust
0xfe60	rsa3072_sphincssha256128frobust
0xfe7a	sphincsshake256128frobust
0xfe7b	p256_sphincsshake256128frobust
0xfe7c	rsa3072_sphincsshake256128frobust



#### 1.4.12 sslCNotValidBefore\_after\_lifetime

The `sslCNotValidBefore_after_lifetime` indicates the validity period of the certificate, i.e., not valid before / after, and the number of seconds between those two dates.

### 1.5 Plugin Report Output

The following information is reported:

- Aggregated `sslStat`
- Number of OpenVPN flows (`SSL_ANALYZE_OVPN=1`)
- Number of Tor flows (`SSL_DETECT_TOR=1`)
- Number of SSL 2.0, 3.0
- Number of TLS 1.0, 1.1, 1.2 and 1.3
- Number of DTLS 1.0 (OpenSSL pre 0.9.8f), 1.0 and 1.2 flows.
- Aggregated `sslProto`
- Number of certificates saved (`SSL_SAVE_CERT=1`)
- Number of blacklisted certificates (`SSL_BLIST=1`)
- Number of JA3 signatures matched (`SSL_JA3=1`)
- Number of JA4 signatures matched (`SSL_JA4=1`)
- Number of JA4S signatures matched (`SSL_JA4=1`)