# Survey of Remote TLS Vulnerability Scanning Tools and Snapshot of TLS Use in Banking Sector

*Jay Chung, Natalija Vlajic*
*Department. of Electrical Engineering & Computer Science*
*York University, Toronto, Canada*
jay0954@my.yorku.ca, vlajic@eecs.yorku.ca

***Abstract*** — With the increasing popularity and real-world use of TLS, the number of vulnerabilities identified in this protocol has also grown. As a result, the protocol has undergone several revisions, with TLS 1.3 being its latest and currently most secure version. In this paper we provide a brief review of some of the most critical vulnerabilities of the earlier versions of TLS (TLS 1.2 and 1.1), and we survey the performance of several popular TLS scanning tools. The paper also provides a summary of our findings obtained by performing remote TLS scanning of the world's 50 largest banks. Contrary to what one would expect, the state of TLS security in the surveyed banks appears to be at (or below) the state of TLS security across the whole WWW. For example, at present less than 50% of the surveyed banks deploy TLS 1.3, while a significant number of them appear vulnerable to some well-known TLS-based attacks.

***Keywords — TLS deployment, TLS-based attacks, TLS vulnerabilities, TLS remote scanning tools***

## I. Introduction: Evolution And Current Vulnerabilities of TLS

Over the last decade, TLS has become the most widely deployed security protocol – not only in the WWW domain, but also in other domains such as Cloud, IoT, etc. According to [1], presently 40.6% of websites supports TLS version 1.3, while TLS 1.2 and 1.1 are supported by 99.7% and 54.2% websites respectively. Unfortunately, TLS 1.2 and 1.1 have a number of known vulnerabilities, thus the websites supporting these versions of TLS are inherently insecure and could fall a victim to various TLS based attacks.

## II. Widely known TLS vulnerabilities

### *A. Vulnerable Key Exchange Algorithm*

In RSA Key Exchange (one of several key exchange methods in TLS 1.2) the client encrypts pre-master secret using the server's public key, and the server authenticates this message using its private key. However, if an attacker compromises the server's private key, while also owning a record of an earlier TLS session affiliated with this key, then it becomes possible to decrypt the entire session. As a result, it is said that RSA Key Exchange cannot ensure Perfect Forward Secrecy [2].

### *B. CBC Cipher Mode*

CBC is a block cipher mode in which each new block of plaintext is XORED with the previously encrypted text. CBC is associated with several vulnerabilities, including Padding Oracle Attacks. Attacks on TLS protocol that exploit CBC cipher mode are BEAST, SWEET32 and LUCKY13 attack.

### *C. RC4 Stream Cipher*

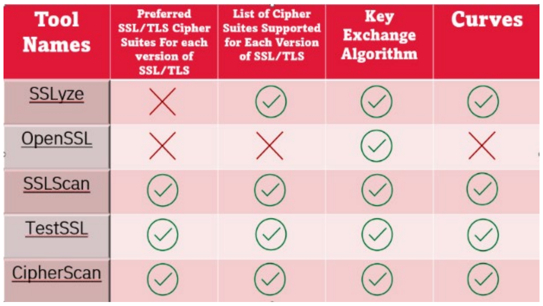
RC4 is the original stream cipher of TLS, and it generally enables much faster encryption and decryption of data compared to block ciphers. However, RC4 is vulnerable to *bit flipping attacks* and *distinguishing attacks*, and thus is no longer considered secure.

### *D. Downgrade Attack*

TLS Downgrade attack is a form of man-in-the-middle (MitM) attack in which the adversary manipulates a server into establishing a less secure version of TLS connection with a particular client, which then opens the door for various types of compromises. The most known downgrade attacks on TLS are POODLE and FREAK.

