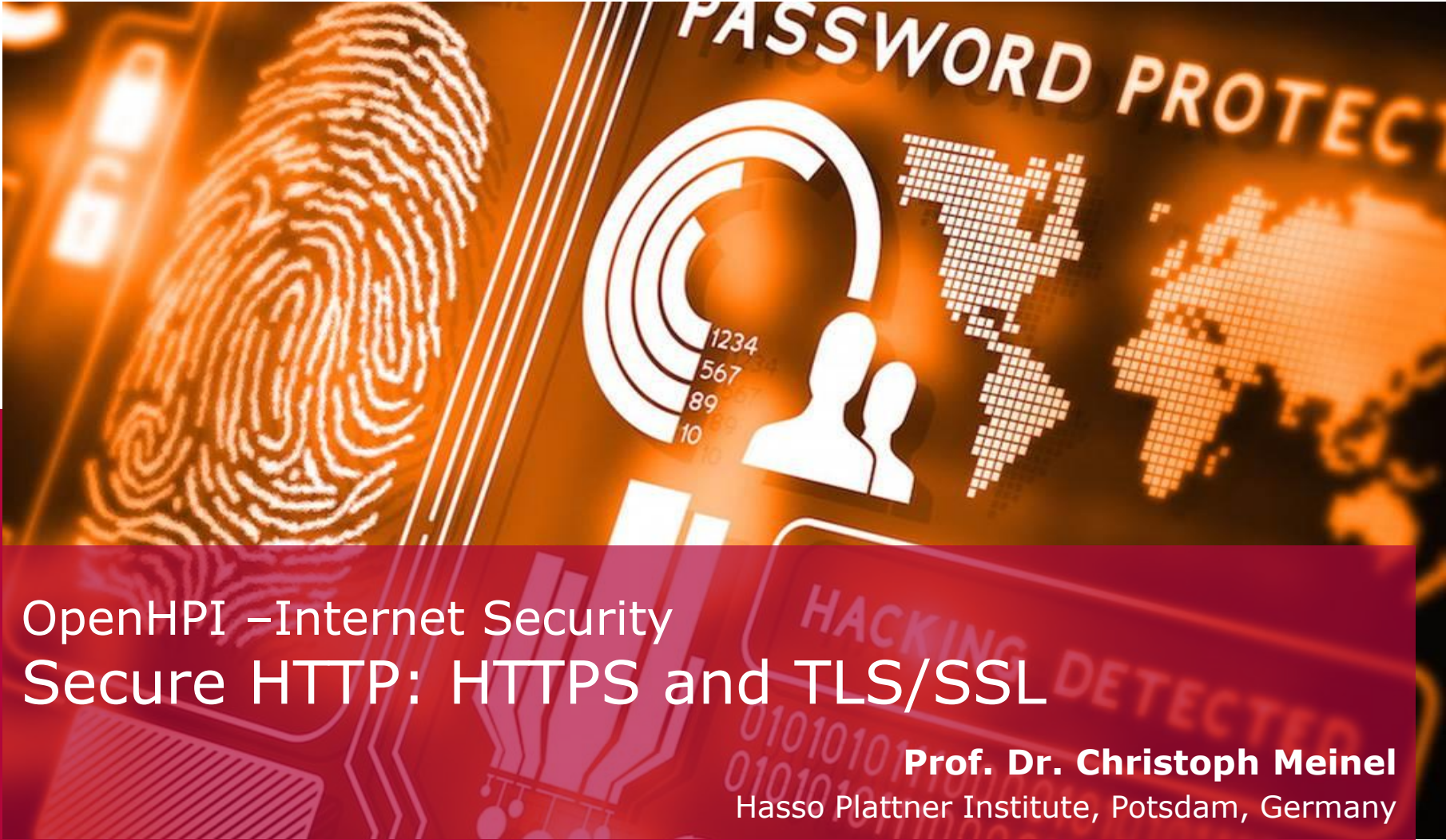




OpenHPI – Internet Security

Secure HTTP: HTTPS and TLS/SSL

Prof. Dr. Christoph Meinel
Hasso Plattner Institute, Potsdam, Germany

HTTPS – Hyper Text Transfer Protocol Secure

In the Internet protocol stack TCP/IP, HTTP is responsible for the connection of the WWW to the Internet.

