colt

External TLS Attack Surface Review

November 2023

What is the potential business impact?

Data Breach and Unauthorised Access to Sensitive Data. An adversary can potentially access and manipulate all data transferred between the parties. This may include customer data, intellectual property, and internal communications.

Financial Loss. Any unauthorised manipulation or access to financial transactions could result in substantial financial loss. In addition, the cost of remediation of an attack can be substantial, including system repair and preventative measures.

Loss of Customer Trust and Damage to Brand Reputation. If an attack results in a data breach, particularly one that compromises customer data, this could result in lost revenue and a tarnished brand image.

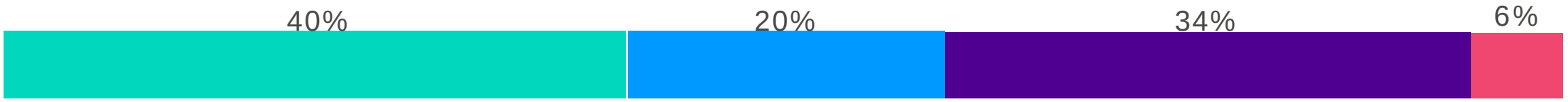
Regulatory Impact and Legal Liability. Data breaches can also result in businesses failing to comply with data protection regulations, such as the General Data Protection Regulation (GDPR) and other government privacy or industry-specific requirements, resulting in fines, legal action, and other legal complications.

Operational Disruption. A successful attack or a misconfigured certificate could disrupt business operations, particularly if systems are offline, to address the security breach. This may result in decreased productivity and missed business opportunities.

Key Findings

40 hosts were analysed, with the overall risk score being High. The following conclusions were made:

- No defined encryption standard could be determined.
- 21% of hosts complied with the Venari Security recommended standard.
- 39% of hosts complied with the NIST 800-52 recommended standard.
- 39% of hosts complied with the NCSC recommended standard.
- 36% of hosts used deprecated protocols that should not be used.
- 90% of hosts carried vulnerabilities that carry significant risk for the organisation.
- 79% of hosts offer weak encryption cipher suites.



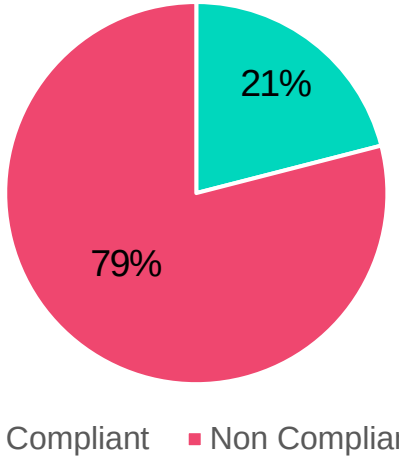
Low	Medium	High	Critical
Matches Venari Security best practice configuration. No relevant vulnerabilities or misconfigurations were detected.	One or more low-risk vulnerabilities or misconfigurations detected.	One or more high-risk vulnerabilities or misconfigurations detected.	A significant number of high-risk vulnerabilities or misconfigurations were detected.

Scores Against Standards Recommendations

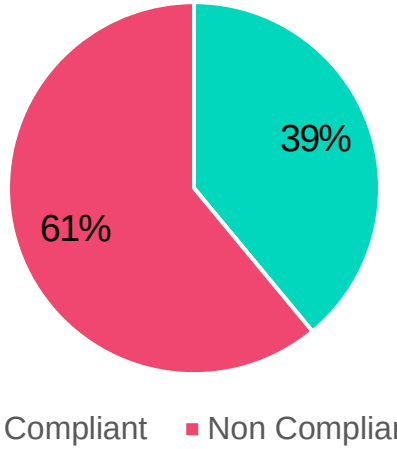
Venari Security has issued scores for three standards. The Venari Security Standard mitigates risk to businesses and guarantees adherence to optimal security, privacy and regulatory compliance practices. The standard employs robust encryption protocols and cipher suites without known vulnerabilities.

