**🧾 Cybersecurity Portfolio**

**Perform a Decoy Scan Using Zenmap (Nmap)**

**🔍 Objective**

To perform a stealth Nmap scan against a target system (192.168.0.10) using decoy IP addresses, to evaluate how well the scan avoids detection by intrusion detection systems (IDS). The goal is to mask the attacker's identity by blending in with other IP sources.

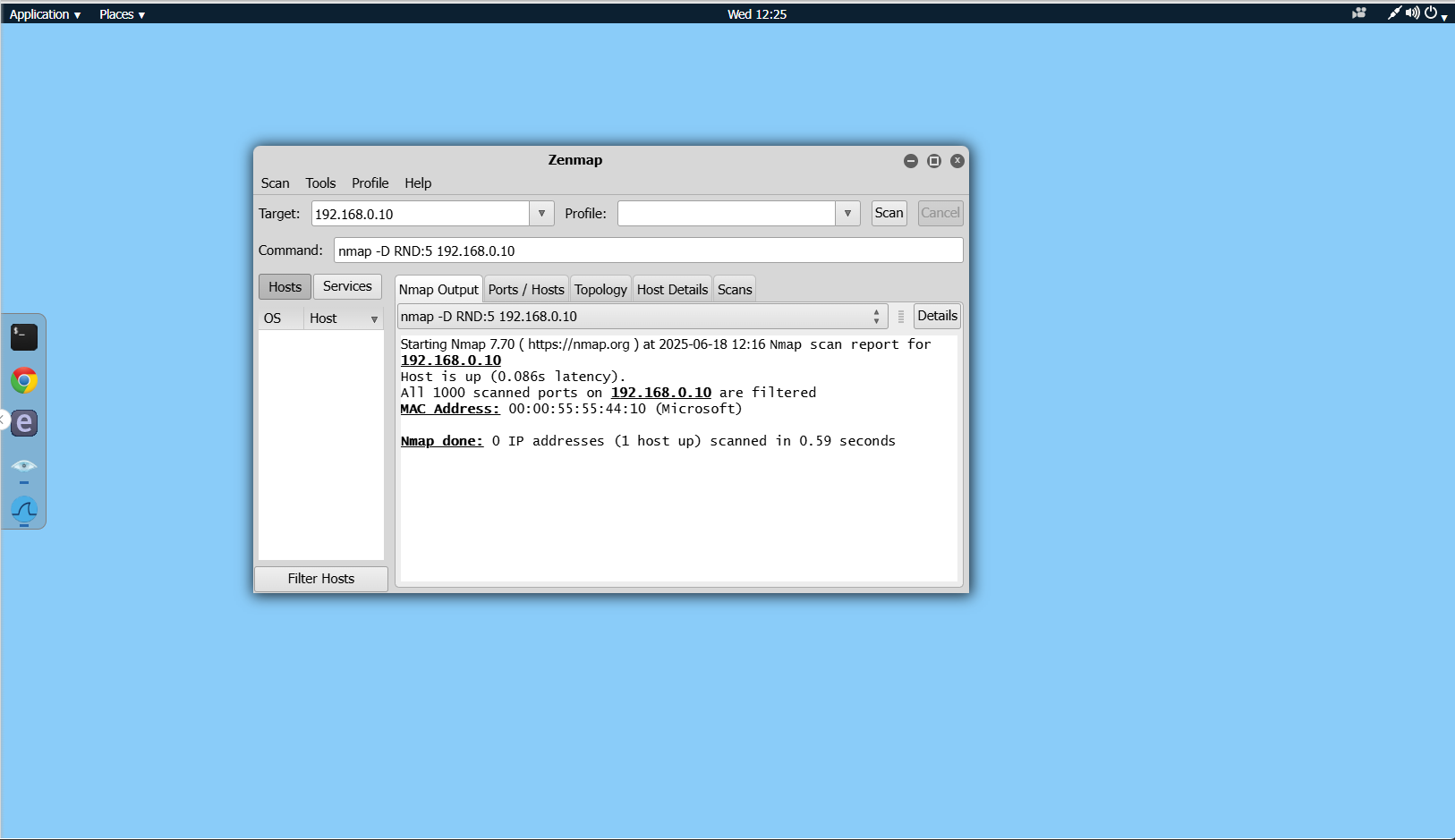
**🛠️ Tools Used**

* Kali Linux
* Zenmap
* Wireshark
* Network Interface: enp2s0
* Target System: 192.168.0.10