- HTTP controls the communication between web browsers and the Internet service provider's web server
- HTTP does not, however, provide security
 - network packets are sent unencrypted and are readable by anyone who has access to the Internet
 - strong authentication is not possible

Therefore, you should always use the **secure HTTPS** instead of HTTP

HTTPS – Hyper Text Transfer Protocol Secure

HTTPS is the secure version of the HTTP protocol for the Web

- HTTPS makes use of TLS/SSL in the TCP/IP stack
 - TLS/SSL ensures the encryption of the communication connection and thus creates confidentiality
 - TLS/SSL provides secure mutual authentication by means of certificates
 - TLS/SSL uses efficient hybrid encryption
- HTTPS can be used to ensure secure communications with strong authentication for
 - Online banking
 - E-shopping
 - Government communications ...

HTTPS – Working Principle

Additional security performance of HTTPS is achieved via the automatically initiated TLS/SSL handshake

1. Browser (client) sends "Hello" message
2. Web server replies with "Hello"
 - in the "Hello" messages, parameters are defined such as, e.g., used encryption procedures, session ID, applied TLS/SSL version, ...
3. Web server sends its certificate to the client
4. Browser verifies certificate and extracts web server's public key
5. To schedule a session key, the browser sends a pre-master secret to the web server, encrypted with its own public key
6. The web server decrypts the pre-master secret and both compute the session key from it

The TLS/SSL protocols used by HTTPS offer a whole range of different cryptographic methods:

- 2-key encryption methods for exchanging the session key:
 - RSA
 - RSA with Diffie-Hellman key exchange
 - RSA with Diffie-Hellman based on elliptic curves
 - Digital Signature Algorithm (DSA) with Diffie-Hellman
- 1-key encryption methods for data traffic protection:
 - AES, Camellia, IDEA, 3DES, ...
- Hash methods to ensure data integrity:
 - HMAC (Hash based Message Authentication Code) based on SHA-1, SHA-256, MD5

HTTPS – Pros and Cons

Pros:

- Every Internet protocol of the application layer can use TLS/SSL and their security features
- TLS/SSL provides an additional and independent layer between the transport and the application layer in the TCP/IP protocol stack and remains transparent for applications

Cons: "Costs"

- Connection setup takes (more) time on the server side
- Encrypted data offers hardly any additional options of data compression to increase **throughput**
- TLS/SSL provides end-to-end encryption between two parties and is not suitable for multi-party services that need (partially) to read packets

HTTPS – Development of TLS/SSL

- SSL 1.0 – 1994 developed and released by Netscape
- SSL 2.0 – 1995 integrated into Netscape Navigator
 - same key for encryption and integrity tests
 - use of MD5 for integrity tests - now considered unsafe
- SSL 3.0
 - integrates certificate validation
 - use of SHA-1 algorithm possible
 - half of the master key depends on the MD5 hash function, considered unsafe because it is vulnerable to collisions
- TLS – Improvement on SSL 3.0
 - 1999 standardized by IETF as successor to SSL
 - integrity protection improved with additional key
 - ...

HTTPS – Implementation of TLS/SSL

- **OpenSSL**

- ☐ widely used version with public source code
- ☐ Heartbleed bug caused a sensation

- **LibreSSL**

- ☐ emerged from OpenSSL project and now further developed by the OpenBSD project

- **Microsoft Secure Channel Package**

- **OS X Secure Transport Package**

- **GnuTLS**

- ☐ freely available implementation under the GNU license

- ...

HTTPS – Attacks on TLS/SSL

BEAST – Browser Exploit Against TLS/SSL

- **Prerequisite:** Violation of the Same Origin Policy – injection of generated content into the website possible
- **Effect:** Reading cookies and stealing TLS sessions

Renegotiation

- Allows an attacker in a MITM position to combine their own data traffic with that of a client, whereby servers incorrectly identify the client as a sender

Heartbleed

- Allows the reading of the TLS/SSL application's memory, allowing disclosure of sensitive data (passwords, certificates, ...)
- Be sure to update old openssl versions!