

Forum:[Hack Forums](#)

Thread: [Securely send encrypted data over a network using AES encryption](#)

Clumsy - March 7, 2024, 6:59 pm

The provided code is a C program designed to securely send encrypted data over a network using AES encryption and a telnet connection. Initially, it includes necessary libraries for standard input/output operations, memory allocation and manipulation, AES encryption (through OpenSSL), networking (sockets), and internet operations. Constants such as block size, server IP, port, and maximum data size are defined for global use.

```
Code #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/aes.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <errno.h>

#define BLOCK_SIZE 16
#define SERVER_IP "127.0.0.1"
#define SERVER_PORT 23
#define MAX_DATA_SIZE 1024

void handle_error(const char *message) {
    perror(message);
    exit(EXIT_FAILURE);
}

void encrypt_data(unsigned char *data, int dataLength, unsigned char *key, unsigned char *iv) {
    int i;

    for (i = 0; i < dataLength; i += BLOCK_SIZE) {
        xor_block(data + i, iv, BLOCK_SIZE);
        aes_encrypt_block(data + i, key);
        iv = data + i; // Update IV for the next block
    }
}

void decrypt_data(unsigned char *data, int dataLength, unsigned char *key, unsigned char *iv) {
    int i;
    unsigned char temp[BLOCK_SIZE];

    for (i = 0; i < dataLength; i += BLOCK_SIZE) {
        memcpy(temp, data + i, BLOCK_SIZE);
        aes_decrypt_block(data + i, key);
        xor_block(data + i, iv, BLOCK_SIZE);
        memcpy(iv, temp, BLOCK_SIZE); // Update IV for the next block
    }
}
```

```

    }
}

int connect_telnet_server() {
    int sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock == -1) {
        handle_error("Socket creation failed");
    }

    struct sockaddr_in server;
    server.sin_family = AF_INET;
    server.sin_port = htons(SERVER_PORT);
    server.sin_addr.s_addr = inet_addr(SERVER_IP);

    if (connect(sock, (struct sockaddr *)&server, sizeof(server)) < 0) {
        close(sock);
        handle_error("Connection failed");
    }

    return sock;
}

void send_encrypted_data(int sock, unsigned char *data, int dataLength) {
    if (send(sock, data, dataLength, 0) < 0) {
        close(sock);
        handle_error("Sending data failed");
    }
}

void receive_and_display_response(int sock) {
    unsigned char serverResponse[MAX_DATA_SIZE];
    int bytesReceived = recv(sock, serverResponse, sizeof(serverResponse), 0);

    if (bytesReceived > 0) {
        printf("Received response from server:\n");
        fwrite(serverResponse, 1, bytesReceived, stdout);
    } else if (bytesReceived == 0) {
        printf("Connection closed by server.\n");
    } else {
        close(sock);
        handle_error("Receiving data failed");
    }
}

int main() {
    unsigned char data[MAX_DATA_SIZE];
    unsigned char key[BLOCK_SIZE];
    unsigned char iv[BLOCK_SIZE];

    int dataLength;

    printf("Enter the data to be encrypted: ");
    if (fgets((char *)data, sizeof(data), stdin) == NULL) {

```

```

    handle_error("Failed to read input");
}
dataLength = strlen((char *)data);

printf("Enter the encryption key (16 bytes): ");
if (fread(key, 1, BLOCK_SIZE, stdin) != BLOCK_SIZE) {
    handle_error("Failed to read encryption key");
}

printf("Enter the initialization vector (16 bytes): ");
if (fread(iv, 1, BLOCK_SIZE, stdin) != BLOCK_SIZE) {
    handle_error("Failed to read initialization vector");
}

encrypt_data(data, dataLength, key, iv);

printf("The encrypted data is: ");
fwrite(data, 1, dataLength, stdout);

int telnet_sock = connect_telnet_server();
send_encrypted_data(telnet_sock, data, dataLength);
receive_and_display_response(telnet_sock);

decrypt_data(data, dataLength, key, iv);

printf("\nThe decrypted data is: ");
fwrite(data, 1, dataLength, stdout);

close(telnet_sock);

return 0;
}

```

Hackers - March 7, 2024, 8:23 pm

(Mar 7, 2024 07:59 PM)ClumsyLulz Wrote: [\[CITE\]](#) The provided code is a C program designed to securely send encrypted data over a network using AES encryption and a telnet connection. Initially, it includes necessary libraries for standard input/output operations, memory allocation and manipulation, AES encryption (through OpenSSL), networking (sockets), and internet operations. Constants such as block size, server IP, port, and maximum data size are defined for global use.

```

Code #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <openssl/aes.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <errno.h>

#define BLOCK_SIZE 16
#define SERVER_IP "127.0.0.1"
#define SERVER_PORT 23

