**Home**

Group Members :

1. Kan Lok Yiu Dorothy (57138602)
2. Wong Pui Ling (57142603)
3. WONG Zi Yin (57838772)
4. Man Nga Chung (56220231)

How HTTPS secures the Internet

To compare HTTPS with HTTP and demonstrate its benefits in security

Techniques :

* CSS
* JavaScript
* HTML

**Background**

What is HTTPS?

Hypertext Transfer Protocol Secure (HTTPS) is a secure version of HTTP used to transfer secure data between web browsers and web servers. This is especially important when users transmit sensitive information, such as log in to bank account. In a web browser, a website URL that does not use HTTPS will display "Not Secure"; conversely, a padlock padlock will indicate that the page is secure. Some web server / browser may even redirect your http request to https (and not giving you a chance to choose).

How it works?

HTTPS uses SSL (Secure Socket Layer) or TLS (Transport Layer Security) protocol. It uses encryption to protect communications by using asymmetric key encryption. This method uses a pair of different keys (Public key is publicly distributed; Private key is kept secret). Either key can be used for encryption and the other key for decryption. They can decode each other but cannot derive the other's key.

**Applications**

Secure Online Transactions

HTTPS encrypts the communication between the user's browser and the server to prevent attackers from intercepting and tampering with transaction data, ensuring the confidentiality and integrity of transactions.

E.g Online shopping, Payment gateways, Financial transactions

Secure Login & Authentication

When a user enters login information (such as a username and password), HTTPS ensures that the information is encrypted and protected during transmission. Prevent attackers from capturing information and gaining unauthorized access to user accounts.

E.g. Login to a Google account or Bank account

Privacy Protection

HTTPS encrypts data to protect users' personal information and browsing behavior from being collected, tracked, or monitored by unauthorized parties. Even if the user is using an unsecured or public network, attackers cannot easily monitor or snoop on the data exchanged between the user and the website.

**Main features/functions**

Authentication

The SSL (Secure Socket Layer) or TLS (Transport Layer Security) protocol performs asymmetric key encryption of data. To make encryption possible, the server must obtain a certificate from the Certificate Authority (CA), as the certificate stores the public key used in HTTPS and the private key is kept secret and secure.

Generally, browsers enable users to view the certificate, which also contains information about the owner, such as the domain name of the website and the issue date. Consequently, when users see the CA seal associated or trust indicators with a website, this authenticates the website has obtained a valid certificate from a trusted CA and is worth trusting for secure data transmission.

Secure Transfer of Data

HTTPS secures all incoming and outgoing traffic using Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols. These protocols utilize both symmetric and asymmetric encryption methods. Asymmetric encryption is used during the initial "handshake" between the browser and server to establish secure communication. Following this, symmetric encryption is used for data transmission to ensure efficient processing of large files. Both encryption methods safeguard transmitted data from theft or tampering, as even if intercepted by a third party, the lack of a public or private key prevents reading or modification, significantly enhancing the security of data transmission.

Data Integrity

HTTPS uses cryptographic techniques like hash functions to guarantee data integrity. Data is split up into small packets when transmitted. Each packet is assigned a message digest using the hash function. A hash function is a mathematical algorithm that generates a fixed-length character string from data. A slight alteration in the data will result in a completely different hash value. The message digest uniquely represents the data and is included in the HTTPS transmission with the actual data. The recipient can use the hash values to confirm the data's integrity after receiving it. By comparing the message digests, HTTPS offers a reliable method for identifying any unauthorized modifications or adjustments to the sent data.

**Discovery**

To compare HTTPS with HTTP and demonstrate its benefits in security.

Speed difference between using HTTP and HTTPS

In general, HTTP is slightly faster than HTTPS. The difference between the speed difference is due to the TLS handshake involve in HTTPS, which is also called the SSL handshake.

A TLS handshake involves multiple procedures. It includes the exchange of messages between the client and server, authentication of the server's SSL certificate by the client, encryption, decryption, and the generation of a session key. The number of roundtrips increases through this process, resulting in more load time for a page to show up and computation power. Until this process comes to an end, transmission of data starts.

In contrast, HTTP is simple. The connection is made directly using TCP, which does not involve encryption. Both client and server communicate in plaintext, with the client submitting HTTP requests and the server returning HTTP responses. No TLS handshake is required, which creates a relatively faster speed of HTTP over HTTPS.

