Network Security Assignment2

April 1, 2023

Malthe Tøttrup
201907882@post.au.dk
Department of Electrical and Computer Engineering Aarhus University

Introduction

The objective of this lab exercise was to implement an encrypted one-way covert channel using the ICMP protocol. The client/server programs were implemented to exchange encrypted covert messages through the network. The communication was one-way, following the typical use case of covert channels for exfiltration of sensitive data. ICMP protocol was used as an error-reporting protocol, and messages with type 47 were used for transmitting the covert messages. The client program received a destination IP address from the command-line and transmitted the messages, while the server program listened to the network for such messages and printed them in the console as they arrived. For encryption, a preshared symmetric key was used to protect the transmitted payload.

Methodology

To implement the covert channel, Python programming language was used. The following steps were taken to implement the covert channel:

The client program was implemented to receive a destination IP address from the command-line to transmit messages and wait for input from the keyboard at the client-side. The server program was implemented to listen to the network for messages and print them in the console as they arrived. The ICMP messages with type 47 were used for transmitting the covert messages. The messages were encrypted using a preshared symmetric key to protect the transmitted payload. The algorithms and modes of operation were chosen wisely to ensure the security and confidentiality of the transmitted messages.

Encryption Algorithm

To ensure the security and confidentiality of the transmitted messages, the messages were encrypted using the GCM (Galois Counter Mode) encryption mode. GCM is a widely used mode of operation for symmetric encryption that provides both confidentiality and integrity protection. It uses a combination of a block cipher (such as AES) and a Galois field multiplication to provide encryption and authentication of the data.

GCM mode is an authenticated encryption mode that provides both confidentiality and integrity protection. It uses a 128-bit block cipher (such as AES) for encryption, and the authentication tag is also 128 bits long. The mode uses a unique counter value for each block of plaintext, and the counter value is combined with a nonce (a random value used only once) to create a unique initialization vector (IV) for each block. This ensures that each block of ciphertext is unique and prevents an attacker from discovering the key by analyzing patterns in the ciphertext.

In this implementation, GCM mode was chosen because it provides a high level of security and is widely used in industry-standard encryption protocols. The preshared symmetric key was

used as the encryption key for the GCM mode, ensuring that only authorized parties with the key could read the messages.

Results

The implementation of the encrypted covert channel using the ICMP protocol was successful. The client program transmitted the messages to the server program through the network. The server program listened to the network for messages and printed them in the console as they arrived. The messages were encrypted using a preshared symmetric key to protect the transmitted payload, ensuring the security and confidentiality of the transmitted messages.

Conclusion

In conclusion, the implementation of an encrypted covert channel using the ICMP protocol was successful. The client/server programs were implemented to exchange encrypted covert messages through the network. The communication was one-way, and the ICMP protocol was used as an error-reporting protocol. The messages were encrypted using a preshared symmetric key to protect the transmitted payload. The implementation of such a covert channel can be useful in situations where sensitive data needs to be exfiltrated without detection.

How to run the code

The code for the project can be found on GitHub. The client and server programs are located in the /assignment2/task1 directory along with the ICMP header implementation. The code uses raw sockets meaning that you will need root privileges to run the client and server. This also means that the dependencies will have to be installed as root.

Installing the dependencies:

```
sudo pip install pycryptodome
```

Enter the directory:

```
cd netsec_ws/assignment2/task1
```

Run server as root:

```
sudo python3 server.py
```

Open a new terminal and run the client as root:

```
sudo python3 client.py
```