Nmap (network mapper) is a powerful network scanning tool used by cyber security professionals to locate vulnerability in an environment.

```
const handleScan = async () => {
  setLoading(true);
  setScanResults([]); // Clear old results

  try {
    const response = await fetch('http://192.168.1.72:5000/scan'); // Fetch from backend
    const data = await response.json();

    // Parse raw nmap output into a structured array
    const parsedResults = parseNmapOutput(data.output);
    setScanResults(parsedResults);
  } catch (error) {
    console.error('Error:', error);
    setScanResults([{ ip: 'Error', mac: 'Failed to scan', vendor: '' }]);
  }

  setLoading(false);
};
```

React Native Frontend:

- Sends a fetch request to a backend server (currently running on localhost).
- Parses the nmap -sn (ping scan) output into a structured format.

```
const express = require('express');
const cors = require('cors');
const { NodeSSH } = require('node-ssh');

const app = express();
const ssh = new NodeSSH();

app.use(cors());

app.get('/scan', async (req, res) => {
    try {
        await ssh.connect({
            host: '192.168.1.89',  //Kali VM IP
            username: 'kali',
            password: 'kali'
        });

        const result = await ssh.execCommand('nmap -sn 192.168.1.0/24');
        res.json({ output: result.stdout });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});

const PORT = 5000;
app.listen(PORT, () => console.log(`SSH API running on port ${PORT}`));
```

Node.js Backend:

- Runs on a separate server (Windows) and uses node-ssh to connect to a Kali Linux VM in VirtualBox.
- Executes nmap scans via SSH and returns results to the React Native app.

Since React Native runs on JavaScript, keeping both frontend (React Native) and backend (Node.js) in JavaScript allows seamless integration.

Challenge	Solution
React Native has no built-in robust network scanning	Used Kali Linux + nmap via SSH for accurate scanning.
node-ssh does not work directly in React Native	Created a Node.js backend to handle SSH and scanning requests.
VirtualBox Kali Linux defaults to penetration testing mode (isolated networking)	Cloned the VM and switched the network adapter to Bridged Mode with Promiscuous Mode enabled to allow SSH via IP.
Windows Defender blocked connections on port 5000	Disabled the firewall temporarily for the demo (controlled network, low risk).

Application:

Penetration testing & War Driving:

- Carrying a laptop for on-site recon or war driving is suspicious.
- Running scans from a phone minimizes exposure and makes penetration testing less detectable.

Network Monitoring:

- Security professionals can use this app to quickly check for unknown or unauthorized devices.