SSL/TLS Vulnerability Detection Using Black Box Approach

To cite this article: D Gunawan et al 2018 J. Phys.: Conf. Ser. 978 012121

View the article online for updates and enhancements.
SSL/TLS Vulnerability Detection Using Black Box Approach

D Gunawan*, E H Sitorus, R F Rahmat, A Hizriadi

Department of Information Technology, Universitas Sumatera Utara, Jl. dr. Mansur No. 9 Kampus USU Medan 20155

*Email: danigunawan@usu.ac.id

Abstract. Socket Secure Layer (SSL) and Transport Layer Security (TLS) are cryptographic protocols that provide data encryption to secure the communication over a network. However, in some cases, there are vulnerability found in the implementation of SSL/TLS because of weak cipher key, certificate validation error or session handling error. One of the most vulnerable SSL/TLS bugs is heartbleed. As the security is essential in data communication, this research aims to build a scanner that detect the SSL/TLS vulnerability by using black box approach. This research will focus on heartbleed case. In addition, this research also gathers information about existing SSL in the server. The black box approach is used to test the output of a system without knowing the process inside the system itself. For testing purpose, this research scanned websites and found that some of the websites still have SSL/TLS vulnerability. Thus, the black box approach can be used to detect the vulnerability without considering the source code and the process inside the application.

1. Introduction

Security plays crucial role in data communication. Socket Secure Layer (SSL) and Transport Layer Security (TLS) are cryptographic protocols that are intended to secure data communication over a network by providing data encryption. However, security threat not only occurs over data communication. For example, heartbleed [1] as one of the most vulnerable SSL/TLS bugs in OpenSSL is caused by the absence of payload length checking by the server. By exploiting this vulnerability, an attacker can request up to 64,000 characters. This research focus on heartbleed case to demonstrate black box approach in detecting SSL/TLS vulnerability. In addition, this research also implements additional SSL verification to gather information about existing SSL in the server. Those are the type of hashing algorithm namely SHA1 and the self-signed status. According to Netcraft and Symantec, SHA1 is known as “insecure” hashing algorithm that is still being used by nearly a million of SSL certificates around the globe [2][3]. Moreover, the self-signed status of SSL certificate is known as a bad practice because it only provides encryption without authentication. There is no way to authenticate the server because the certificate is issued by the server itself, not by the trusted Certificate Authority (CA).

Recent research using black box approach [4] to detect weakness and to test system effectivity in detecting vulnerability such as cross-site scripting, SQL injection and so on. Another research utilized black box approach to identify web application vulnerability and group the attacking scenarios [5]. This method was proposed to reduce the number of false positives. By using black box method, this allow the application to find a new page in the web which may contain vulnerability.

Another research proposed clustering approach to find vulnerabilities in web applications [6]. This research is proved as the alternative utilization of clustering algorithm [7]. This research was not
intended to search the website vulnerability. However, this research analyses the structure of a web application, group several application and then finding anomaly from the structure.

This paper is organized as follow: the first section is the introduction about the background of the conducted research. Then, section two discusses the process inside the black box approach. Section three discusses the implementation of the black box approach. The last section is conclusion.

2. Research Methodology
This research is intended to detect Socket Secure Layer (SSL) or Transport Layer Security (TLS) vulnerability in web application by applying black box approach. Black box approach will apply a series of procedures to verify and detect SSL/TLS vulnerability in web applications. This research will focus on heartbleed vulnerability and detecting SSL/TLS version. The black box testing includes two procedures in order to scan a web application, such as heartbleed scan and SSL verification.

The first procedure, scanning heartbleed requires three additional steps: handshake, attack scenario and vulnerability identification [8]. These steps are depicted in figure 1.

A. Handshake
The first step is connecting to the server based on certain Uniform Resource Locator (URL). The handshake process will negotiate automatically and then dynamically determine parameters to build communication channel between normal entities before the communication through that channel is begun. This step is required to ensure the communication has been established between server and clients. In this step scanner will send a message to web server and ensure the channel or protocol that will be used between client and server when communicating. As shown in figure 1, the handshake process is started with a scanner sends “HELLO” message to the server. If the handshake process is successful, the server will response with “HELLO” message as well.
B. Attack Scenario
Figure 1 shows that attack scenario is commencing after handshake process and the server has responded it to the scanner. In this scenario, scanner will deliver heartbleed packet in hexadecimal format, which will be converted to byte before it is sent to the server. This heartbleed packet contains content type, TLS version, length, handshake type and payload length. The attack process is begun by deceiving heartbeat extension in the server. The scanner will send response in three bytes but require response in more than three bytes.