The NIST standard was published in 2019, followed by the NCSC standard in 2021. Venari Security expects NIST and NCSC to update their recommendations based on risks and vulnerabilities. Unlike the Venari Security standard, both NIST and NCSC allow using protocols and vulnerabilities that have been deprecated or contain vulnerabilities.

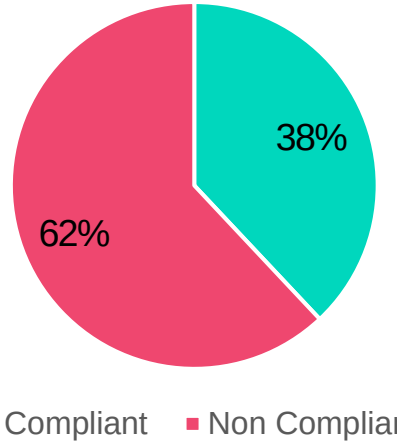
Venari Security Recommended



NIST 800-52, r2 Recommended



NCSC Recommended



Technical Risks

The below charts highlight the risk posed relating to certificates, encryption protocols and cipher suites.

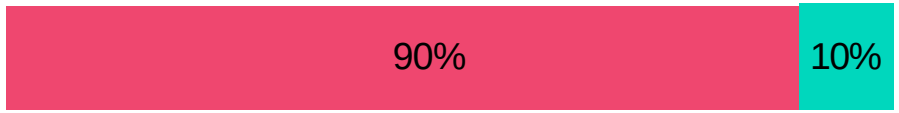
Percentage of hosts accepting protocols

Deprecated protocols are SSL V2, V3, TLS 1.0 and TLS 1.1. The IETF has deprecated these protocols and should not be used due to security concerns.

Percentage of hosts with certificate issues

Certificate issues include excessive certificate life span, expired certificates, soon to expire certificates, self-signed certificates.

Percentage of hosts that have vulnerabilities



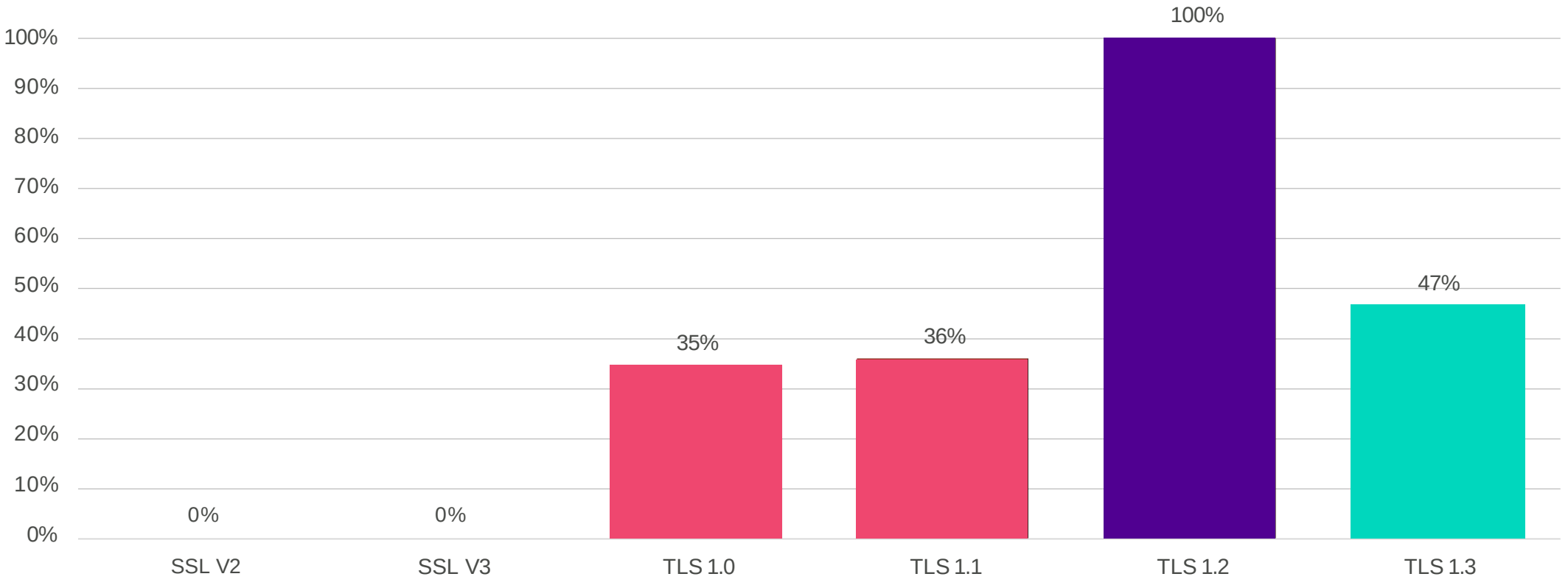
Hosts carrying vulnerabilities that could be exploited to attack servers.

