CRYPTOMATHIC

HTTPS Encryption and Attacks on Authentication in Remote Banking Services - a Russian Perspective

by Ivan Piskunov (guest) on 17. August 2017

Authentication Banking

Read the Russian Version ***Шифрование HTTPS***

This article discusses the secure HTTPS Protocol intended for web-resources and its principles of operation as well as its strengths and weaknesses. It explains how attacks on HTTPS may lead to traffic being decrypted, particularly in systems for remote banking services and personal logins to web-resources.

Web resources, such as remote banking services, web portals for private offices, e-mail, instant messaging and VoIP-telephony require protocols for secure data exchange. This ensures the privacy of personal customer data and protection from tampering of the data exchanged between an Internet server and the user's electronic device. One of the most common and well-known application protocols for web-resources is HTTPS. The HTTPS protocol is essentially the implementation of the standard for Internet protocol HTTP using encryption. Asymmetric encryption such as RSA algorithms or Diffie-Hellman for authentication is used by systems, which utilize public and private keys to secure data.

The HTTPS Protocol

In the past, all data in HTTPS was encapsulated and sent over SSL and TLS cryptographic protocols. Today, only TSL Protocol is used because a critical vulnerability called Heartbleed (CVE-2014-0160) was found with SSL in 2014, which prevented it from being used in the future. The security of TSL sometimes

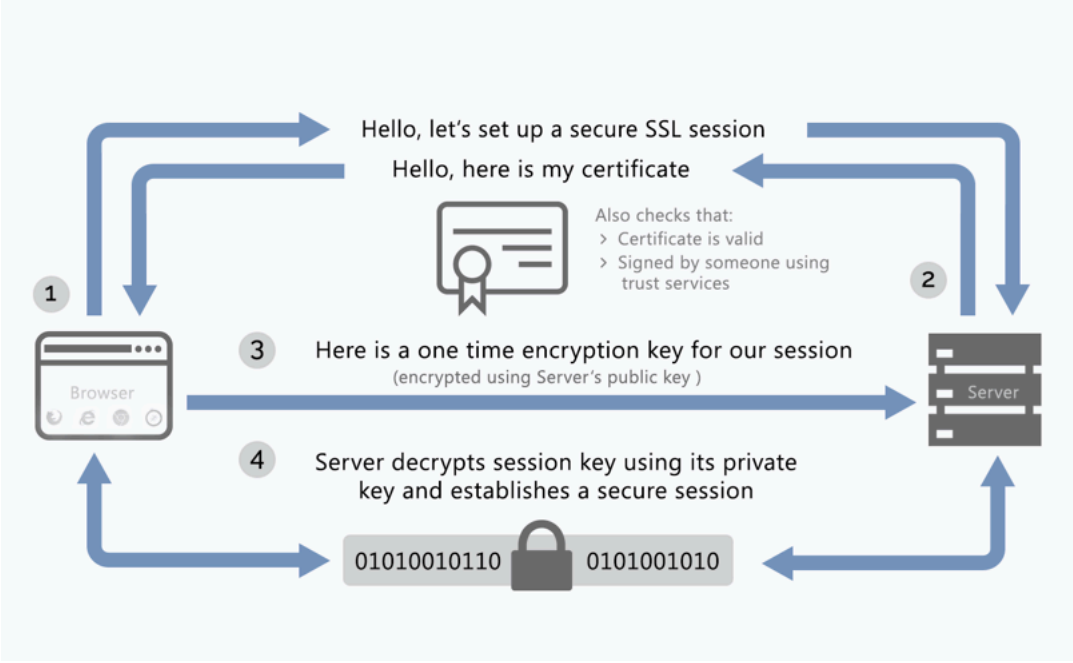
remains in question. For example, there have been vulnerabilities in the past relevant to open source clients like OpenSSL allowing the attack FREAK — TLS Downgrade. After those vulnerabilities were discovered, developers released patches to fix them.

