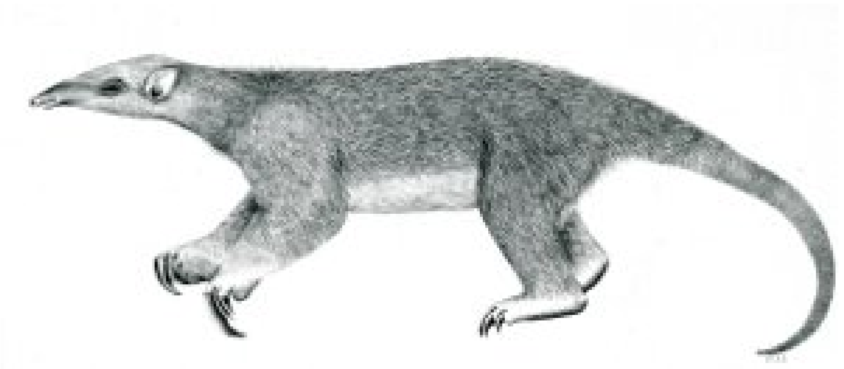
Tranalyzer2

sslDecode

SSL/TLS and OpenVPN



Tranalyzer Development Team

Contents

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.

Arch: `sudo pacman -S openssl`

Ubuntu/Kali: `sudo apt-get install libssl-dev`

OpenSUSE: `sudo zypper install libopenssl-devel`

Red Hat/Fedora: `sudo yum install openssl-devel`

Mac OSX: `brew install openssl`

1.3 Configuration Flags

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

Name	Default	Description
SSL_ANALYZE_OVPN	0	Analyze OpenVPN (Experimental)
SSL_EXT_LIST	1	Output the list and number of extensions
SSL_MAX_EXT	8	Maximum number of extensions to store
SSL_EC	1	Output the list and number of elliptic curves
SSL_MAX_EC	6	Maximum number of elliptic curves to store
SSL_EC_FORMATS	1	Output the list and number of elliptic curve formats
SSL_MAX_EC_FORMATS	6	Maximum number of elliptic curve formats to store
SSL_PROTO_LIST	1	Output the list and number of protocols
SSL_MAX_PROTO	6	Maximum number of protocols to store
SSL_PROTO_LEN	16	Maximum number of characters per protocol
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:

Name	Default	Description
SSL_CERT_SERIAL	1	Print the certificate serial number
SSL_CERT_FINGPRINT	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	Whether to 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_BLIST	0	Flag blacklisted certificates
SSL_JA3	1	Output JA3 fingerprints (hash and description)
SSL_JA3_STR	0	Also output JA3 fingerprints before hashing

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.

1.4 Flow File Output

The sslDecode plugin outputs the following columns:

Column	Type	Description	Flags
sslStat	H16	Status	
sslProto	H16	Protocol	
ovpnType	H16	OpenVPN message types	SSL_ANALYZE_OVPN=1
ovpnSessionID	U64	OpenVPN session ID	SSL_ANALYZE_OVPN=1

Column	Type	Description	Flags
sslFlags	H8	SSL flags	
sslVersion	H16	SSL/TLS Version	
sslVuln	H8	Vulnerabilities	
sslAlert	H32	Alert type	
sslCipher	H16	Preferred (Client)/Negotiated (Server) cipher	
sslNumExt	U16	Number of extensions	SSL_EXT_LIST=1
sslExtList	RH16	List of extensions	SSL_EXT_LIST=1
sslNumECPt	U16	Number of elliptic curve points	SSL_EC=1
sslECPt	RH16	List of elliptic curve points	SSL_EC=1
sslNumECFormats	U8	Number of EC point formats	SSL_EC_FORMATS=1
sslECFormats	RH8	List of EC point formats	SSL_EC_FORMATS=1
sslNumProto	U16	Number of protocols	SSL_PROTO_LIST=1
sslProtoList	RS	List of protocols	SSL_PROTO_LIST=1
sslNumCipher	U16	Number of supported ciphers	SSL_CIPHER_LIST=1
sslCipherList	RH16	List of supported ciphers	SSL_CIPHER_LIST=1
sslNumCC_	U16_	Number of change_cipher records,	
A_	U16_	Number of alert records,	
H_	U16_	Number of handshake records,	
AD_	U64_	Number of application data records,	
HB	U64	Number of heartbeat records	
sslSessIdLen	U8	Session ID length	
sslGMTTime	RTS	GMT Unix Time	
sslServerName	RS	server name	