Percentage of hosts that offer weak encryption

Weak encryption including cipher suites with known exploit vantages (ex. DES/3DES, SHA1, MD5, CBC, etc.)

Protocol Analysis

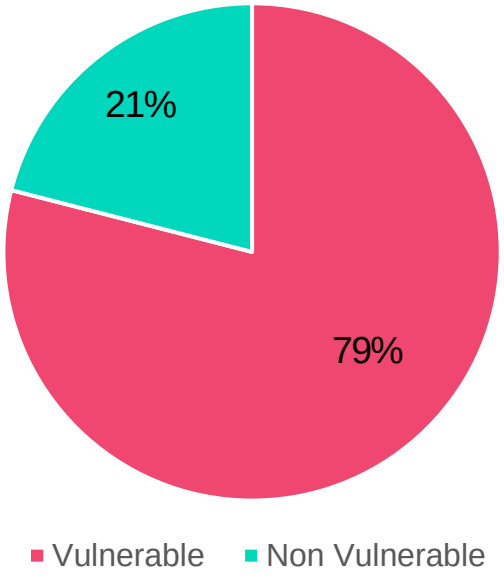
Below is a summary of the TLS protocols used in this analysis. Regarding security concerns, protocols including SSL V2, SSL V3, TLS 1.0, and TLS 1.1 are considered obsolete and should be avoided. For TLS 1.2, its usage is acceptable, provided it's paired with cipher suites devoid of known vulnerabilities.



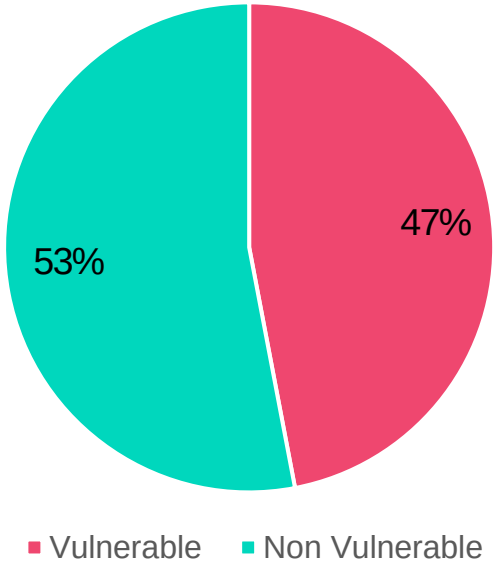
Vulnerability Analysis – Top 3

The following vulnerabilities have been identified. Below is a summary of the SSL/TLS vulnerabilities identified in the analysis. The charts below highlight the percentage of vulnerable hosts in red.

