Master 1 Cryptis - Computer Science
Faculty of Science and Technology

Network Audit and Security

TP Project - IP communication using secure hidden channel

Authors :
Minh Luan NGUYEN
Thi Ha Trang NGUYEN

Academic year 2023 - 2024

Contents

1 Introduction

1.1 General

This report contains analysis, understanding, and implementation of the TP Project *Hidden channel*. We also show how to create a tool that enables communication between two machines over the Internet through a "hidden channel," which remains undetectable by standard monitoring tools that typically intercept IP packets.

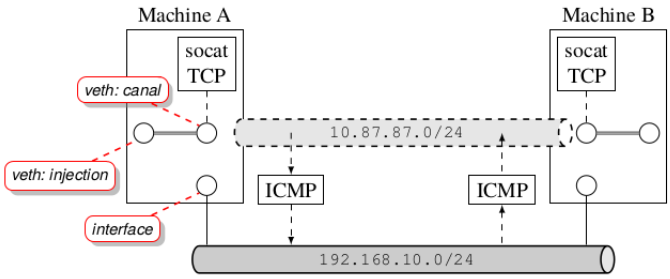

Figure 1: Covert channel example

The specific objectives of this project:

- The creation of a tunnel-like hidden channel for secure communication.
- The use of obfuscation techniques, and encryption using to protect datagram contents.
- The selection of the ICMP protocol as the carrier for the hidden channel.
- The implementation with Scapy for packet interception, encapsulation, encryption, and transmission, as well as for receiving and extracting hidden packets on the receiving end.

1.2 WhisperNet - A Covert Channel Tool

WhisperNet is a tool written in Python, designed for covert communication using the ICMP protocol. This tool can set up virtual Ethernet (veth) interfaces, capture network packets, obfuscate and forward them, and even provide a chat mode using UDP or TCP protocols.

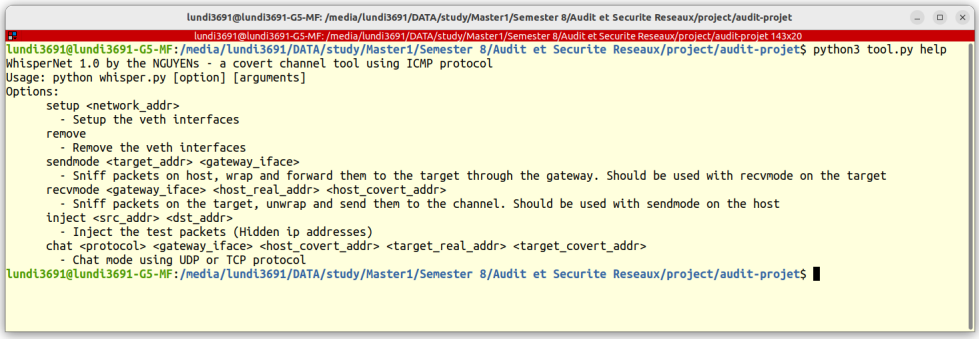

Figure 2: WhisperNet tool

1.3 Infastructure

In this project, we will use the *netlab* configuration provided by Mr. Bonnefoi. We will attempt to send packets from *h1* to *h2*.

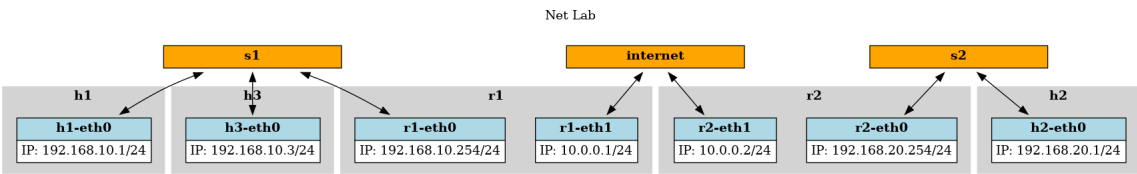

Figure 3: Covert channel example

2 Implementation

We can use the *help* option to show all the possible options WhisperNet offers.

```
python tool.py help
```