Despite some vulnerabilities (which every system has), HTTPS remains a very commonly used security protocol on the Internet. It is used to authenticate web-resources requiring login to personal accounts and is often used with remote banking services. It primarily defends against attacks such as man-in-the-middle and active sniffing /packet analyzing of traffic using specialized hacking software. Relying on HTTPS is not infallible because hackers can still attempt to decrypt intercepted traffic that is transmitted over HTTPS. However, this is not a vulnerability of the cryptographic algorithms used or the technical implementation of this protocol. Often, this type of breach can be traced back to an incorrect configuration of the operating environment and building the chain of trust for the certificate and its public keys.

How Secure HTTPS Sessions Could Be Attacked

The process to establish a secure HTTPS session is basically a four-step process:

1. Through HTTPS, the user requests a secure session by sending a request to the server to set up a secure SSL session.
2. The web-server responds by sending its certificate to the user's browser where the certificate will be checked for validity and whether it is "signed" by "someone" that is trusted.
3. The browser responds by sending a one-time encryption key for the session that has been encrypted by using the public key supplied by the web-server.
4. The web-server then decrypts the session key with its private key, thus establishing a secure session between the two.



It is possible to create such a certificate without going to a single CA. Under Unix\Linux, tools such as *ssl-ca* or *gensslcert* (utility) can be used to create a certificate with two keys, which is called «self-signed».

Using this procedure, an attacker or legitimate network administrator could replace the original certificates from legitimate web resources with these self-signed certificates generated on the local proxy server, which is accessed by users the Internet. The administrator or the attacker:

- Generates certificates for each user
- Forces group policies in MS Active Directory
- Loads certificates into the browser of each user

After this, the user's computer will accept all certificates signed by any organizations that are trusted by the proxy server.

This scheme sniffs server traffic and is actually akin to a man-in-the-middle attack that is often used by hackers, on open Wi-Fi networks where the victim connects to an open and unsecured network in a public place, such as a café or subway. Alternatively, this tactic can be used by a network administrator to gain control of all network traffic by employees on a corporate network, similar to the same way a DPL-system would operate.

Russia's Response with Law of Yarovaya

In 2016, Russia approved a federal law called the "Law of Yarovaya." This law requires the transfer of encryption keys of all developers of all software and

hardware products in the Federal Security Service of Russia (FSB). The intent of this law is that the FSB will have access to all systems and all traffic arising on the Internet to:

- Ensure national security
- Prevent terrorist acts
- Investigate criminal acts

The need for such measures arose after terrorist attacks in Paris and explosions at the airport in Brussels when the confiscated iPhone of one of the terrorists was found to have encrypted content of communications between terrorists in preparing attacks.

Theoretically, by using the above-described method of attack on HTTPS, the FSB is able to access and decrypt all traffic that passes through encrypted channels. This would require large data centers and computing facilities. Many experts say that such action is a violation of the constitutional rights of citizens to privacy and secrecy of their communications. However, the FSB argued that this measure would be used only in a critical situation. This measure has already been legalized and put into effect in Kazakhstan during 2016.

References and Further Reading

- Selected articles on Authentication (2014-16), by Heather Walker, Luis Balbas, Guillaume Forget, Jan Kjaersgaard, Dawn M. Turner and more
- Selected articles on Electronic Signing and Digital Signatures (2014-16), by Ashiq JA, Guillaume Forget, Jan Kjaersgaard , Peter Landrock, Torben Pedersen, Dawn M. Turner, Tricia Wittig and more
- RFC 2818 HTTP Over TLS (2000), by E. Rescorla, Network Working Group
- Recommendations for the Security of Internet Payments (Final Version) (2013), by the European Central Bank
- Draft NIST Special Publication 800-63-3: Digital Authentication Guideline (2016), by the National Institute of Standards and Technology, USA.
- NIST Special Publication 800-63-2: Electronic Authentication Guideline (2013), by the National Institute of Standards and Technology, USA.
- Security Controls Related to Internat Banking Services (2016), Hong Kong Monetary Authority