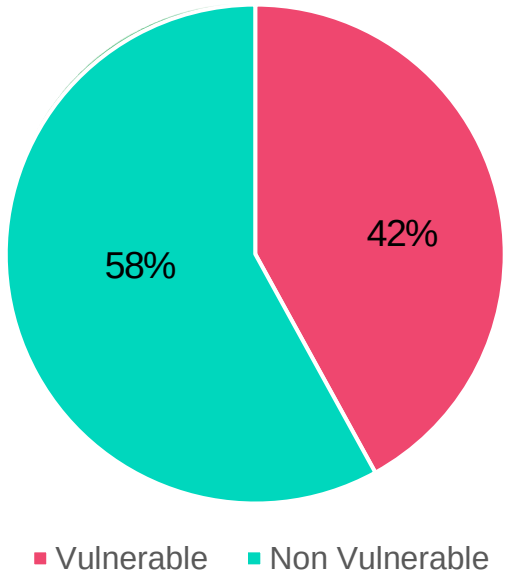
Lucky13



BEAST TLS



Insecure Renegotiation



Sweet32
3DES
CRIME
ROBOT
Ticketbleed

* The vulnerability definitions are available in the appendix

Venari Security Recommended Standard

Venari Security has established a standard to mitigate risk and maintain adherence to privacy, regulations and best practices within your organisation. This standard utilises modern TLS protocols and strong cipher suites, ensuring no known vulnerabilities exist. This design strengthens your organisation's security posture by preventing the potential exploitation of weak encryption protocols and cipher suites.

Guidelines for the selection, configuration and use of Transport Layer Security (TLS) implementations

- Accept only TLS 1.2 and above
- You should disable any SSL V3, SSL V2, TLS 1.1 and 1.0.
- Modern and strong cipher suites with no known vulnerabilities. The list of cipher suites are available in the appendix.

NIST Recommended Standard

In August 2019, NIST published the NIST 800-52 Revision 2. This standard intends to provide governmental bodies and the private sector with guidelines corresponding to industry best practices. It's crucial to highlight that TLS 1.1 is now considered outdated and poses significant security threats; hence, its usage is not advised.

Guidelines for the selection, configuration and use of Transport Layer Security (TLS) implementations

- TLS v1.0, SSL v2 and v3 must never be used.
- TLS v1.1+ is used when interoperability is required. The IETF deprecated TLS 1.1 in March 2021 and strongly recommends that it is not used.
- Only specific, recommended cipher suites should be used. Further information is available in the appendix.
- TLS v1.2 has been strongly recommended since 2016.
- TLS v1.3 should be supported by your services by 2024.

Venari Security expects the NIST 800-52 revision 2 to be updated in line with the recent changes by the US Government to ensure organisations use quantum-safe cryptography.

Link to NIST 800-52 revision 2

NCSC Recommend Standard

The NCSC published the “Using TLS to protect data” guide in July 2021. These recommendations aim to offer a standard for UK Government entities and private sector organisations that aligns with industry best practices. . It’s crucial to highlight that TLS 1.1 is now considered outdated and poses significant security threats; hence, its usage is not advised.

Guidelines for the selection, configuration and use of Transport Layer Security (TLS) implementations

- Disable TLS 1.1 and 1.0, SSL V3 and SSL V2
- Use TLS 1.2 as a minimum TLS protocol.
- Use TLS 1.3 and/or TLS 1.2, configured with the recommended profiles.
- Only specific, recommended cipher suites should be used. Further information is available in the appendix.
- All servers and clients should use the most up-to-date software version available. Implementation issues can introduce vulnerabilities that can be exploited if not patched promptly.
- Link to NCSC recommendations

Appendix

Venari Security “Recommended” cipher suites

Recommended TLS 1.3 Cipher Suites

TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256
TLS_CHACHA20_POLY1305_SHA256

Recommended TLS 1.2 Cipher Suites

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CCM
TLS_ECDHE_ECDSA_WITH_AES_256_CCM
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_CCM
TLS_DHE_RSA_WITH_AES_128_CCM
TLS_DHE_RSA_WITH_AES_128_CCM_8
TLS_DHE_RSA_WITH_AES_256_CCM_8
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_DH_DSS_WITH_AES_128_GCM_SHA256
TLS_DH_DSS_WITH_AES_256_GCM_SHA384
TLS_DH_RSA_WITH_AES_128_GCM_SHA256
TLS_DH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256