2.1 Setup

To be able to communicate with each other through the covert channel, the sender and the recipient must first establish the covert interfaces. We will use Whispernet to set up the interface on both machines. The provided address is the covert channel address for each machine.

```
[h1]
python tool.py setup 10.87.87.1/24

[h2]
python tool.py setup 10.87.87.2/24
```

```
lundi3691@lundi3691-G5-MF: /media/lundi3691/DATA/study/Master1/Semester 8/Audit et Securite Reseaux/projec...

lundi3691@lundi3691-G5-MF: /media/lundi3691/DATA/study/Master
dy/Master1/Semester 8/Audit et Securite Reseaux/pro
ject/audit-projet$ [h1] sudo python3 tool.py setup
10.87.87.1/24
[sudo] password for lundi3691:
Setting up the tool...
2: channel@injection: <BROADCAST,MULTICAST,UP,LOWER
_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 7e:ca:a9:2b:fe:f4 brd ff:ff:ff:ff:ff
:ff
    inet 10.87.87.1/24 scope global channel
       valid_lft forever preferred_lft forever
    inet6 fe80::7cca:a9ff:fe2b:fef4/64 scope link t
entative
       valid_lft forever preferred_lft forever
lundi3691@lundi3691-G5-MF:/media/lundi3691/DATA/stu
dy/Master1/Semester 8/Audit et Securite Reseaux/pro
ject/audit-projet$ [h1] ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noque
ue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00
:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: channel@injection: <BROADCAST,MULTICAST,UP,LOWER
_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 7e:ca:a9:2b:fe:f4 brd ff:ff:ff:ff:ff
:ff
    inet 10.87.87.1/24 scope global channel
       valid_lft forever preferred_lft forever
    inet6 fe80::7cca:a9ff:fe2b:fef4/64 scope link
       valid_lft forever preferred_lft forever
3: injection@channel: <BROADCAST,MULTICAST,UP,LOWER
_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 46:05:93:99:6c:e4 brd ff:ff:ff:ff:ff
:ff
    inet6 fe80::4405:93ff:fe99:6ce4/64 scope link
       valid_lft forever preferred_lft forever
33: h1-eth0@if32: <BROADCAST,MULTICAST,UP,LOWER_UP>
 mtu 1500 qdisc noqueue state UP group default qlen
 1000
    link/ether a2:11:17:5f:d3:25 brd ff:ff:ff:ff:ff
:ff link-netnsid 0
    inet 192.168.10.1/24 scope global h1-eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a011:17ff:fe5f:d325/64 scope link
       valid_lft forever preferred_lft forever
lundi3691@lundi3691-G5-MF:/media/lundi3691/DATA/stu
dy/Master1/Semester 8/Audit et Securite Reseaux/pro
ject/audit-projet$ [h1]
```

```
lundi3691@lundi3691-G5-MF: /media/lundi3691/DATA/study/Maste
dy/Master1/Semester 8/Audit et Securite Reseaux/pro
ject/audit-projet$ [h2] sudo python3 tool.py setup
10.87.87.2/24
[sudo] password for lundi3691:
Setting up the tool...
2: channel@injection: <BROADCAST,MULTICAST,UP,LOWER
_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 56:06:de:ce:50:15 brd ff:ff:ff:ff:ff
:ff
    inet 10.87.87.2/24 scope global channel
       valid_lft forever preferred_lft forever
    inet6 fe80::5406:deff:fece:5015/64 scope link t
entative
       valid_lft forever preferred_lft forever
lundi3691@lundi3691-G5-MF:/media/lundi3691/DATA/stu
dy/Master1/Semester 8/Audit et Securite Reseaux/pro
ject/audit-projet$ [h2] ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noque
ue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00
:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: channel@injection: <BROADCAST,MULTICAST,UP,LOWER
_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 56:06:de:ce:50:15 brd ff:ff:ff:ff:ff
:ff
    inet 10.87.87.2/24 scope global channel
       valid_lft forever preferred_lft forever
    inet6 fe80::5406:deff:fece:5015/64 scope link
       valid_lft forever preferred_lft forever
3: injection@channel: <BROADCAST,MULTICAST,UP,LOWER
_UP> mtu 1500 qdisc noqueue state UP group default
qlen 1000
    link/ether 1a:1f:e4:79:11:a9 brd ff:ff:ff:ff:ff
:ff
    inet6 fe80::181f:e4ff:fe79:11a9/64 scope link
       valid_lft forever preferred_lft forever
35: h2-eth0@if34: <BROADCAST,MULTICAST,UP,LOWER_UP>
 mtu 1500 qdisc noqueue state UP group default qlen
 1000
    link/ether 4a:d7:1b:a6:12:13 brd ff:ff:ff:ff:ff
:ff link-netnsid 0
    inet 192.168.20.1/24 scope global h2-eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::48d7:1bff:fea6:1213/64 scope link
       valid_lft forever preferred_lft forever
lundi3691@lundi3691-G5-MF:/media/lundi3691/DATA/stu
dy/Master1/Semester 8/Audit et Securite Reseaux/pro
ject/audit-projet$ [h2]
```

