Chapter 11. Basic Issues in Web Security

Table of Contents

Objectives

At the end of this chapter you will be able to:

- Understand the need for web security;
- Understand some of the common web security vulnerabilities; and
- Understand some of the web security solutions.

11.1 Introduction to Web Security

When information transmitted over the web, not only does the data have reach its destination, but it needs to arrive intact and uncorrupted (**integrity**), and other people should be prevented from seeing it (**confidentiality**). The nature of the Internet makes directing information to reach its destination relatively trivial, but ensuring its integrity and confidentiality is more difficult. Fortunately, encryption algorithms have made both integrity and confidentiality feasible. Additionally, users like to know that the information they receive is genuine (**authentication**) and that the sender of the information cannot deny that they sent it (**non-repudiation**).

The web is an interconnection of networks. Everybody uses the Internet to transfer data and that the data has value (and cost), and so it is a subject to theft. Types of information that are stolen include personal user's information, commercial or technical data (including commercial secrets and intellectual property), or even security and military information. Leaking of such information can stay undiscovered for months, if not years, doing damage to people that sent information and also to third parties.

11.1.1 Why the Internet is Insecure?

One of the main reasons for such vulnerabilities is the fact that web application developers are often not very well versed with secure programming techniques. As a result, security of the application is not necessarily one of the design goals. This is exacerbated by the rush to meet deadlines in the fast-moving e-commerce world.

The Internet is a packet-passing network, and so information sent from one machine to another passes through

many intermediate machines as the data is routed towards its destination. These intermediate machines can see all the packets routed through them, as well as keep copies of the packets and possibly change their data content before passing them on. Information on a network or internetwork is clearly not confidential by nature.

It also means that the information's receiver cannot be sure that the information has been unchanged: in other words, there are doubts about the information's integrity.

As any intermediate machine may have changed the data, the data can also not be authenticated, and the original source can deny that they originally sent the data (they can repudiate the data).

While some of these problems are alleviated due to the nature of the Internet (since the various packets containing the data may go via different routes), they cannot be eliminated.

11.1.2 Why make information secure?

While a large portion of information on the Internet is meant to be widely shared (such as a company's website), there is also important information transmitted over the Internet that is meant to be private and secure.

Consider the needs of e-commerce, where private information, such as credit card details, are transmitted online.

When consumers purchase goods via credit card, they do not want any intermediate people to know their credit details.

Generally, any important information sent over the Internet should be secured in some way. There are obviously different types of information, and some need more security than others.

The important issues around obtaining a credit card number from a customer are:

- If the transmission of the credit card details isn't confidential, customers are open to credit card fraud.
- If the data's integrity isn't assured, then their credit details, or their purchase information, may be invalid.
- If the communication details cannot be authenticated, then there is no guarantee that the purchased products are being sent to the right person.
- If there is no non-repudiation, the customer can deny that they ordered the product once they have received it, and cancel the credit card payment.

Exercise 1

How would you go about trying to make messages secure? Consider the realm of popular books and movies, can you think of any examples where information is made secure? What were the means used to secure information in your examples?

You can find a discussion of this exercise at the end of the chapter.

11.2 Common vulnerabilities

Some of the top five security vulnerabilities are SQL injection; buffer overflow; sensitive data exposure broken authentication and session management; and security misconfiguration[1]. These are briefly discussed below.

11.2.1 SQL Injection

SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of a web application.

A Database is the heart of many web-applications and is used to store information needed by the application, such as, credit card information, customer demographics, customer orders, client preferences, etc. SQL Injections happen when a developer accepts user input that is directly placed into a SQL Statement and doesn't properly validate and filter out dangerous characters. This can allow an attacker to alter SQL statements passed to the

[1] http://resources.infosecinstitute.com/the-top-five-cyber-security-vulnerabilities-in-terms-of-potential-for-catastrophic-damage/

database as parameters and enable her to not only steal data from your database, but also modify and delete it.

A database is vulnerable to SQL injections when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed. SQL injection attacks are also known as SQL insertion attacks.

11.2.2 Buffer Overflow

A buffer overflow vulnerability condition exists when an application attempts to put more data in a buffer than it can hold. Writing outside the space assigned to buffer allows an attacker to overwrite the content of adjacent memory blocks causing data corruption, crash the program, or the execution of an arbitrary malicious code

11.2.3 Sensitive Data Exposure

Sensitive data exposure occurs every time a threat actor gains access to the user sensitive data. Data could be stored (at rest) in the system or transmitted between two entities (i.e. servers, web browsers), in every case a sensitive data exposure flaw occurs when sensitive data lack of sufficient protection. Sensitive data exposure refers the access to data at rest, in transit, included in backups and user browsing data. The attacker has several options such as the hack of data storage, for example by using a malware-based attack, intercept data between a server and the browser with a Man-In-The-Middle attack, or by tricking a web application to do several things like changing the content of a cart in an e-commerce application, or elevating privileges.