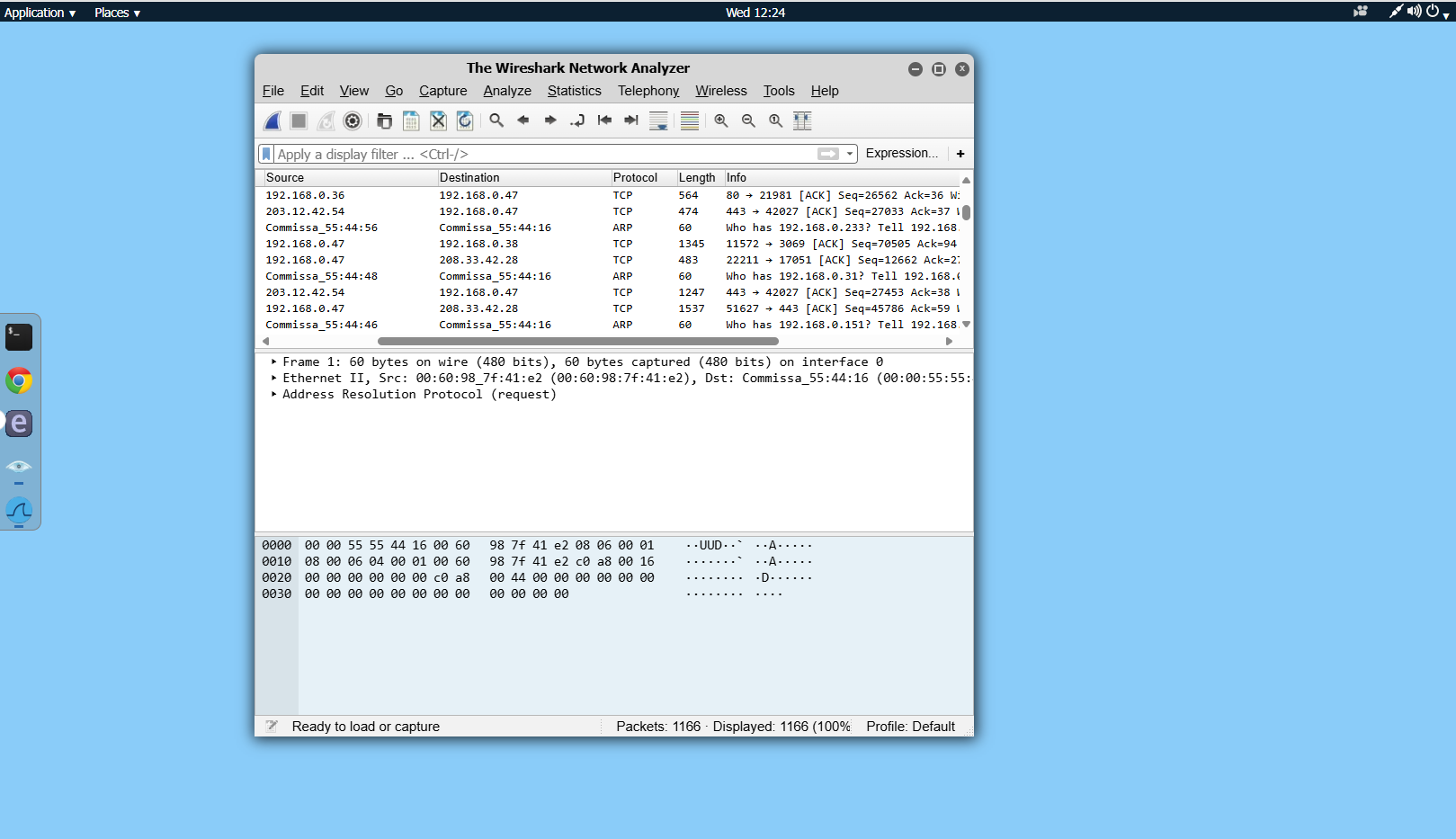
**📜 Steps Taken**

1. **Launched Wireshark:**
   * Started a packet capture on the enp2s0 interface.
   * Began capturing before running the scan to ensure full coverage.
2. **Ran the Decoy Scan Using Nmap:**
   * Used Zenmap in Kali and ran the following command: **nmap -D RND:5 192.168.0.10**

This command instructed Nmap to generate traffic as if 5 random decoy IPs (plus the real one) were scanning the target.



1. **Stopped Wireshark After Scan Completion:**
   * Saved the packet capture for analysis.
2. **Analyzed the Capture in Wireshark:**
   * Observed multiple IPs sending SYN packets to the target.
   * Identified the attacker’s real IP by finding the only one that:
     + Sent a **SYN**
     + Received a **SYN-ACK** in return from 192.168.0.10
     + Sent a **RST** back to end the connection



**✅ Result**

The scan appeared to originate from 6 different IP addresses (5 decoys + attacker), successfully obfuscating the source of the scan. In the Wireshark capture, the attacker’s IP address was identified as the **only** one that responded to a SYN-ACK from the target with an RST packet. This is because decoy IPs don’t actually complete the handshake — only the true scanner does.

**💡 Reflections / Lessons Learned**

This lab demonstrated how attackers can use Nmap’s decoy option to confuse network defenders and avoid straightforward attribution. It also showed that careful traffic analysis using Wireshark can still reveal the true source of a scan by examining TCP handshake behavior.