Figure 4: Running setup option and results

Note: We might need to use *sudo* because some commands required administrator's permission.

2.2 Send mode

To make the data undetected while sending through the network, our hidden packets need to be obfuscated or encrypted and then wrapped around by an ICMP packet to be sent.

On the sender machine (*h1*), run WhisperNet in *sendmode*. This will make WhisperNet listen for any outbound packet from the *channel* interface, it will then obfuscate/encrypt and put the

data into an ICMP echo request packet, with the IP header containing the "real" addresses of the sender (as *src*) and the recipient (as *dst*, will be provided by the user). This ICMP packet will then be sent through the gateway interface.

```
[h1] python tool.py sendmode 1 192.168.20.1 h1-eth0
```

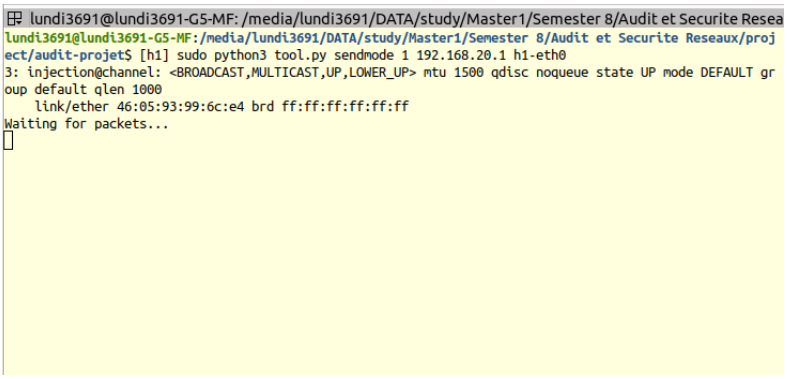

Figure 5: WhisperNet on *sendmode* options

2.3 Receive mode

As for the recipient, we use WhisperNet in *recvmode*. It will monitor every packet that goes through the specified interface for an ICMP packet with the sender's physical IP address as the *src*. If the packet is intercepted, the tool will deobfuscate/decrypt the hidden data and reclaim the packet. The packet will be sent to the *channel* interface.

```
[h2] python tool.py recvmode 1 h2-eth0 192.168.10.1 10.87.87.2
```

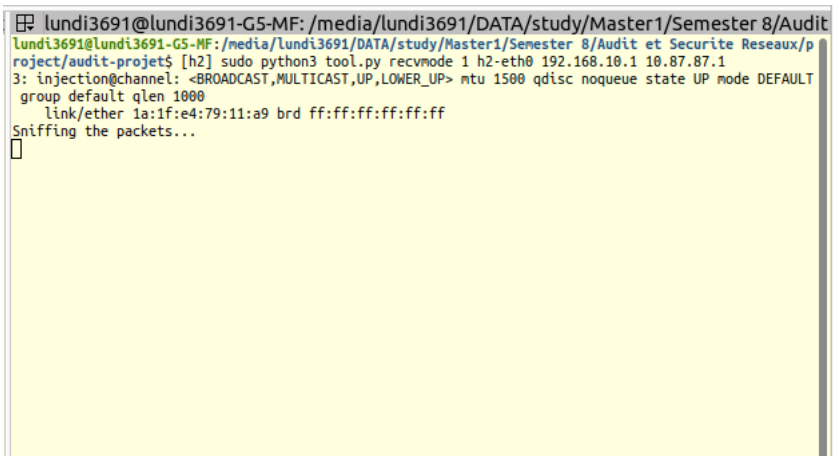

Figure 6: WhisperNet on *recvmode* option

2.4 Inject packet

To test how the packet will go through the covert channel, we can use the *inject* option. The option was pre-implemented with a test UDP and a test TCP packet. We just need to specify the hidden source and destination address for the packet to be sent.

```
[h1] python tool.py inject 10.87.87.1 10.87.87.2
```

