Fraunhofer
FKIE

FRAUNHOFER INSTITUTE FOR COMMUNICATION, INFORMATION PROCESSING AND ERGONOMICS FKIE

TLExport - GENERATING DECRYPTED TLS PCAPS

Daniel Baier*, Jannis F. Borg-Olivier, Lars Morkovsky

OVERVIEW

Network forensic tools often do not support full decryption of TLS traffic, especially when not all key material is provided. TLExport allows the generation of copies of traffic captures containing the decrypted TLS data, enabling forensic researchers to use tools and analyses that lack decryption capabilities.

TLExport highlights:

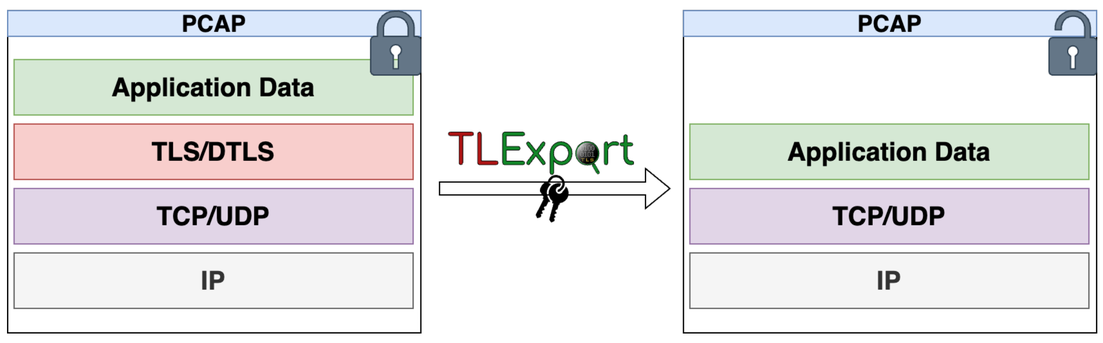
- Decryption of **TLS payload as pcap** for the given keys
- First tool enabling **TLS 1.3 application traffic decryption** utilizing only the **application traffic secrets**
- Open source and publicly available at **`https://github.com/fkie-cad/TLExport`**

MOTIVATION

More and more malware leverages TLS encryption to hide its communications and to exfiltrate data to its command server, effectively bypassing traditional detection platforms. This trend underscores the importance of accessing decrypted network traffic in digital forensics and cybercrime investigations. While there are methods allowing the extraction of TLS keys required for decryption, the range of tools available that can apply these keys for decryption is still limited. For example, projects such as Zeek and Scapy offer only rudimentary support for TLS decryption.

CONCEPT

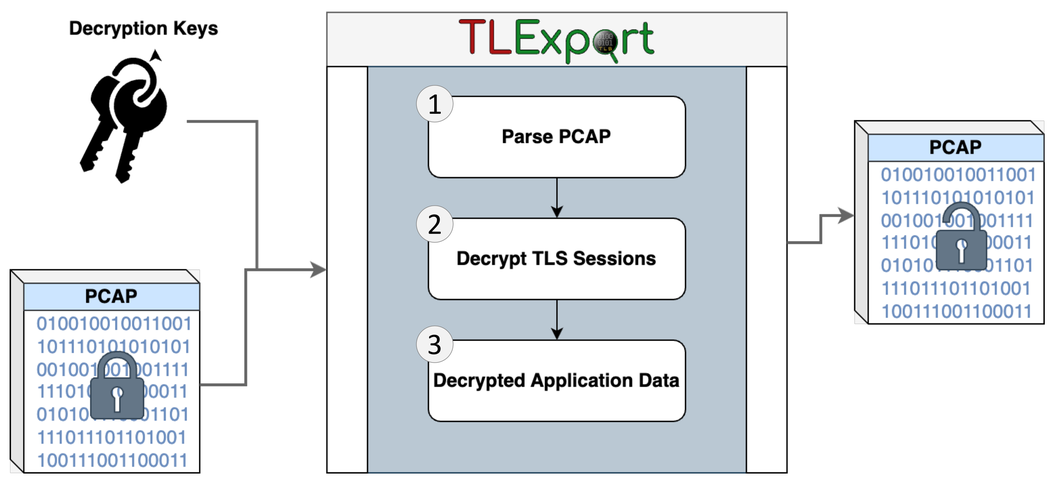
TLExport is a tool for decrypting TLS traffic and exporting it as unencrypted TCP traffic. The goal is to support network analysis tools, which have no or limited support for TLS decryption.



In the first step (1) , TLExport parses the pcap and extracts the data blocks and contents. Moreover, it organizes the packets into sessions using IP addresses and ports. Finally, TLExport identifies which sessions are TLS sessions and performs stream reassembly.

The decryption process (2) begins with extracting the required keys for a TLS connection from dynamic secret blocks (DSBs) or an SSLKEYLOG file, using information from the client hello message and the client random.

The server hello message further assists in determining the ciphersuite and TLS version in use. This data, combined with the secrets from the SSLKEYLOG or DSB, enables the derivation of client and server write keys and initialization vectors.



In the last step (3), TLExport replaces the encrypted stream with the decrypted one. The decrypted stream is still a valid TCP connection but without any TLS-specific artifacts. The timestamps of the new TCP connections are meticulously aligned with the timestamps from the original TLS packets. This way, the integrity of the following analyses is maintained. It is also possible to specify custom ports for decrypted traffic to filter decrypted streams more efficiently.

WORKING WITH TLExport

TLExport offers two operating modes for processing the decryption keys. One is to get they keys from the DSB and the other is processing the SSLKEYLOG-file. When the DSB of a pcapng contains the keys, the decryption is easy:

```
FKIE ~/DEF/research/TLExport ./tlexport.py -i in.pcapng -o out.pcapng
[*] using keys from DSB
[*] Checking for TLS Traffic on these ports: [443]
[*] Decrypting session: [192.168.0.149:51132-10.37.129.2:443]
…
```

The `-i` parameter indicates the pcap with the encrypted TLS streams, and the `-o` parameter is used to specify the new decrypted pcap. To provide the SSLKEYLOG file the `-s` parameter is used:

```
FKIE ~/DEF/research/TLExport ./tlexport.py -i in.pcapng -o out.pcapng -s tls_13_keylog.log
[*] using keys from SSLKEYLOG: tls_13_keylog.log
[*] Checking for TLS Traffic on these ports: [443]
[*] Decrypting session: [192.168.0.149:51132-10.37.129.12:443]
…
```

FUTURE WORK

- Support for **QUIC**
- Support for **DTLS**

Daniel Baier	I +(49) 228 50212-427	I daniel.baier@fkie.fraunhofer.de
Jannis F. Borg-Olivier	I +(49) 228 50212-621	I jannis.finn.borg-olivier@fkie.fraunhofer.de
Lars Morkovsky	I +(49) 228 50212-621	I lars.morkovsky@fkie.fraunhofer.de