NIST 800-52 Rev2 “approved” Cipher Suites

Recommended TLS 1.3 Cipher Suites

TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256

Recommended TLS 1.2 Cipher Suites

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CCM
TLS_ECDHE_ECDSA_WITH_AES_256_CCM
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
TLS_DH_DSS_WITH_AES_128_GCM_SHA256
TLS_DH_DSS_WITH_AES_256_GCM_SHA384
TLS_DH_RSA_WITH_AES_128_GCM_SHA256
TLS_DH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_CCM
TLS_DHE_RSA_WITH_AES_256_CCM
TLS_DHE_RSA_WITH_AES_128_CCM_8
TLS_DHE_RSA_WITH_AES_256_CCM_8
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_DH_DSS_WITH_AES_128_CBC_SHA256
TLS_DH_DSS_WITH_AES_256_CBC_SHA256
TLS_DH_RSA_WITH_AES_128_CBC_SHA256
TLS_DH_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA2586
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA34

Recommended TLS 1.2, 1.1, 1.0 Cipher Suites

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DH_DSS_WITH_AES_128_CBC_SHA
TLS_DH_DSS_WITH_AES_256_CBC_SHA
TLS_DH_RSA_WITH_AES_128_CBC_SHA
TLS_DH_RSA_WITH_AES_256_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA

* Venari Security suggests avoiding these cipher suites due to known vulnerabilities

NCSC “approved” Cipher suites

NCSC Recommended Ciphers

TLS_AES_256_GCM_SHA38451
TLS_CHACHA20_POLY1305_SHA25651
TLS_AES_128_GCM_SHA25651

Sufficient Ciphers

TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA25652
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA25652
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA38452
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA25652
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA25652
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA

* Venari Security suggests avoiding these cipher suites due to known vulnerabilities

IETF Deprecated TLS Versions

SSL V2

With RFC 6176 the IETF officially depreciated SSL V2. With SSL V2 an active attacker can force an “early connection close” and the client has no way to know whether this is the genuine end-of-file, or a malicious truncate. Handshake messages are not protected, and additionally Message integrity and encryption are using the same key, both exploits make targets for man-in-the middle (MITM) attacks.

SSL V3

With RFC 7568 the IETF officially deprecated SSL V3 in June 2015. SSL V3 was suspended because it’s an old and very insecure protocol, the most famous issue being an exploit named ‘POODLE‘ found by Google in 2014. While this does require another MITM vantage the POODLE vulnerability allows an attacker to virtually “eavesdrop” on encrypted communication.

TLS 1.0

With RFC 8996 the IETF officially deprecated TLS 1.0 in March 2021. TLS 1.0 was superseded by TLS 1.1 in 2006 as a result of vulnerabilities detected in initialisation vector selection and padding error processing. These specifically helped to target vulnerabilities in CBC ciphers. TLS 1.0 also included support for a number of ciphers utilising SHA1 which has also been deprecated by IETF.

TLS 1.1

With RFC 8996 the IETF officially deprecated TLS 1.1 in March 2021. TLS 1.1 was superseded by TLS 1.2 in 2008 as a result of vulnerabilities detected in the protocol, including the removal of MD5 and SHA1 hashing algorithms, the IDEA and DES encryption.

IETF – Internet Engineering Task Force

Algorithms and support for AES encryption ciphers.

TLS Vulnerabilities

Sweet32 – CVE-2016-2183

Legacy block ciphers having a block size of 64 bits are vulnerable to a practical collision attack when used in CBC modes, such as DES, 3DES and RC4. Remote attackers can obtain cleartext data via a birthday attack against a long-duration encrypted session. Exposing PII or business- critical data to the attacker in plain text.

Lucky 13 – CVE-2013-0169

The TLS protocols 1.1 and 1.2 can allow malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets. The exploit only "…*requires about 223 TLS sessions to collect a whole block of TLS-encrypted plaintext in its most basic version.* ". Typically viewed as a man-in-the-middle (MITM) exploit can render PII & business-critical data to the threat-actor in plain text.

BEAST TLS – CVE-2011-3389

Short for Browser Exploit Against SSL/T LS, BEAST is a browser exploit against SSL 3.0/TLS 1.0 revealed in late September 2011. This attack leverages weaknesses in cipher block chaining (CBC) to exploit the Secure Sockets Layer (SSL) / Transport Layer Security (TLS) protocol. The CBC vulnerability can enable man-in-the-middle (MITM) attacks against SSL to silently decrypt and obtain authentication tokens, thereby providing attackers access to data passed between a Web server and the Web browser accessing the server.