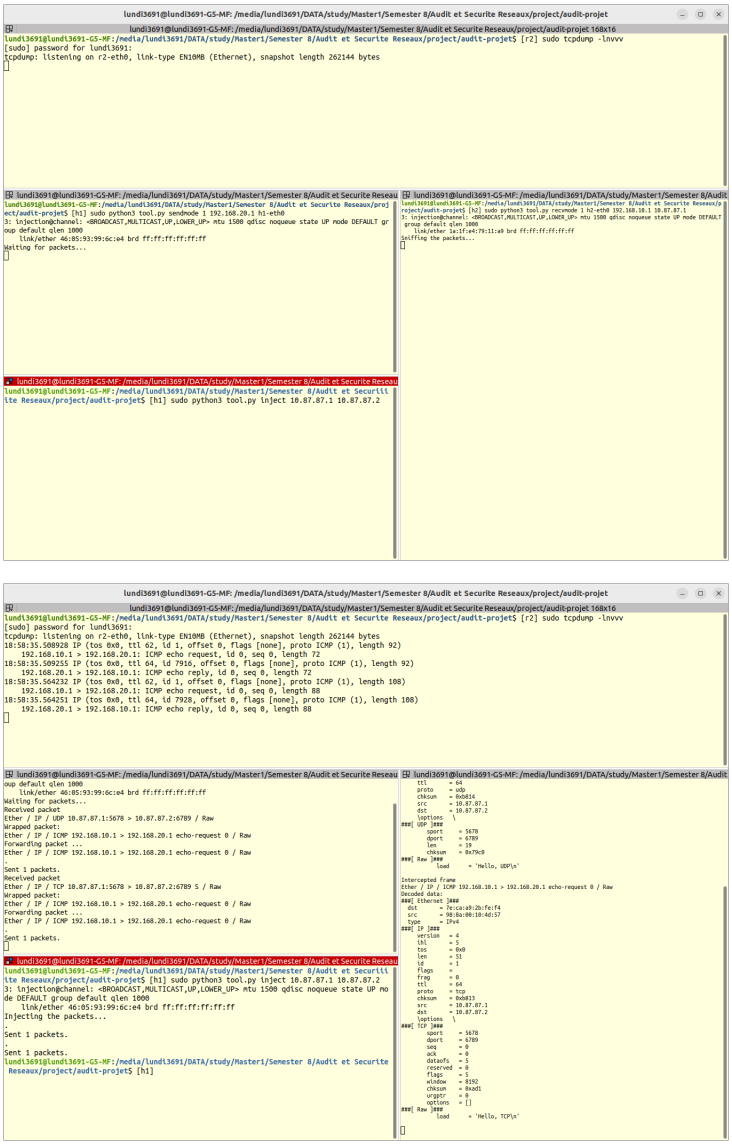

Figure 7: Injection operation

As shown in the result, *h2* received two hidden packets: one UDP and one TCP, and they

are the WhisperNet's original packets:

```
udp_frame = Ether(src=RandMAC(), dst=mac_addr) / IP(src=src_addr, dst=dst_addr) /
↪  UDP(sport=5678, dport=6789) / "Hello, UDP\n"
tcp_frame = Ether(src=RandMAC(), dst=mac_addr) / IP(src=src_addr, dst=dst_addr) /
↪  TCP(flags="S", sport=5678, dport=6789) / "Hello, TCP\n"
```

2.5 Protected data

WhisperNet offers two ways to secure data, which can be configured in different options:

- Obfuscation: XORing the data with a sequence of 1s with the same length.
- Cryptography: Encrypts data using AES encryption.