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

sslCertVersion	RU8	Certificate version	SSL_CERT_FINGPRINT=1
sslCertSerial	RSC	Certificate serial number	SSL_CERT_SERIAL=1
sslCertSha1FP	RSC	Certificate SHA1 fingerprint	SSL_CERT_FINGPRINT=1
sslCertMd5FP	RSC	Certificate MD5 fingerprint	SSL_CERT_FINGPRINT=2
sslCNotValidBefore_	TS_	Certificate validity: not valid before,	SSL_CERT_VALIDITY=1
after_	TS_	not valid after,	
lifetime	U64	lifetime	
sslCSigAlg	RS	Certificate signature algorithm	SSL_CERT_SIG_ALG=1
sslCKeyAlg	RS	Certificate public key algorithm	SSL_CERT_PUBKEY_ALG=1
sslCPKeyType_	SC_	Certificate public key type,	SSL_CERT_PUBKEY_TS=1
Size	U16	Certificate public key size (bits)	

If SSL_CERT_SUBJECT > 0, the following columns are output:

sslCSubject	RS	Certificate subject	SSL_CERT_SUBJECT=1
sslCSubjectCommonName	RS	Certificate subject common name	SSL_CERT_SUBJECT=2
sslCSubjectOrgName	RS	Certificate subject organization name	SSL_CERT_SUBJECT=2
sslCSubjectOrgUnit	RS	Certificate subject organizational unit name	SSL_CERT_SUBJECT=2
sslCSubjectLocality	RS	Certificate subject locality name	SSL_CERT_SUBJECT=2

Column	Type	Description	Flags
sslCSubjectState	RS	Certificate subject state or province name	SSL_CERT_SUBJECT=2
sslCSubjectCountry	RS	Certificate subject country name	SSL_CERT_SUBJECT=2
If SSL_CERT_ISSUER > 0, the following columns are output:			
sslCIssuer	RS	Certificate issuer	SSL_CERT_ISSUER=1
sslCIssuerCommonName	RS	Certificate issuer common name	SSL_CERT_ISSUER=2
sslCIssuerOrgName	RS	Certificate issuer organization name	SSL_CERT_ISSUER=2
sslCIssuerOrgUnit	RS	Certificate issuer organizational unit name	SSL_CERT_ISSUER=2
sslCIssuerLocality	RS	Certificate issuer locality name	SSL_CERT_ISSUER=2
sslCIssuerState	RS	Certificate issuer state or province name	SSL_CERT_ISSUER=2
sslCIssuerCountry	RS	Certificate issuer country name	SSL_CERT_ISSUER=2
sslBlistCat	RS	Blacklisted certificate category	SSL_BLIST=1
sslJA3Hash	RSC	JA3 fingerprint	SSL_JA3=1
sslJA3Desc	RS	JA3 description	SSL_JA3=1

sslJA3Str	RS	JA3 string	SSL_JA3=1&&
			SSL_JA3_STR=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
`0x0001`	message had mismatched version
`0x0002`	record was too long (max 16384)
`0x0004`	record was malformed, eg, invalid value
`0x0008`	certificate had expired
`0x0010`	connection was closed due to fatal alert
`0x0020`	connection was renegotiated (existed before)
`0x0040`	peer not allowed to send heartbeat requests
`0x0080`	cipher list truncated. . . increase `SSL_MAX_CIPHER`
`0x0100`	extension list truncated. . . increase `SSL_MAX_EXT`
`0x0200`	protocol list truncated. . . increase `SSL_MAX_PROTO`
`0x0400`	protocol name truncated. . . increase `SSL_PROTO_LEN`
`0x0800`	EC or EC formats list truncated... increase `SSL_MAX_EC` or `SSL_MAX_EC_FORMATS`
`0x1000`	Certificate is blacklisted
`0x2000`	weak cipher detected (Null, DES, RC4 (RFC7465), ADH, 40/56 bits)
`0x4000`	weak protocol detected (SSL 2.0, SSL 3.0)
`0x8000`	weak key detected

