1. Disabling TLS 1.0 and 1.1, and unsafe algorithms in AS2 and SFTP

In support to provide secure services and solutions to our customers, Nordea will discontinue the support TLS 1.0 and 1.1, and insecure configuration in SFTP and AS2 protocols.

Insecure configuration will be removed in PREPROD environment first so that customers can perform connectivity tests there. Later, by the end of Q3 2023, insecure configuration will be removed in PROD too. Nordea will send newsletter to customers in advance to inform the exact date.

1.1 TLS 1.0 and 1.1 to be removed

Transport Layer Security (TLS) 1.0 and 1.1 are security protocols of HTTPS service for establishing encryption channels over computer networks. File Transfer protocols including AS2, Corporate Access Web Services and EBICs have supported these protocols in the past. However, due to evolving regulatory requirements as well as new security vulnerabilities in TLS 1.0, Nordea requires customers to remove TLS 1.0/1.1 dependencies in customers' software, and Nordea will stop the support of TLS 1.0 and 1.1.

TLS 1.2 and 1.3 are supported by Nordea.

1.2 Insecure AS2 configuration to be removed

Following insecure configurations are going to be removed in AS2 protocol

1. 3DES algorithm for message encryption, which is known to be susceptible to the SWEET32 attack.
2. SHA-1 algorithm for message signing, which is considered weak due to known collision attacks

Customers should use these instead

1. AES-256 algorithm for message encryption
2. SHA-256 algorithm for message signing
3. Nordea also suggests customer to start to have certificate of key length 2048 instead of 1024

1.3 Insecure SFTP configuration to be removed

Following insecure configurations are going to be removed in SFTP protocol

1. Encryption algorithm 3des-ctr, arcfour256, arcfour128 and arcfour which generally considered to be cryptographically weak. And aes256-cbc, 3des-cbc, aes192-cbc, aes128-cbc and blowfish-cbc which is known to be vulnerable to clear-text recovery attacks via cryptanalysis. And chacha20-poly1305@openssh.com which allows remote attackers to bypass integrity checks such that some packets are omitted. (CVE-2023-48795)
2. Key exchange algorithms with usage of SHA1 hash function: rsa1024-sha1, diffie-hellman-group14-sha1, diffie-hellman-group-exchange-sha1 and diffie-hellman-group1-sha1.
3. Authentication algorithms of ssh-rsa and ras-sha1
4. MAC algorithms use weak hashing algorithms SHA1 and MD5: hmac-sha1, hmac-sha1-

etm@openssh.com, hmac-sha1-96, hmac-md5, hmac-md5-etm@openssh.com, hmac-md5-96. And any Encrypt-then-MAC algorithms: hmac-sha2-512-etm@openssh.com, hmac-sha2-256-etm@openssh.com

Customers should use these instead

1. Encryption algorithm: aes256-ctr, aes192-ctr, aes128-ctr, aes256- gcm@openssh.com, aes128-gcm@openssh.com
2. Key exchange algorithms: curve25519-sha256@libssh.org, diffie-hellman-group-exchange-sha256, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, Diffie-hellman-group18-512
3. Authentication algorithms: ED25519 with SHA512 hash function, and RSA with SHA-256/SHA-512 hash function
4. MAC algorithms: hmac-sha2-256, hmac-sha2-512, hmac-ripemd160-etm@openssh.com, umac-128- etm@openssh.com