DEFINITIONS

"Data in Motion" (also referred to as "Data in Transit") is any data being transferred over a network. Examples include data sent over email, data accessed through a web portal, and data being sent through a server-to-server communication process.

"Encryption" is a technology that protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people.

"*Transport Layer Security"* or ***"TLS"*** is a protocol created to provide authentication, confidentiality, and data integrity between two communicating applications.

"Cipher Suite" is a set of algorithms, including a cipher, a key-exchange algorithm and a hashing algorithm, which are used together to establish a secure TLS connection.

STANDARD

1) Restricted Information (RI) must be encrypted when transmitted across external open networks, such as the Internet, to protect against interception of network traffic by unauthorized individuals.
2) RI should be encrypted when transmitted within internal networks such as MedNet whenever possible. Protocols that cannot support encryption (including but not limited to DICOM, HL7, SMB) will be excepted if there are no reasonable alternatives.
3) All passwords and other authentication secrets must be encrypted in transit.

TECHNICAL REQUIREMENTS

1) For applications using TLS for encryption, NIST SP 800-52 R2 guidelines must be followed.
   a) TLS 1.2 must be supported. TLS 1.3 should be supported by January 2024.
   b) TLS version 1.1 is no longer supported, and must be disabled in existing applications as soon as is practical. New applications must not allow TLS 1.1.
   c) TLS 1.0 must be disabled.
   d) Only cipher suites listed in NIST SP 800-52 R2 shall be used. Please see Appendix 1 for the list of recommended TLS Cipher Suites.
2) Valid encryption processes for remote access across the Internet are those which comply, as appropriate, with NIST Special Publication 800-77, Guide to IPsec VPNs; or NIST Special Publication 800-113, Guide to SSL VPNs.
3) For Secure Shell (SSH), the guidelines of NIST IR 7966, Security of Interactive and Automated Access Management Using Secure Shell (SSH), should be followed.

Only SSH version 2.0 should be used. Note that SSH is the underlying protocol for SFTP.

EXCEPTIONS

Any exceptions from this standard must be approved by the UCLA Health Sciences Chief Compliance Security Officer.

QUESTIONS

1) If you have questions on how to configure TLS for different OS and/or applications, please contact your IT Support Group.
2) If you have questions about this Standard, please contact the Office of Compliance Services – Information Security (CompOffice@mednet.ucla.edu)

REFERENCES

HS Policy No. 9457, "Minimum Security Standards for Network Devices"

National Institute of Standards and Technology, SP 800-52 Rev. 2 *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations.* https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final

National Institute of Standards and Technology, SP 800-77, *Guide to IPsec VPNs*. https://www.nist.gov/publications/guide-ipsec-vpns-recommendations-national-institute-standards-and-technology

National Institute of Standards and Technology, SP 800-113, *Guide to SSL VPNs*. https://csrc.nist.gov/publications/detail/sp/800-113/final

National Institute of Standards and Technology, IR 7966, *Security of Interactive and Automated Access Management Using Secure Shell (SSH)*. https://csrc.nist.gov/publications/detail/nistir/7966/final

Office for Civil Rights, *Guidance to Render Unsecured Protected Health Information Unusable, Unreadable or Undecipherable to Unauthorized Individuals* https://www.hhs.gov/hipaa/for-professionals/breach-notification/guidance/index.html

CONTACT

Chief Compliance Security Officer, Office of Compliance Services

REVISION HISTORY

Created Date: February 7, 2019

Updated Date: November 12, 2020

APPENDIX 1: Recommended TLS cipher suites

UCLA Health shall follow the NIST guidelines as set out in NIST SP 800-52 R2, https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final for cipher suites. Specifically, NIST provides a complete list of approved cipher suites, some of which are required and some of which are recommended. This is the list provided below. It is not required to support all the cipher suites but cipher suites that do not appear on this list shall not be used.

