FACULTY
OF INFORMATION
TECHNOLOGY

Identification of Mobile Traffic using TLS Fingerprinting
PDS/e project, academic year 2020/2021

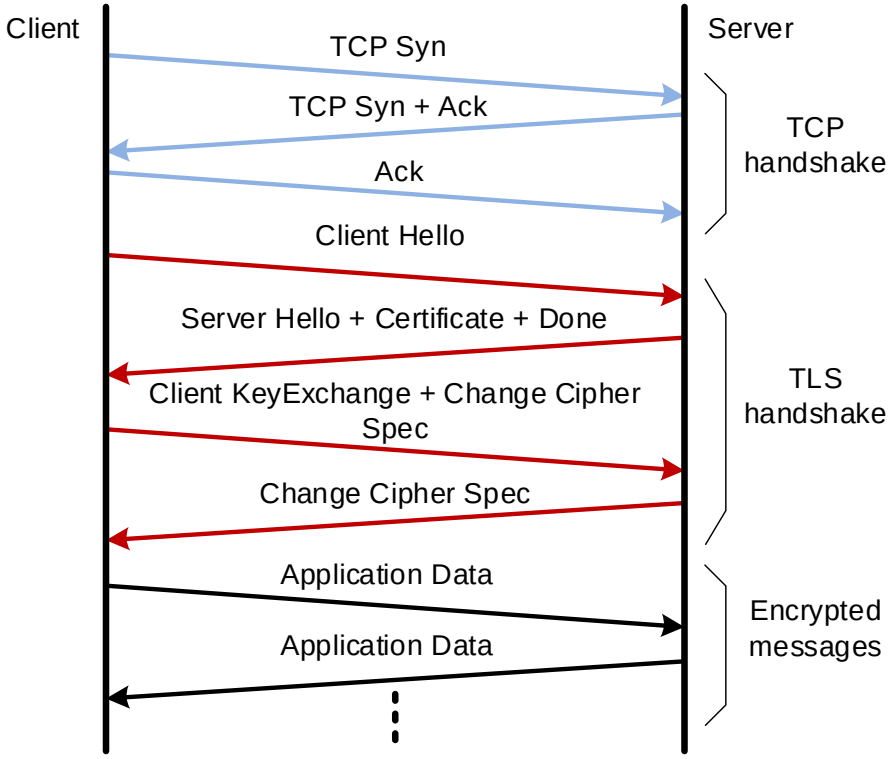
Dr. Petr Matoušek
Brno University of Technology, Faculty of Information Technology
Bozetechova 1/2, 612 66 Brno - Kralovo Pole
matousp@fit.vutbr.cz

PDS/e Project

- Goal: Evaluate how mobile encrypted traffic can be identified using TLS fingerprinting.
- The project includes several steps:
  1. Creation of a dataset with PCAP files for selected mobile apps.
  2. Data processing and extraction of relevant features that form a fingerprint.
  3. Computation of fingerprints using published methods.
  4. Evaluation of the identification: accuracy, precision, recall.
  5. Writing the project report.
  6. Submission of the project using FIT information system.
- Project deadline: 18th April 2021
- Maximum points: 21 (extra points for extensions, see the last slide)
- Online consultations available using Forum in IS FIT.
- Individual project – each student creates its own solution.
- Plagiarism prohibited – see copyrights and the publication policy.

TLS Fingerprinting

- During TLS handshake, TLS parameters are negotiated



Observation:

TLS handshake parameters depend on packages and methods that were used to build the application. These parameters are relatively stable and related to the application.

=> TLS handshake can be used for application fingerprinting [1,2].

[1] Husák, M., Čermák, M., Jirsík, T., Čeleda, P.: Https traffic analysis and client identification using passive SSL/TLS, fingerprinting. EURASIP Journal on Information Security (2016)

[2] Anderson, B., Paul, S., McGrew, D.: Deciphering malware's use of TLS (without decryption). Journal of Computer Virology and Hacking Techniques pp. 195-211 (2018)

How to Create a TLS Fingerprint using JA3 algorithm [3]

1. Extract selected fields from Client Hello: version, cipher suites, extensions, supported groups, EC formats.
2. Concatenate data in decimal format into one string.
3. Compute MD5 hash of the string => JA3 fingerprint of a client.

Version, Cipher Suites, Extensions, Supported Groups, EC format

0x00000303 - 49195,49196,52393,49199,49200,52392,158,159,49161,49162,49171,49172,51,57,156,157,47,53 - 65281,0,23,35,13,16,11,10 - 0x00000017,0x00000018,0x00000019 - 0

↓ Hex to Decimal Format

771, 49195-49196-52393-49199-49200-52392-158-159-49161-49162-49171-49172-51-57-156-157-47-53, 65281-0-23-35-13-16-11-10, 23-24-25, 0

↓ 32-bit MD5 hash

n8bvbvyZuTPF4tj89PaJVQ

- Similarly, we can compute JA3S fingerprint extracted from Server Hello data exchange [3] => it describes the server application.