Image: Matryoshka dolls, Moscow, courtesy of Neiljs, Flickr (CC BY 2.0)

Ivan Piskunov (guest)

SHARE

FACEBOOK

TWITTER

LINKEDIN

E MAIL

RECENT

MORE STORIES

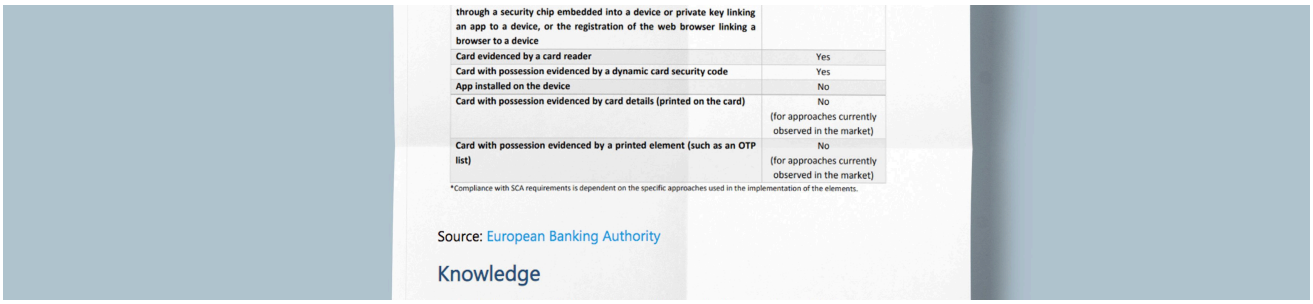
SUBSCRIBE

The biggest stories, delivered to your inbox.

Enter your email here

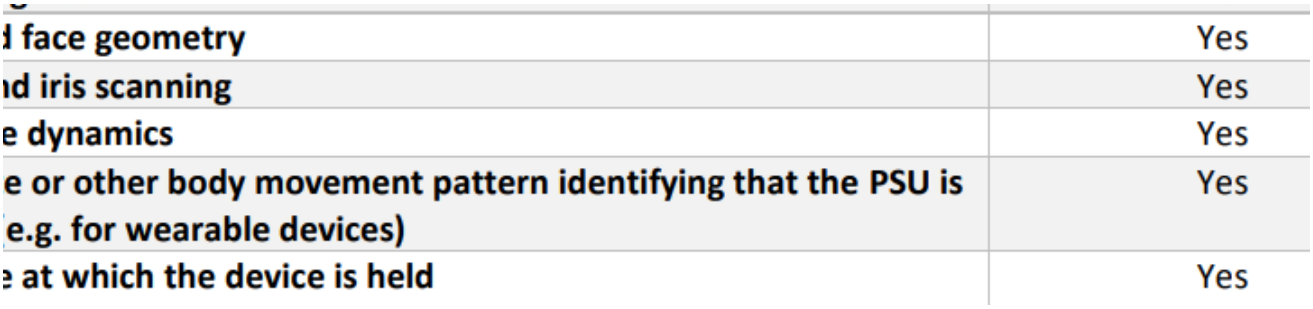
SUBMIT

RELATED ARTICLES



EBA's opinion on elements of Strong Customer Authentication under PSD2 – Part 2 – Possession and Knowledge

Gaurav Sharma (guest)



EBA's Opinion on elements of Strong Customer Authentication under PSD2 – Part I - Inherence

Gaurav Sharma (guest)

The PSD2 - Directive and Distributed Authentication

Peter Smirnoff & Ulrich Scholten (guests)

OTHER RELATED ARTICLES: # AUTHENTICATION # BANKING

Get in touch to better understand how our solutions secure ecommerce and billions of transactions worldwide.

CONTACT US

CRYPTOMATHIC

e: enquiry@cryptomathic.com

t: + 45 8676 2288

HEADQUARTERS

Cryptomathic A/S
Jaegergardsgade 118
DK-8000 Aarhus C
Denmark

Company

Home
About Us
Solutions
Contact Us
Offices

Resources

News
Events
Blog
Case studies
White papers