However, the speed difference can be mitigated through enhanced TLS protocols and optimizations. Through a technology called TLS False Start, the TLS handshake could start to transmit data earlier before the handshake is completed. In addition, an updated version of TLS 1.3 has lowered the requirement of roundtrip by one.

Data packet

An experiment is done to track differences on data packets between transfering data through http and https.

1. Install software to track packet sniffer Wireshark (https://www.wireshark.org/) is used in this experiment.
2. Find a website that can be used by both HTTP and HTTPS Wikidot (http://www.wikidot.com/) is chosen.
3. Use https://temp-mail.org/en/email-generator to generate a temporary email address and create an account for testing in wikidot.
4. Use command prompt to find out the ip address of www.wikidot.com. The ip address is used to filter the necessary packets in Wireshark.
5. Open Wireshark to start the tracking.
6. Sign in to Wikidot using HTTP.
7. View the tracked data packet. (Image 1)

It is found that the login information and cookie are tracked. The login information is not transferred safely, it hurt our privacy. Hacker can use the collected username and password to login to my account. They may also use them for illegal usage.

1. Sign in to Wikidot using HTTPS.
2. View the tracked data packet.

Transport Layer Security (TLS) protocol is used. A handshake is done between client side and server side. (Image 2)

All the data transferred are encrypted using hashing algorithm. Even if the data packets are sniffed, they cannot be understandable because they are some nonsensical characters. (Image 3)

Content variation (sometimes) between HTTP and HTTPS

Certain elements, such as images and cascading style sheets, may not be displayed when utilising HTTPS pages, yet they appear correctly on HTTP pages. This situation can be attributed to two reasons.

1. Mixed content: When a Web page contains encrypted HTTPS elements and unencrypted HTTP elements, only encrypted HTTPS content is transmitted to protect user security. Because unencrypted HTTP content is vulnerable to third-party attacks, web browsers often block HTTP requests within web pages for the user's experience. Therefore, unencrypted HTTP content cannot be displayed on HTTPS pages. Mixed content usually occurs during a website's transition from HTTP to HTTPS. When some content continues using HTTP, it becomes incompatible with the secure HTTPS environment, resulting in its inability to be displayed on the web page.
2. Domain Name and SSL Certificate Mismatch: While applying for an SSL certificate, the Certification Authority records the webpage's domain name. The browser establishes a secure encrypted connection only when the domain name matches the one in the certificate. However, if the domain names do not match, the browser treats the data transfer as insecure. It prevents the user from accessing the webpage. In contrast, HTTP webpages do not require an SSL Certificate. There is no problem of a mismatch between the domain name and SSL Certificate so that the user can access the web page.

CS1102 - Course Project - 2023/2024 Semester B - Intelligence Girls

© Group Members : Kan Lok Yiu Dorothy (57138602) | Wong Pui Ling (57142603) | WONG Zi Yin (57838772) | Man Nga Chung (56220231)

**References**

1. What is HTTPS? (n.d.). Cloudflare. https://www.cloudflare.com/learning/ssl/what-is-https/
2. Alomgir Hossain. (2024, February 14). Download personal files security, safe file storage, access to files in cloud, encrypted folder animation with Alpha channel. for free. [Video] Vecteezy. https://www.vecteezy.com/video/38770548-personal-files-security-safe-file-storage-access-to-files-in-cloud-encrypted-folder-animation-with-alpha-channel
3. What is an SSL certificate? (n.d.). Cloudflare. https://www.cloudflare.com/learning/ssl/what-is-an-ssl-certificate/
4. What is TLS (Transport Layer Security)? (n.d.). Cloudflare. https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/
5. Differences Between HTTP and HTTPS. (2022, January 31). Sectigo. https://www.sectigo.com/resource-library/http-vs-https
6. What happens in a TLS handshake? | SSL handshake (n.d.). Cloudflare. https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/
7. What is mixed content? (2018, June 22). Outspoken Media. https://outspokenmedia.com/https/mixed-content/
8. Troubleshooting a Name Mismatch in Web Browser. (n.d.). SSL Store. https://www.thesslstore.com/knowledgebase/ssl-support/troubleshooting-name-mismatch-web-browser/