```

```
#define MAX_DATA_SIZE 1024
```

```
void handle_error(const char *message) {  
    perror(message);  
    exit(EXIT_FAILURE);  
}
```

```
void encrypt_data(unsigned char *data, int dataLength, unsigned char *key, unsigned char *iv) {  
    int i;  
  
    for (i = 0; i < dataLength; i += BLOCK_SIZE) {  
        xor_block(data + i, iv, BLOCK_SIZE);  
        aes_encrypt_block(data + i, key);  
        iv = data + i; // Update IV for the next block  
    }  
}
```

```
void decrypt_data(unsigned char *data, int dataLength, unsigned char *key, unsigned char *iv) {  
    int i;  
    unsigned char temp[BLOCK_SIZE];  
  
    for (i = 0; i < dataLength; i += BLOCK_SIZE) {  
        memcpy(temp, data + i, BLOCK_SIZE);  
        aes_decrypt_block(data + i, key);  
        xor_block(data + i, iv, BLOCK_SIZE);  
        memcpy(iv, temp, BLOCK_SIZE); // Update IV for the next block  
    }  
}
```

```
int connect_telnet_server() {  
    int sock = socket(AF_INET, SOCK_STREAM, 0);  
    if (sock == -1) {  
        handle_error("Socket creation failed");  
    }  
  
    struct sockaddr_in server;  
    server.sin_family = AF_INET;  
    server.sin_port = htons(SERVER_PORT);  
    server.sin_addr.s_addr = inet_addr(SERVER_IP);  
  
    if (connect(sock, (struct sockaddr *)&server, sizeof(server)) < 0) {  
        close(sock);  
        handle_error("Connection failed");  
    }  
  
    return sock;  
}
```

```
void send_encrypted_data(int sock, unsigned char *data, int dataLength) {  
    if (send(sock, data, dataLength, 0) < 0) {  
        close(sock);  
        handle_error("Sending data failed");  
    }  
}
```

```
}
```

```
void receive_and_display_response(int sock) {
    unsigned char serverResponse[MAX_DATA_SIZE];
    int bytesReceived = recv(sock, serverResponse, sizeof(serverResponse), 0);

    if (bytesReceived > 0) {
        printf("Received response from server:\n");
        fwrite(serverResponse, 1, bytesReceived, stdout);
    } else if (bytesReceived == 0) {
        printf("Connection closed by server.\n");
    } else {
        close(sock);
        handle_error("Receiving data failed");
    }
}
```

```
int main() {
    unsigned char data[MAX_DATA_SIZE];
    unsigned char key[BLOCK_SIZE];
    unsigned char iv[BLOCK_SIZE];

    int dataLength;

    printf("Enter the data to be encrypted: ");
    if (fgets((char *)data, sizeof(data), stdin) == NULL) {
        handle_error("Failed to read input");
    }
    dataLength = strlen((char *)data);

    printf("Enter the encryption key (16 bytes): ");
    if (fread(key, 1, BLOCK_SIZE, stdin) != BLOCK_SIZE) {
        handle_error("Failed to read encryption key");
    }

    printf("Enter the initialization vector (16 bytes): ");
    if (fread(iv, 1, BLOCK_SIZE, stdin) != BLOCK_SIZE) {
        handle_error("Failed to read initialization vector");
    }

    encrypt_data(data, dataLength, key, iv);

    printf("The encrypted data is: ");
    fwrite(data, 1, dataLength, stdout);

    int telnet_sock = connect_telnet_server();
    send_encrypted_data(telnet_sock, data, dataLength);
    receive_and_display_response(telnet_sock);

    decrypt_data(data, dataLength, key, iv);

    printf("\nThe decrypted data is: ");
    fwrite(data, 1, dataLength, stdout);
}
```

```
close(telnet_sock);
```

```
return 0;
```

```
}
```

Nice, didn't look through all code yet but will try your code out soon when I have the time.

Are you using this currently for anything?

Clumsy - March 9, 2024, 4:22 am

(Mar 7, 2024 09:23 PM)nodes Wrote: [\[CITE\]](#) Nice, didn't look through all code yet but will try your code out soon when I have the time.

Are you using this currently for anything?

Currently no lmk if you make something out of using it