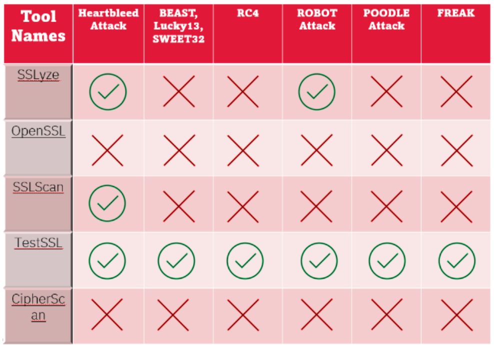
## II. Survey of remote TLS scanning tool

The goal of our research was to conducting a comparative analysis of five most popular TLS scanning tools (**SSLyze**, **OpenSSL**, **SSLScan**, **TestSSL**, **CipherScan**) in terms of their scanning capabilities in the following 3 categories: a) cipher-suite scanning, b) vulnerability scanning, and c) server-certificate scanning. The summary of our findings obtained through extensive hands-on experimentation with the five scanning tools are shown in Figure 1 to 3. Based on these findings, it is evident that TestSSL scanner provides most information and is also more user friendly than the other four tools. Consequently, TestSSL was the tool of choice for the next stage of our study outlined in Section III.



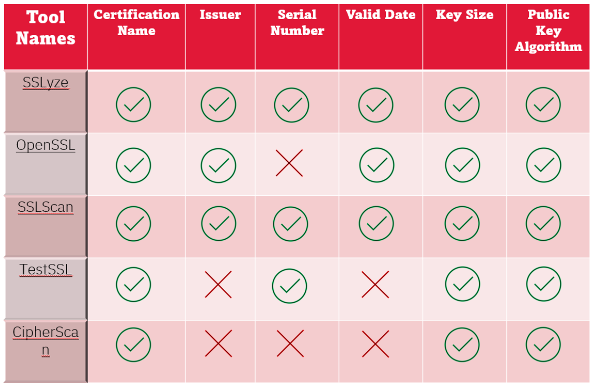
| Tool Names | Preferred SSL/TLS Cipher Suites For each version of SSL/TLS | List of Cipher Suites Supported for Each Version of SSL/TLS | Key Exchange Algorithm | Curves |
|---|---|---|---|---|
| SSLyze | ✗ | ✓ | ✓ | ✓ |
| OpenSSL | ✗ | ✗ | ✓ | ✗ |
| SSLScan | ✓ | ✓ | ✓ | ✓ |
| TestSSL | ✓ | ✓ | ✓ | ✓ |
| CipherScan | ✓ | ✓ | ✓ | ✓ |

*Figure 1. Cipher-suite scanning capabilities of 5 TLS scanners*



| Tool Names | Heartbleed Attack | BEAST, Lucky13, SWEET32 | RC4 | ROBOT Attack | POODLE Attack | FREAK |
|---|---|---|---|---|---|---|
| SSLyze | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| OpenSSL | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| SSLScan | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| TestSSL | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CipherScan | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

*Figure 2. Vulnerability scanning capabilities of 5 TLS scanners*



| Tool Names | Certification Name | Issuer | Serial Number | Valid Date | Key Size | Public Key Algorithm |
|---|---|---|---|---|---|---|
| SSLyze | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| OpenSSL | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| SSLScan | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TestSSL | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ |
| CipherScan | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ |

*Figure 3. Certificate scanning capabilities of 5 TLS scanners*

## IV. State of TLS Security In World's Top 50 Banks: Real-World Results

In this stage of our study we performed TLS vulnerability scanning of the world's 50 largest banks [3] (i.e., their front-end Web servers). The following is a summary of our findings:

### A. *TLS versions supported*

According to the conducted scans, about 49% of analyzed banks support TLS 1.3 and TLS 1.2 as their **highest/preferred TLS version** respectively (Figure 4). However, when it comes to the **overall support of various TLS versions** (Figure 5**)**, 94% of the surveyed banks (i.e., banking websites) support TLS 1.2, while only 49% of them support TLS 1.3. Unfortunately, 18% of the analyzed banking websites support TLS version 1.1 and 1.0 which should be retired due to their critical security vulnerabilities.

