Encrypt and Decrypt IM&P Compliance Encryption Key

Contents

Introduction

This document describes how to encrypt and decrypt the encryption key generated by IM&P for the compliance encrypted configuration.

Prerequisites

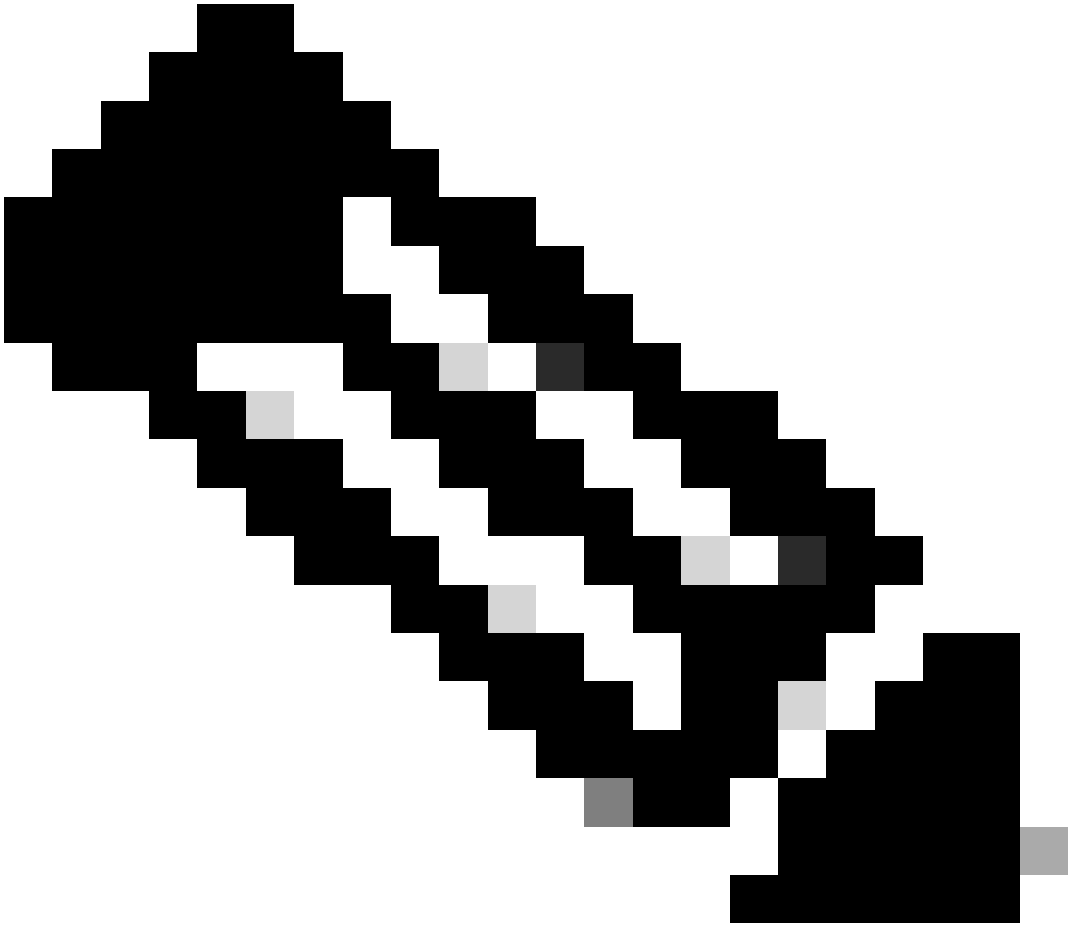
Cisco recommends that you have knowledge of these topics:

- Message Archiver Configuration
- OpenSSL

Components Used

The information in this document is based on these software versions:

- MacOS 15.5
- IM and Presence(IM&P) version 15su2
- OpenSSL 3.3.6



Note: The commands shown in this document can vary based on your OpenSSL version or platform. The Internet is a good source to find those who fit your environment.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

The Message Archiver feature provides a basic IM compliance solution. This feature allows your system to comply with regulations that require logging of all instant messaging traffic in your company. Many industries require that instant messages adhere to the same regulatory compliance guidelines as for all other business records. To comply with these regulations, your system must log and archive all business records, and archived records must be retrievable.

For added security, you can enable an encrypted database for the Message Archiver. When this option is enabled, the IM and Presence Service encrypts IMs before archiving them in the external database. With this option, all data in the database is encrypted and you cannot read archived IMs, unless you possess the encryption key.

The encryption key can be downloaded from the IM and Presence Service and used in conjunction with whatever tool you use to view data in order to decrypt archived data.

Encrypt / Decrypt

1. Open your OpenSSL terminal.
2. Generate private key.

```
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048
```

3. Extract the public key from the private key.

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

4. At this point, we have 2 files private_key.pem and public_key.pem.

- private_key.pem: Used to decrypt the encrypted key from IM&P.
- public_key.pem: This is the key you share with the IM&P server to allow them to encrypt the AES key and IV.

Additionally, the IM&P server adds Base64 encoding to the encrypted encryption key.

5. Download the encryption key from the IM&P server, please refer to the section *Download Encryption Key* in the guide Instant Messaging Compliance Guide for the IM and Presence Service.
6. At this point you have 3 files private_key.pem, public_key.pem and encrypted_key.pem.
7. In this case encrypted_key.pem is Based64-encoded for safe transmission.
8. Decode the Base64-encoded encrypted key.

```
base64 -D -i encrypted_key.pem -o encrypted_key.bin
```

This removes Base64 encoding and produces a 256-byte file that was originally encrypted with your public RSA key.

9. Decrypt the Encrypted Key with your RSA private key.

```
openssl pkeyutl -decrypt -inkey private_key.pem -in encrypted_key.bin -out decryptedkey.bin
```

This decrypts the AES key (K) and IV used for IM&P message encryption.

Example decrypted file:

key = 0ec39f2a22abf63d4452b932f12de

iv = 6683bb3d7e59e82e3fa9f42

10. Decrypt the AES-encrypted messages.

```
openssl enc -aes-256-cbc -d -in encrypted.bin -out decrypted.txt -K <hex_key> -iv <hex_iv>
```

Troubleshoot

A common error when trying to decrypt the encrypted file is:

```
Public Key operation error 60630000:error:0200006C:rsa routines:rsa_ossl_private_decrypt:data greater t
```

This error occurs when you try to RSA-decrypt data that is too large for the size of your RSA private key. RSA can only decrypt data up to the size of its modulus. In our case, a 2048-bit RSA key can only decrypt 256 bytes.

If you check the encrypted key file generated by IM&P, itis 344 bytes. Youcan only decrypt 256 bytes with our private key.

```
-rw-rw-rw-@ 1 testuser staff 344 Jun 5 13:10 encrypted_key.pem
```

As mentioned previously in this document, the encrypted key is Base64 encoded for safe transmission, which adds bytes to the file size.

Once we remove the Base64 encoding, you have a 256-bytes file, easily decryptable with our private key.

```
-rw-r--r-- 1 testuser staff 256 Jun 12 09:16 encrypted_key.bin
```

Security Best Practices

- Store your private key (private_key.pem) securely.
- Do not share your private key with others or upload it to untrusted systems.
- Clean up temporary files like decryptedkey.bin after decryption.