[3] For original JA3 algorithm, see https://github.com/salesforce/ja3 or https://ja3er.com/

1) Creating a dataset

1. Choose 10 mobile apps for your experiments.
2. Download the apps to your mobile device or use Android Virtual Studio.
   - AVD provides an emulator where you can create a virtual device, see [4].
   - Using ADB CLI you can install the app, see [5].
3. Set the environment for capturing the network traffic.
   - For real mobile device, use WiFi connection to your laptop/desktop where you capture communication using Wireshark/tshark/tcpdump.
   - For emulated virtual device, use Wireshark/tshark/tcpdump on your local device where emulation is running.
4. Emulate traffic of the mobile app and capture the communication.
   - Start network capturing.
   - Launch your application, send/receive data for about 5 minutes.
   - For ADB, use CLI emulator monkeyrunner, see [6].
5. For each application, saved the capture data into separated PCAP file.

[4] https://developer.android.com/studio/run/managing-avds#about
[5] https://developer.android.com/studio/command-line/adb
[6] https://developer.android.com/studio/test/monkeyrunner

2) Data processing and extraction of TLS features

1. Select TLS communication from the PCAP file.
2. Extract relevant TLS attributes from TLS Client and Server Hellos.
3. Filter out TLS traffic related to the app only. Use SNI feature for filtering.

- You can use *tshark* command for TLS filtering, e.g.:

  tshark -r <PCAP File> -T fields -E separator=";" -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport -e tls.handshake.type -e tls.handshake.version -e tls.handshake.ciphersuite -e tls.handshake.extension.type -e tls.handshake.extensions_server_name -e tls.handshake.extensions_supported_group -e tls.handshake.extensions_ec_point_format -R "tls.handshake.type==1 or tls.handshake.type==2" -2

- See scripts available at https://github.com/matousp/ja3s-fingerprinting.
- The original JA3 tool is available at https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967.
- Another TLS fingerprinting tool is available at https://github.com/cisco/mercury.

3) Compute fingerprints for your mobile apps.

1. Select a method for TLS fingerprinting using features obtained from TLS headers.
2. Compute fingerprints of your mobile apps. Use your own tool or tools recommended on the previous slide.
3. Save fingerprints into the database (CSV file).

4) Evaluate your results

1. Create a new PCAP file with communication of some of your apps.
2. Extract TLS fingerprints from this PCAP file.
3. Compare the obtained fingerprints wrt. your fingerprint database.
4. Compute the accuracy, precision and recall.
   - For this, you need to compute number of true and false positives, and true of false negatives [7].

[7] See Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques* (3rd. ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

5) Compose the report with following parts (5-10 pages)

1. Problem description.
2. Problem analysis with description of the used TLS method.
3. Solution with description your mobile apps, dataset creation, extraction of TLS features and their processing.
4. Testing and evaluation: experiments with TLS fingerprint extraction and detection. Evaluation of your results.
5. Conclusion, discussion, contribution.

- For document use template for these available at https://www.fit.vut.cz/study/theses/bachelor-theses/.en.

6) Submit your results

1. Submit your report in PDF format (file *xlogin.pdf*).
2. Submit a zip file that includes following files (file *xlogin.zip*):
   - Readme.txt – your name, login, a list of files in ZIP archives.
   - PCAP files – each file with TLS learning data for one mobile app.
   - Source code of your scripts/tools you developer (optional).

Recommended references

- MATOUŠEK Petr, BURGETOVÁ Ivana, RYŠAVÝ Ondřej and VICTOR Malombe. On Reliability of JA3 Hashes for Fingerprinting Mobile Applications. In: *Digital Forensics and Cyber Crime. ICDF2C 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 351. Boston: Springer International Publishing, 2021, pp. 1-22. ISBN 978-3-030-68733-5, 10.1007/978-3-030-68734-2_1.
- Anderson, B., McGrew, D.: Accurate TLS Fingerprinting using Destination Context and Knowledge Bases, 2020, arXiv:2009.01939.
- Anderson, B., McGrew, D.: TLS Beyond the browser: combining end host and network data to understand application behavior. In: Proceedings of the Internet Measurement Conference, pp. 379–392 (2019), https://dl.acm.org/doi/10.1145/3355369.3355601.
- Anderson, B., Paul, S., McGrew, D.: Deciphering malware's use of TLS (without decryption). Journal of Computer Virology, Hacking Tech. (2018), https://arxiv.org/abs/1607.01639v1.

Concluding remarks

- The goal of the project is to create your own dataset, extract data from TLS communication, create TLS fingerprints using available tools and published methods, and evaluation TLS fingerprinting method using your own data.

- Partial solution is also accepted. This must be explicitly stated in Readme.txt.

- Any external tools, code, sources of information must be properly referenced, otherwise the work is considered as plagiarism.

- Extra points can be obtained for the following extensions:
  - Evaluation of your fingerprinting method with available external datasets.
  - Implementation of advanced accuracy using data context or additional features.
  - Application of advanced classification methods like naïve Bayes, etc. for fingerprint comparison.

Questions?