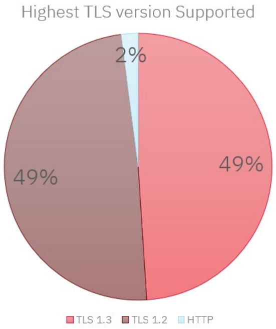


*Figure 4. Highest TLS version supported by 50 largest banks*

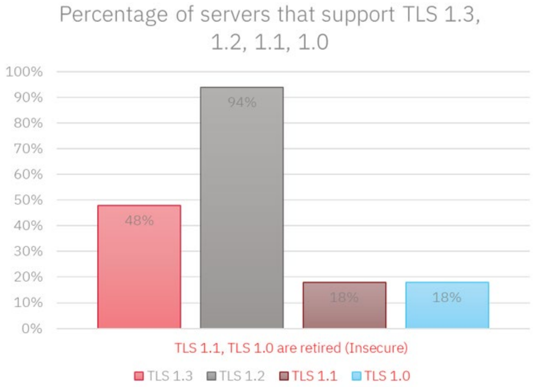


*Figure 5. Overall TLS versions supported by 50 largest banks*

### B. *Vulnerable encryption methods and algorithm*

The results pertaining to the vulnerabilities of deployed cryptographic algorithms by the 50 surveyed banks are shown in Figure 6. As evident from this figure, about 80% of banks support CBC cipher mode and thus could be vulnerable to BEAST, SWEET32 and Lucky13 attacks. 2% of banks use 64 Bit Block Cipher + DES, or RC2/4 with MD5, both of which are known to be vulnerable. Lastly, 4% of banks do not support perfect forward secrecy.

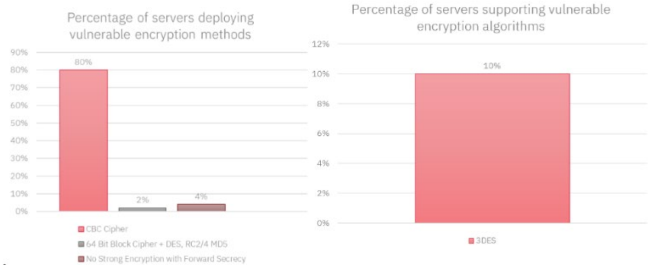


*Figure 6. Vulnerable encryption supported by 50 largest banks*

### C. *Vulnerability to TLS Attacks*

The potential vulnerability of the surveyed banks to some of the best known TLS attacks are outlined in Figure 7. Clearly, about 78% of the scanned banking websites are vulnerable to Lucky13 attack, 64% to BREACH, 18% to BEAST, 10% to SWEET32, and 2% to RC4 attacks.

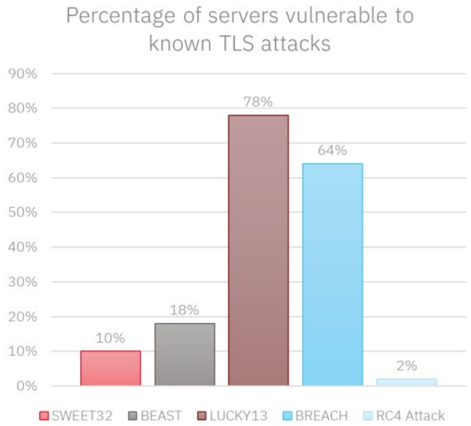


*Figure 7. Vulnerability to TLS attacks by 50 largest banks*

### D. *Conclusions*

By conducting remote TLS scanning of the world's 50 largest banks (i.e., their respective websites), we were able to observe that the majority of these sites (94%) are or could be vulnerable to TLS attacks due to supporting TLS 1.2 version of the protocol, while 18% of them are critically vulnerable due to supporting TLS 1.1 and 1.0. Furthermore, about 80% of the examined banking websites allow the use of CBC cipher mode, and 78% are specifically vulnerable to Lucky13 attack.

## References

[1] "SSL Pulse." *Qualys SSL Labs*, https://www.ssllabs.com/ssl-pulse/.

[2] IETF Network Working Group, "Deprecating Obsolete Key Exchange Methods in TLS", https://www.ietf.org/id/draft-ietf-tls-deprecate-obsolete-kex-00.html.

[3] https://en.wikipedia.org/wiki/List_of_largest_banks