**LibreSSL**

OpenSSL was affected by a severe vulnerability popularly known as the Heartbleed vulnerability. Due to the vulnerability mentioned and the multiple that followed affecting OpenSSL, multiple resolutions were applied under extreme urgency which led to an overall compromise of the security of OpenSSL by design. This is when LibreSSL was introduced as a fork of OpenSSL. It was intended and created to take the place of OpenSSL. As per the OpenBSD project, the goals of LibreSSL was to modernize the code, improve security and introduce best practices to software development. [1]

As far as modernization is concerned. LibreSSL attempts to counter the issues associated to support for C functions by using a compiler associated to the standard and also by assuming that the system that LibreSSL is going to be implemented on to be OpenBSD. In other words, due to conformity to the C standard, it allowed LibreSSL to get rid of code that was not associated to the standard. LibreSSL also removed the memory management layer that existed in OpenSSL which also helped discover multiple unreported buffer overflow errors which could then be resolved. One main goal of LibreSSL is to enhance the security. This was achieved by improving the readability of the underlying code, getting rid of dangerous features and the cleaning of memory which is missing in OpenSSL. Unfortunately, even after the positive changes were introduced to combat the issues surrounding OpenSSL, LibreSSL was not able to become FIPS compliant simply because FIPS object module was also one of the components that LibreSSL removed from the original OpenSSL implementation. [1]

Chart, bar chart

Description automatically generated

Fig 1: Performance test between OpenSSL and LibreSSL [4]

The above figure shows the performance difference between OpenSSL implementations compared against that of LibreSSL. The performance difference is based off the efficiency achieved by LibreSSL due to the memory sanitization within its design. Additionally, there are other features that have been heavily optimized for better performance as compared to OpenSSL. [4]

**Vulnerabilities**

Among multiple vulnerabilities, two of the more popularly known vulnerabilities are mentioned below.

1. CVE 2014-9424: This is a vulnerability that causes a Denial of Service situation on the target machine. This is caused due to a double free vulnerability which is essentially a flaw that attempts to free a memory location twice. A function within versions of LibreSSL before the version 2.1.2 enables adversaries to DoS a system by making use of length-verification error which exists during the DTLS handshake. A DTLS handshake is essentially the preliminary connection request that is sent between the RDG server and the client to allow a secure encrypted connection. [2]
2. CVE 2015-5333: The vulnerability marked by this CVE causes a buffer overflow vulnerability which ultimately leads to a DoS condition on the target machine. The vulnerability starts by calling the function OBJ\_obj2txt() which was initially known to be vulnerable to a memory leak vulnerability as well. It is then accessed via the X509 function which is finally used to decode the X509 certificate that is present during the preliminary communication i.e. handshake. [3]

**References**

1. F. Khan, “LibreSSL: The Secure OpenSSL Alternative,” *Infosec Resources*, Oct. 21, 2015. https://resources.infosecinstitute.com/topic/libressl-the-secure-openssl-alternative/ (accessed Mar. 18, 2023).
2. “Red Hat Customer Portal - Access to 24x7 support and knowledge,” *access.redhat.com*, Dec. 28, 2014. https://access.redhat.com/security/cve/CVE-2014-9424 (accessed Mar. 19, 2023).
3. “Full Disclosure: Qualys Security Advisory - LibreSSL (CVE-2015-5333 and CVE-2015-5334),” *seclists.org*, Oct. 15, 2015. https://seclists.org/fulldisclosure/2015/Oct/75 (accessed Mar. 19, 2023).
4. F. Khan, “LibreSSL: The Secure OpenSSL Alternative,” *Infosec Resources*, Feb. 21, 2015. https://resources.infosecinstitute.com/topic/libressl-the-secure-openssl-alternative/