Ncat Chat Terminal Project Report – Assignment 1

Name: Anamika Manoj **Reg. No:** 2463006 **Course:** Honors Cybersecurity

---

🔧 Methodology

To simulate a basic terminal-based chat system between two computers, I used `ncat`, a tool from the Nmap suite that facilitates raw TCP and UDP connections. The task was performed in a Kali Linux environment. Two shell scripts were created:

- One to act as a **listener** (Terminal A)
- One to act as a **sender** (Terminal B)

Communication was done over TCP using **port 5000**.

⚙️ Installation & Setup

Before starting, ensure `ncat` is installed. It usually comes pre-installed with Kali Linux, but if not, install it using:

```
sudo apt update
sudo apt install ncat
```

```
ncat --version
```

💻 Code

Open terminal and type:

nano listener.sh

nano sender.sh

listener.sh

```
#!/bin/bash
echo "Starting listener on port 5000..."
ncat -lvp 5000
```

sender.sh

```
#!/bin/bash
echo "Connecting to chat..."
ncat 127.0.0.1 5000
```

Make Scripts Executable

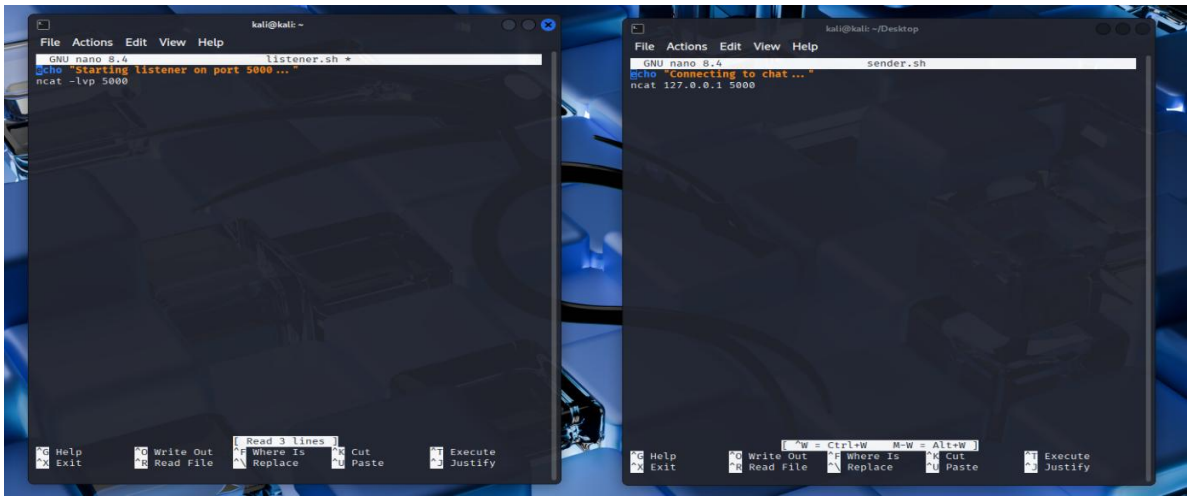
```
chmod +x listener.sh sender.sh
```

Execution Steps

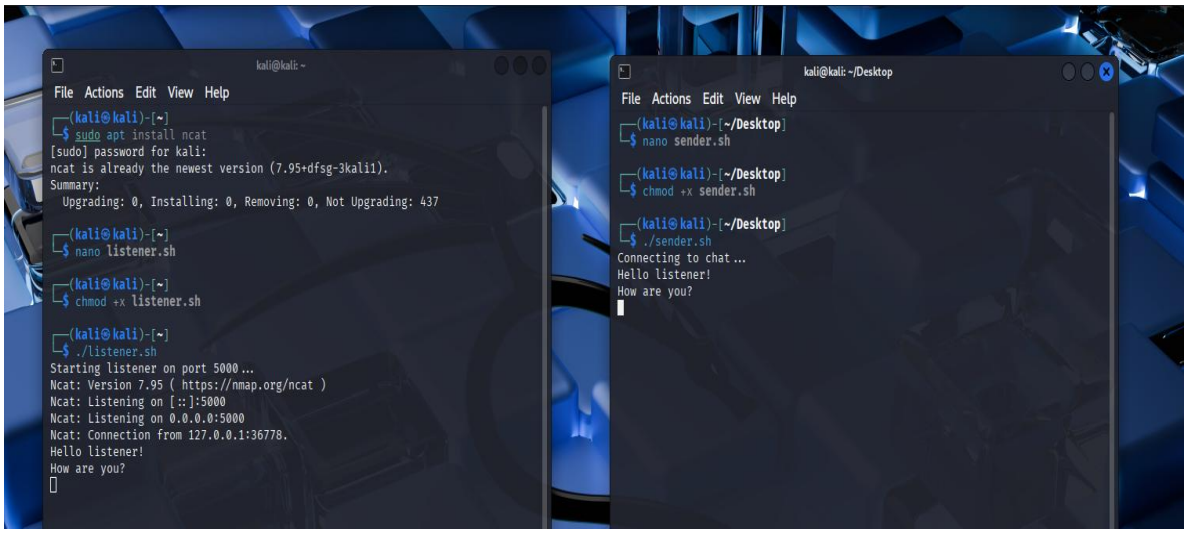
1. Open **Terminal A** and run `./listener.sh`
2. Open **Terminal B** and run `./sender.sh`
3. Type messages in either terminal to simulate chat

Screenshots

- Listener and sender terminal on ncat.



- Sender and listener terminal connected and two-way message flow demonstrated



Findings

- **Communication:** Two-way text messages in real-time
- **Protocol Used:** TCP via port 5000
- **Localhost Setup:** 127.0.0.1 (loopback IP), can switch to LAN IP
- **Setup Time:** Less than 5 minutes
- **Learning Outcome:** Demonstrates ports and sockets usage in real-world scenarios

---

Security Insight

- **Port Exposure:** Listener opens TCP port 5000, which can be targeted
- **Risk:** If the terminal is left unattended, it can be hijacked or misused
- **Relevance in Cybersecurity:** Common method in red teaming, used to open backdoors or initiate reverse shells

---

Conclusion

This assignment provided a hands-on understanding of TCP-based communication using `ncat`. With only 10 lines of shell scripting and two terminal sessions, I simulated a working chat system that demonstrates key cybersecurity and networking concepts. This experiment builds foundational knowledge useful in penetration testing, network diagnostics, and defensive monitoring.