1.4.2 sslProto

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

sslProto	Description
`0x0001`	HTTP/0.9, HTTP/1.0, HTTP/1.1 (ALPN starts with `http`)
`0x0002`	HTTP/2 (`h2`, `h2c`)
`0x0004`	HTTP/3 (`h3`)
`0x0008`	SPDY
`0x0010`	IMAP
`0x0020`	POP3
`0x0040`	FTP
`0x0080`	XMPP jabber
`0x0100`	STUN/TURN
`0x0200`	APNS (Apple Push Notification Service)
`0x0400`	WebRTC Media and Data
`0x0800`	CoAP
`0x1000`	ManageSieve
`0x2000`	RTP or RTCP[1]
`0x4000`	OpenVPN[2]

[1] Guessed by the presence of the `use-srtp` hello extension

[2] Guessed by being able to decode the protocol

sslProto	Description
0x8000	Unknown protocol (ALPN matched none of the above)

1.4.3 ovpnType

The ovpnType column is to be interpreted as follows:

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

The hex based version variable sslVersion is defined as follows:

sslVersion	Description
0x0300	SSL 3.0
0x0301	TLS 1.0
0x0302	TLS 1.1
0x0303	TLS 1.2
0x0304	TLS 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)

1.4.7 sslAlert

The hex based alert variable `sslAlert` is defined as follows:

sslAlert	Description
0x00000001	close notify
0x00000002	unexpected message
0x00000004	bad record MAC
0x00000008	decryption failed
0x00000010	record overflow
0x00000020	decompression failed
0x00000040	handshake failed
0x00000080	no certificate
0x00000100	bad certificate
0x00000200	unsupported certificate
0x00000400	certificate revoked
0x00000800	certificate expired
0x00001000	certificate unknown
0x00002000	illegal parameter
0x00004000	unknown CA
0x00008000	access denied

sslAlert	Description
0x00010000	decode error
0x00020000	decrypt error
0x00040000	export restriction
0x00080000	protocol version
0x00100000	insufficient security
0x00200000	internal error
0x00400000	user canceled
0x00800000	no renegotiation
0x01000000	unsupported extension
0x02000000	inappropriate fallback
0x04000000	certificate unobtainable
0x08000000	unrecognized name
0x10000000	bad certificate status response
0x20000000	bad certificate hash value
0x40000000	unknown PSK identity
0x80000000	no application protocol

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.

1.4.9 sslNumCC_A_H_AD_HB

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

sslNumCC_A_H_AD_HB	Description
sslNumCC	number of change cipher records

sslNumCC_A_H_AD_HB	Description
`sslNumA`	number of alerts records
`sslNumH`	number of handshake records
`sslNumAD`	number of application data records
`sslNumHB`	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 algorithms
`0x000e`	Use SRTP
`0x000f`	Heartbeat

sslExt	Description
`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 type
`0x0023`	Session ticket
`0x0028`	Extended random
`0x3374`	NPN
`0x3377`	Origin bound cert
`0x337c`	Encrypted client cert
`0x754f`	Channel ID old
`0x7550`	Channel ID
`0xff01`	renegotiation_info

1.4.11 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 number of OpenVPN, Tor, SSL 2.0, 3.0, TLS 1.0, 1.1, 1.2 and 1.3 and DTLS 1.0 (OpenSSL pre 0.9.8f), 1.0 and 1.2 flows is reported.

1.6 TODO

In order to analyze all certificates, we need to reassemble packets.