This prevents the data from eavesdropping by third parties. If we look into the ICMP packet, we can see the hidden data:

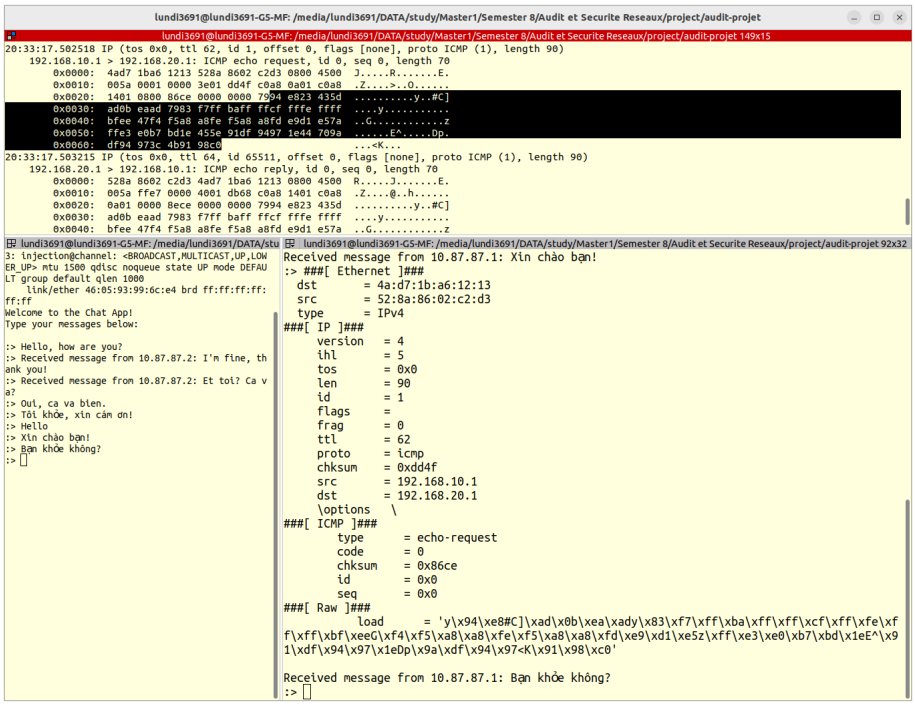

Figure 8: Hidden data in ICMP packet

2.6 Chat mode

With the following procedures, we can utilize this covert channel to send hidden messages between two clients. We have integrated them into each thread so they won't interfere with one another but instead, work together to send messages continuously like a chat application but the messages are hidden using a covert channel.

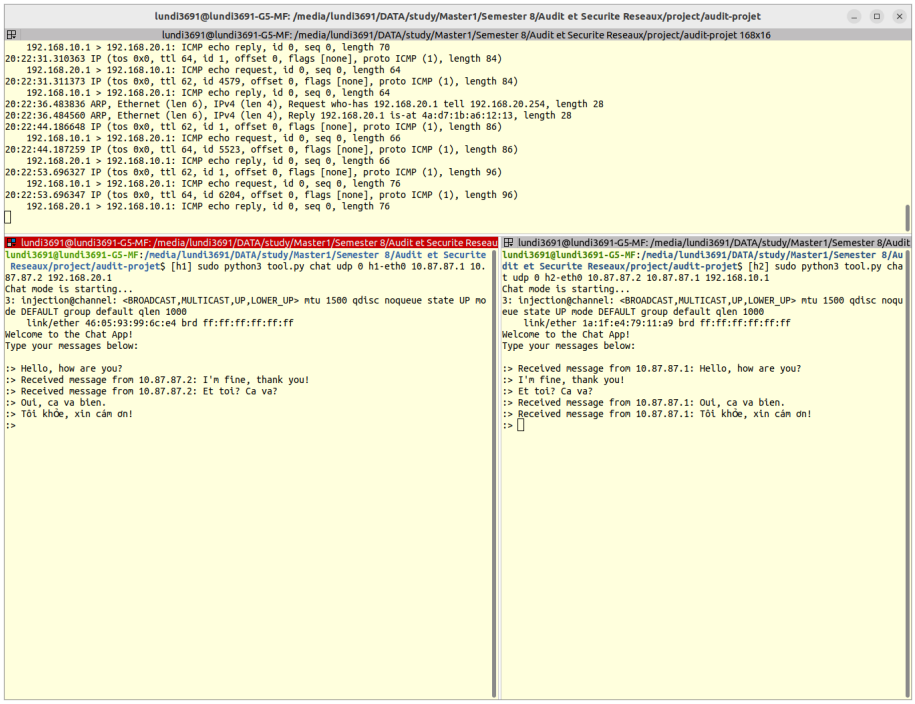

Figure 9: Chat mode

2.7 Clean up

After we have finished sending packets, we can remove the interfaces by using:

```
python tool.py remove
```