3DES – CVE-2016-2183

The DES and Triple DES ciphers, as used in the TLS, SSH, and IPsec protocols and other protocols and products, have a birthday bound of approximately four billion blocks, which makes it easier for remote attackers to obtain cleartext data via a birthday attack against a long- duration encrypted session, as demonstrated by an HTTPS session using Triple DES in CBC mode, aka a "Sweet32" attack.

TLS Vulnerabilities

CRIME – CVE-2012-4929

A CRIME attack can be executed against SSL/TLS protocols (v1.2 and older) and the SPDY protocol to hijack a user's session cookies. At the same time, they're still authenticated to a website exposing sensitive & business-critical data to the attacker in plain text. With an MTIM exploit, browsers can encrypt compressed data without properly obfuscating the length of the unencrypted data. Attackers can obtain plaintext HTTP headers by observing length differences during a series of "guesses" in which a string in an HTTP request potentially matches an unknown string in an HTTP header.

ROBOT – CVE-2017-13099 and CVE-2017-13098

The ROBOT attack entails using a vulnerability in the RSA encryption to authorise operations with the private key of an SSL/TLS server. Specific to the wolfSSL library embedded in commercial products (an alternative to the OpenSSL library) that uses a weak Bleichenbacher oracle when any TLS cipher suite is negotiated when an RSA key exchange is used. Attackers can record traffic and decrypt it afterwards to access sensitive information.

RC4 – CVE-2013-2566

The RC4 algorithm, as used in the TLS and SSL protocols, has many single-byte biases, making it easier for remote attackers to conduct plaintext-recovery attacks via statistical analysis of ciphertext in a large number of sessions that use the same plaintext. Many legacy commercial systems still have (default) support for RC4 enabled "…so as NOT to have compatibility issues…" and is up to the administrators to disable.

Ticketbleed – CVE-2016-9244

Ticketbleed is a software vulnerability in the TLS/SSL stack of F5 BIG-IP appliances allowing a remote attacker to extract up to 31 bytes of uninitialised memory at a time. This memory can potentially contain key material or sensitive data from other connections.

TLS Vulnerabilities

SLOTH – CVE-2015-7575

Leverages the Mozilla Network Services library (often seen use in Linux-based hosts) does not reject MD5 signatures in Server Key Exchange messages in TLS 1.2 Handshake Protocol traffic. This facilities a man-in-the-middle (MITM) attack vector to spoof servers by triggering a collision, thereby gaining control of a network session, leading to potential data loss in transit.

BREACH – CVE-2013-3587

The HTTPS protocol, as used in unspecified web applications, can encrypt compressed data without properly obfuscating the length of the unencrypted data, which makes it easier for man-in-the-middle attackers to obtain plaintext secret values by observing length differences during a series of guesses in which a string in an HTTP request URL potentially matches an unknown string in an HTTP response body, aka a "BREACH" attack, a different issue than CVE-2012-4929.

DROWN – CVE-2016-0800

This attack, DROWN (Decrypting RSA using Obsolete and Weakened eNcryption), uses a form of Bleichenbacher attack that enables the decryption of RSA ciphertexts. It does depend upon SSLv2, which SHOULD be depreciated; specific OpenSSL vulnerabilities discovered have been designated CVE-2015-3197, CVE-2016-0703, and CVE-2016-0704 in conjunction with DROWN. Since this was released, OpenSSL has made it impossible to configure a TLS server suspectable to DROWN. If unpatched or legacy deployments exist, they still have this exploit open. Insecure Client Renegotiation. The SSL/TLS renegotiation vulnerability is a cyber threat in cases when a client can initiate a renegotiation process. An attacker can abuse this situation by making the server unavailable with a Denial of Service attack or can execute a Man-in-the-Middle injection attack into the HTTPS sessions.

Thank you