C. Vulnerability Identification
The next step is the scanner will read the web server response to the heartbleed packet that is sent previously. Web server will response header parameters which contain type, payload and length. Type and payload value will determine that the server is vulnerable to heartbleed or not. If the server is determined as vulnerable, the scanner (or in the real world is the attacker) may read the hexadecimal dump memory to view memory server contents.

The example of header data that will be sent by the server to the scanner is as follow:
- Type = 24 and payload > 3. This means that the server is vulnerable to heartbleed bug. Server will response with the memory server contents.
- Type = 24 and payload < 3. This means that the server is vulnerable because the server response the heartbleed request by the scanner.
- Type 21 or type = 0. This means server error or server good.

The second procedure is SSL verification. SSL is used to provide data encryption in data communication. This is an additional procedure which is used to gather the information about existing SSL in the server. By gathering this information, we will know the type of signature algorithm in the server, signature expiry date, and certificate chain. The type of signature algorithm determines the strength of data encryption. Signature expiry date will cause an error and warning that the service is not guaranteed to be secured. Certificate chain which is used by client and server also determines the strength the SSL security. Certificate can be issued by the Certificate Authority (CA) or by the server itself (also known as self-signed certificate). To ensure that a certificate is issued by the CA, the scanner should verify subjectdn and issuerdn.

Therefore, the scanner will try to apply private and public key verification to obtain the information about certificate chain status. SSL verification can also be used to obtain the type of cryptography algorithm which is applied in SSL certificate to provide secure communication channel. The information gathered by the second procedure can be used as analysis and supporting the first procedure. Although the result of the scanning shows that the website or web application is still using obsolete signature algorithm or self-signed certificate, this does not mean that the website or web application is very insecure. This means that the website or web application requires improvement in certificate algorithm or using the certificate issued by the CA.

3. The Implementation of SSL/TLS Vulnerability Detection
This section will discuss the implementation of the vulnerability detection by using the scanner. The vulnerability detection is tested to the secure websites and insecure websites. This experiment shows that the SSL/TLS vulnerability especially for heartbleed bug can be detected without knowing how the target’s business process is running. Heartbleed occurs because of a bug in request handling by the heartbeat extension in OpenSSL. This extension is used to maintain client and server communication. Because of this bug, the server does not check the message and the length of the message which are requested by the client. Therefore, client can freely obtain the server memory contents by requesting certain payload length.

The result of the implementation of SSL/TLS vulnerability detection is shown in table 1. These data are taken in the late 2015. According to the data testing, there are several websites that are still suffering
to heartbleed vulnerability. In addition, several websites are still using “insecure” SHA1 algorithm and using self-signed certificate.

<table>
<thead>
<tr>
<th>Web Address</th>
<th>Heartbleed</th>
<th>SSL Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://www.koreabridge.net">https://www.koreabridge.net</a></td>
<td>✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://www.voobys.com">https://www.voobys.com</a></td>
<td>✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://worthytoshare.net">https://worthytoshare.net</a></td>
<td>✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://www.uns.ac.id">https://www.uns.ac.id</a></td>
<td>-</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://www.ugm.ac.id">https://www.ugm.ac.id</a></td>
<td>-</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://www.ui.ac.id">https://www.ui.ac.id</a></td>
<td>-</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://mail.usu.ac.id">https://mail.usu.ac.id</a></td>
<td>-</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://news.detik.com">https://news.detik.com</a></td>
<td>-</td>
<td>✓ ✓</td>
</tr>
<tr>
<td><a href="https://www.olx.co.id">https://www.olx.co.id</a></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="https://www.its.ac.id">https://www.its.ac.id</a></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="https://www.lazada.co.id">https://www.lazada.co.id</a></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="https://www.kaskus.co.id">https://www.kaskus.co.id</a></td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><a href="https://www.bukalapak.com">https://www.bukalapak.com</a></td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

4. Conclusion
Socket Secure Layer (SSL) and Transport Layer Security (TLS) are used to secure the data communication. However, some websites or web applications might use obsolete version of SSL/TLS that may lead to security vulnerability. This research shows black box approach can be used to test or detect SSL/TLS vulnerability without knowing business process inside the websites or web applications. Black box approach can be used as complement to another testing scheme.

References
[3] Symantec 2014 SHA1 certificate shown as insecure or with mix content warning on Google Chrome 39 Available: https://www.symantec.com/connect/blogs/sha1-certificate-shown-insecure-or-mix-content-warning-google-chrome-39