**Apple Secure Transport**

Secure network communication using standardized transport layer security mechanisms. The Security.SecureTransport API gives you access to Apple's implementation of Secure Sockets Layer version 3.0 (SSLv3), Transport Layer Security (TLS) versions 1.0 through 1.2, and Datagram Transport Layer Security (DTLS) version 1.0.

Below are some instances of protocol-based and software-based vulnerabilities in the Apple SecureTransport library for the SSL/TLS protocol.

**Protocol-based vulnerabilities**

1. **"FREAK" vulnerability (CVE-2015-0204):** The SSL/TLS protocol had a design issue that allowed attackers to coerce clients into using poor encryption cyphers, which could then be used to intercept and decrypt SSL/TLS traffic.
2. **"Logjam" vulnerability (CVE-2015-4000):** A design weakness in the SSL/TLS protocol led to this vulnerability because it permitted attackers to utilise a weaker encryption protocol, which could then be used to intercept and decrypt SSL/TLS traffic.
3. **"POODLE" vulnerability (CVE-2014-3566):** Similarly, because of the design defect in the SSL/TLS protocol, attackers were able to intercept and decode SSL/TLS traffic by taking advantage of a hole in the SSLv3 protocol.

**Software-based vulnerabilities**

1. **"Gotofail" vulnerability (CVE-2014-1266):** A programming issue in the SecureTransport library led to SSL/TLS connections being formed without sufficient certificate checking, leading to the "Gotofail" vulnerability
2. **"Triple Handshake" vulnerability (CVE-2014-3566):** A programming mistake in the SecureTransport library led to the establishment of SSL/TLS connections using a less secure handshake protocol, which may be used to intercept and decrypt SSL/TLS communication.
3. **"Heartbleed" vulnerability (CVE-2014-0160):** The SecureTransport library for OS X, which makes use of the OpenSSL library, contained a bug that led to this vulnerability. By taking advantage of a weakness in the way OpenSSL processed specific heartbeat messages, the attackers were able to retrieve critical information from memory.

To fix these flaws in their devices, Apple has released patches and updates. In all the above vulnerabilities except Heartbleed both iOS and OS X devices were impacted.

**"FREAK" vulnerability (CVE-2015-0204)**

Researchers revealed the FREAK attack, a new SSL/TLS vulnerability, on March 3, 2015, on Tuesday. As a result, an attacker can force susceptible clients and servers to utilise weakened encryption, which the attacker can then crack to steal or alter sensitive data. This enables an attacker to intercept HTTPS communications between vulnerable clients and servers.

The term "FREAK vulnerability" describes a flaw in the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols that is brought on using "export-grade" encryption. "Factoring RSA Export Keys" is the name's acronym.

The threat enables a hacker to force a weak client to employ a weaker key exchange cypher, which would let them to intercept data transmission. Because the RSA encryption can be easily cracked when utilising 512 bits or fewer.

**Working of FREAK Vulnerability**

In general, the FREAK vulnerability makes it possible for hackers to intercept HTTPS traffic between clients and susceptible servers to access the private key of a website. As a result, they are now able to decode login cookies, passwords, credit card numbers, and other sensitive data from HTTPS connections.

The client is compelled to utilise an "export-grade" key, often known as a 512-bit export RSA key, which is significantly simpler to trace and crack than current encryption standards, thereby compromising secure connections.

**Actual Working -** In place of the conventional RSA cypher suites, an attacker can request "export RSA" via the client's Welcome message. The server then responds with a 512-bit export cypher key rather than the high-security keys used today. The answer is signed using its permanent key.

The Man-in-the-Middle attacker can obtain the RSA decryption key and utilise the "pre-master secret" to gain access to the TLS' master secret, which is used for symmetric encryption of messages in the connection, because the website client accepts the weak "export-grade" key. The core of command injection threats is when an attacker inserts harmful code into a plaintext file.

**Impact & Post-Discovery Action**

Millions of users have been exposed for years due to the FREAK vulnerability. Due to the vulnerability's size, it affected the entire sector and had a high potential for damaging cyberattacks. For instance, OpenSSL is used by numerous programmes, including browsers for Android. Apple's Secure transport, on the other hand, is used in both iOS and OS X applications, affecting iPhones, iPads, and Macs.

The FREAK issue was soon addressed by updates from OpenSSL, Google, and Apple.

The problem with computers and mobile devices was resolved by updates released by Apple's product security team. Additionally, it was discovered that the Safari browser proved resistant to the danger.

**Prevention**

It was urged to turn off support for any cyphers whose security was in doubt, including export versions of cypher suites. Also, there was detailed configuration advice for default configurations and suggestions for cypher suite enforcement laws.

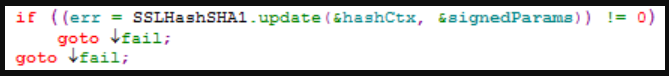
**Gotofail vulnerability (CVE-2014-1266)**

The [code](https://opensource.apple.com/source/Security/Security-55471/libsecurity_ssl/lib/sslKeyExchange.c) shows there was a bug in the implementation of SSLVerifySignedSeverKeyExchange function.

Graphical user interface, text

Description automatically generated with medium confidence

The issue was the two consecutive goto fail; statements. The second line is run regardless of whether the predicate in the if-statement is true or false, despite the indentation of the lines making it appear as though they will both be executed only if the predicate is true. Correcting the indentation makes the issue more visible.



When we call SSLHashSHA1.update it will return an error, the value of err will almost always be zero when the second goto fail; statement is executed. But when goto fail is executed, the caller gets a return value as zero, who believes the signature verification of the “Server Key Exchange” passed.

Text

Description automatically generated with low confidence

This vulnerability enables man-in-the-middle attack, the attackers spoof SSL servers by –

1. By using an arbitrary private key for the signing step or
2. Omitting the signing step.

The problem could be prevented in several ways such as –

1. Writing/compiling code.
2. Reviewing code
3. Testing

To overcome this vulnerability end users should apply the patches provided by Apple.

References

1. <https://developer.apple.com/documentation/security/secure_transport>
2. <https://freakattack.com/>
3. <https://crashtest-security.com/prevent-ssl-freak/>
4. <https://www.synopsys.com/blogs/software-security/understanding-apple-goto-fail-vulnerability-2/>
5. <https://nvd.nist.gov/vuln/detail/CVE-2014-1266>