TLS 1.2		
NIST/IANA	**OpenSSL**	**Hex Code**
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	ECDHE-ECDSA-AES128-GCM-SHA256	(0xC0, 0x2B)
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	ECDHE-ECDSA-AES256-GCM-SHA384	(0xC0, 0x2C)
TLS_ECDHE_ECDSA_WITH_AES_128_CCM	ECDHE-ECDSA-AES128-CCM	(0xC0, 0xAC)
TLS_ECDHE_ECDSA_WITH_AES_256_CCM	ECDHE-ECDSA-AES256-CCM	(0xC0, 0xAD)
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	ECDHE-ECDSA-AES128-CCM8	(0xC0, 0xAE)
TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8	ECDHE-ECDSA-AES256-CCM8	(0xC0, 0xAF)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	ECDHE-ECDSA-AES128-SHA256	(0xC0, 0x23)
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	ECDHE-ECDSA-AES256-SHA384	(0xC0, 0x24)
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE-ECDSA-AES128-SHA	(0xC0, 0x09)
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE-ECDSA-AES256-SHA	(0xC0, 0x0A)
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	ECDHE-RSA-AES128-GCM-SHA256	(0xC0, 0x2F)
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	ECDHE-RSA-AES256-GCM-SHA384	(0xC0, 0x30)
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	DHE-RSA-AES128-GCM-SHA256	(0x00, 0x9E)
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	DHE-RSA-AES256-GCM-SHA384	(0x00, 0x9F)
TLS_DHE_RSA_WITH_AES_128_CCM	DHE-RSA-AES128-CCM	(0xC0, 0x9E)
TLS_DHE_RSA_WITH_AES_256_CCM	DHE-RSA-AES256-CCM	(0xC0, 0x9F)
TLS_DHE_RSA_WITH_AES_128_CCM_8	DHE-RSA-AES128-CCM8	(0xC0, 0xA2)
TLS_DHE_RSA_WITH_AES_256_CCM_8	DHE-RSA-AES256-CCM8	(0xC0, 0xA3)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	ECDHE-RSA-AES128-SHA256	(0xC0, 0x27)
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	ECDHE-RSA-AES256-SHA384	(0xC0, 0x28)
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	DHE-RSA-AES128-SHA256	(0x00, 0x67)
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	DHE-RSA-AES256-SHA256	(0x00, 0x6B)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	ECDHE-RSA-AES128-SHA	(0xC0, 0x13)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	ECDHE-RSA-AES256-SHA	(0xC0, 0x14)
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE-RSA-AES128-SHA	(0x00, 0x33)
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE-RSA-AES256-SHA	(0x00, 0x39)
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	DHE-DSS-AES128-GCM-SHA256	(0x00, 0xA2)
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	DHE-DSS-AES256-GCM-SHA384	(0x00, 0xA3)
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	DHE-DSS-AES128-SHA256	(0x00, 0x40)
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256	DHE-DSS-AES256-SHA256	(0x00, 0x6A)
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE-DSS-AES128-SHA	(0x00, 0x32)
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE-DSS-AES256-SHA	(0x00, 0x38)
TLS_DH_DSS_WITH_AES_128_GCM_SHA256	DH-DSS-AES128-GCM-SHA256	(0x00, 0xA4)
TLS_DH_DSS_WITH_AES_256_GCM_SHA384	DH-DSS-AES256-GCM-SHA384	(0x00, 0xA5)
TLS_DH_DSS_WITH_AES_128_CBC_SHA256	DH-DSS-AES128-SHA256	(0x00, 0x3E)
TLS_DH_DSS_WITH_AES_256_CBC_SHA256	DH-DSS-AES256-SHA256	(0x00, 0x68)
TLS_DH_DSS_WITH_AES_128_CBC_SHA	DH-DSS-AES128-SHA	(0x00, 0x30)
TLS_DH_DSS_WITH_AES_256_CBC_SHA	DH-DSS-AES256-SHA	(0x00, 0x36)
TLS_DH_RSA_WITH_AES_128_GCM_SHA256	DH-RSA-AES128-GCM-SHA256	(0x00, 0xA0)
TLS_DH_RSA_WITH_AES_256_GCM_SHA384	DH-RSA-AES256-GCM-SHA384	(0x00, 0xA1)
TLS_DH_RSA_WITH_AES_128_CBC_SHA256	DH-RSA-AES128-SHA256	(0x00, 0x3F)
TLS_DH_RSA_WITH_AES_256_CBC_SHA256	DH-RSA-AES256-SHA256	(0x00, 0x69)
TLS_DH_RSA_WITH_AES_128_CBC_SHA	DH-RSA-AES128-SHA	(0x00, 0x31)
TLS_DH_RSA_WITH_AES_256_CBC_SHA	DH-RSA-AES256-SHA	(0x00, 0x37)
Note: The following cipher suites, using an ECDH certificate with static parameters, are allowed but not preferred, and, when included at all, should be lower in the preference list than the above listed cipher suites.		
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	ECDH-ECDSA-AES128-GCM-SHA256	(0xC0, 0x2D)
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	ECDH-ECDSA-AES256-GCM-SHA384	(0xC0, 0x2E)
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	ECDH-ECDSA-AES128-SHA256	(0xC0, 0x25)
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	ECDH-ECDSA-AES256-SHA384	(0xC0, 0x26)
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	ECDH-ECDSA-AES128-SHA	(0xC0, 0x04)
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	ECDH-ECDSA-AES256-SHA	(0xC0, 0x05)
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256	ECDH-RSA-AES128-GCM-SHA256	(0xC0, 0x31)
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384	ECDH-RSA-AES256-GCM-SHA384	(0xC0, 0x32)
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	ECDH-RSA-AES128-SHA256	(0xC0, 0x29)
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	ECDH-RSA-AES256-SHA384	(0xC0, 0x2A)
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	ECDH-RSA-AES128-SHA	(0xC0, 0x0E)
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	ECDH-RSA-AES256-SHA	(0xC0, 0x0F)

TLS 1.3		
NIST/IANA	**OpenSSL**	**Hex Code**
TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256	(0x13, 0x01)
TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384	(0x13, 0x02)
TLS_AES_128_CCM_SHA256	TLS_AES_128_CCM_SHA256	(0x13, 0x04)
TLS_AES_128_CCM_8_SHA256	TLS_AES_128_CCM_8_SHA256	(0x13, 0x05)
Note: These suites can be used with either RSA or ECDSA server certificates.		