11.2.4 Broken Authentication and Session Management

The exploitation of a broken Authentication and Session Management flaw occurs when an attacker uses leaks or flaws in the authentication or session management procedures (e.g. Exposed accounts, passwords, session IDs) to impersonate other users. This kind of attack is very common.

11.2.5 Security Misconfiguration

Below some typical example of security misconfiguration flaws:

- Running outdated software.
- Applications and products running in production in debug mode or that still include debugging modules.
- Running unnecessary services on the system.
- Not configuring problems the access to the server resources and services that can result in the disclosure of sensitive information or that can allow an attacker to compromise it.
- Not changing factory settings (i.e. default keys and passwords).
- Incorrect exception management that could disclose system information to the attackers, including stack traces.
- Use of default accounts.

11.3 Web Security Solutions

Two main tasks of any web security solution is to:

- Provide correct identification of the remote side in network conversation; and
- Prevent third parties that have possibility to access the network, over which the data is transmitted, from accessing the data being sent.

Some of the web security solutions that allow the client and the server to securely exchange information and minimize the chance for attack are described below.

11.3.1 HTTPS

There are several widely used protocol schemes available. They are SSH (Secure Shell) and SSL (Secure Socket Layer/Transport Level Security). Both protocols work on transport network level ("above" TCP protocol) and utilize similar schemes. SSL is more widely used because of its adoption for secure WWW data transfer. Both protocols provide transparent security; this allows use of standard Internet protocols over SSL or SSH.

The most well-known application for SSL protocol is securing commercial Internet communications. Most of commercial web sites offer an option (or even force) for use of SSL, which is used for HTTPS protocol. This is however not the only protocol to use SSL. Actually most TCP-based protocols (like POP3 and IMAP for mail, NNTP for news etc.) can work over SSL. SSH is also used to provide security for FTP and shell protocols.

HTTPS

Hypertext Transfer Protocol Secure or Hypertext Transfer Protocol over SSL is used for secure communication over a network, or perhaps more importantly – over the Internet. You would see https:// in the URL and a lock icon in the browser when you access a page that uses HTTPS. Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms.

Consider developing an e-commerce website that requires your users to enter sensitive information, such as credit card details, in order to proceed with an online transaction. If the information travels over the Internet as is and is intercepted by someone, it could be easily understood and misused. This is where HTTPS comes in – if you need to prevent these types of threats, you need to go HTTPS.

HTTPS promises you two things; first, the sensitive data is encrypted into gibberish by applying a cryptography mechanism which can be decrypted only by your server, the certificate owner. Now, if this information is intercepted with a man-in-the-middle attack, it will be meaningless. Secondly, HTTPS authenticates that the website is the website it claims to be. In your case, it will validate your website before sending your user's encrypted credit card details so no one else can imitate you.

Thus, going HTTPS authenticates your website and protects sensitive information being communicated over the Internet. This is made possible with the help of Certificates and Encryption.

11.3.2 Certificates

In order for you to go HTTPS, you need a Certificate. It is a digital document that your website submits to proclaim your identity to the user, the web browser. The certificates are issued by companies known as Certificate Authorities (CA) which will encrypt your web related information such as your domain name, server platform and identity information such as company's name address, phone number etc. within the certificate. You may wonder how a browser would trust a certificate. All browsers come with a set of pre-installed information letting them know of trusted certificate authorities. When you go HTTPS, you'll have your certificate in your server which will be sent to your user whose browser will certify you.

11.3.3 Encryption

We know that HTTPS encrypts data before sending it over the Internet and the server decrypts it. In the encryption-decryption scenario, a pair of keys is involved. One is public and the other is private. When your website wants your user to send information, your server instructs the user's browser with a key (public) to encrypt the data which is to be sent over. Once the encrypted message is received, the server will use its private key to decrypt and understand the data. In HTTPS, any plain text encrypted with the public key can only be decrypted by the holder of the private key.

11.4 Discussion

EXERCISE 1

Both these examples are based upon securing verbal information by means of a code.

1. In the movie Mission Impossible, the hero breaks into the CIA's headquarters and steals a list of all the undercover agents around the world. This is obviously very important information and it should be kept secure. The security measures the CIA used were extensive, and required a lot of effort to overcome them.
2. In the book, Enigma, by Robert Harris — a novel based closely on Turing and the work at Bletchley Park in the UK during the Second World War. The 'Enigma' machine was a mechanical coding device used to pass messages between U-boats and their command headquarters. In times of war, secure communication is perhaps extra important. The cracking of the code at Bletchley Park was greatly assisted by the capture of an Enigma machine and various decoding pamphlets from an abandoned German U-boat. Alan Turing, the mathematician and computer pioneer, was a leading figure in this work.