Final Report on

ICMP Ping Spoofing and ICMP Redirect Attacks

CSE 406 — Computer Security Sessional

Lab Group B2 – Group 08
Mohammad Ishrak Adit (2005105)
Shahad Shahriar Rahman (2005092)

July 30, 2025

Contents

1 Introduction

This report presents the final implementation and demonstration results for two ICMP-based network attacks:

- ICMP Ping Spoofing Attack
- ICMP Redirect Attack

We implemented both the attacks in dockerized environments and the icmp redirecting attack in Oracle VirtualBox VMs with a better result, experimenting with packet crafting tools such as `scapy` and raw sockets. The goal was to demonstrate traffic manipulation and redirection using unauthenticated ICMP packets. This report outlines the steps, success criteria, platform observations, and countermeasures.

2 Attack Implementation Steps

2.1 ICMP Ping Spoofing (Dockerized)

- Three containers were used: **Attacker**, **Victim**, and **Receiver**.
- The attacker sent ICMP Echo Requests to Receiver with a **spoofed source IP** of the Victim.
- The Receiver replied to the spoofed IP address, proving that the attack successfully diverted replies to an unintended host.
- The Victim received unsolicited ICMP Echo Replies, despite never initiating any ping, validating the spoofing.

2.2 ICMP Redirect Attack (Dockerized)

- Four nodes were involved: **Attacker**, **Victim**, **Router**, and **Target**.
- Four dockerized containers were created.
- The Attacker sent forged ICMP Redirect packets (Type 5) to the Victim, claiming that traffic to target should route through the Attacker.
- Although the Victim received the redirect, **it did not automatically update its routing table** due to kernel restrictions in the containerized environment.

2.3 ICMP Redirect Attack (Oracle VirtualBox)

- This setup involved three full VMs: **Attacker**, **Victim**, and **Router**.
- The Victim attempted to ping an external IP (8.8.8.8) through the legitimate Router.
- The Attacker crafted raw ICMP Redirect packets (using python) spoofing the Router's IP.

- The Victim **accepted the redirect and updated its route automatically**, forwarding subsequent packets through the Attacker.
- The Attacker could then intercept, forward, or drop traffic — achieving full redirection.

3 Success Analysis

- **Ping Spoofing (Dockerized)** — *Successful.*
  - The attacker's spoofed ping triggered replies to a third host.
  - The victim unknowingly received replies for traffic it never initiated.
- **Redirect (Dockerized)** — *Partially Successful.*
  - Victim received the redirect.
  - But kernel-level security ignored the packet, and routing table remained unchanged.
  - Manual routing confirmed attack viability.
- **Redirect (VirtualBox)** — *Successful.*
  - Victim honored the forged redirect.
  - Traffic was transparently hijacked via the Attacker.

4 Observed Output Across Hosts

4.1 ICMP Ping Spoofing

4.1.1 Attacker

- Generated ICMP Echo Requests with the Victim's IP spoofed as the source.
- Never received replies, as they were routed to the Victim.

4.1.2 Receiver

- Received spoofed ICMP Echo Requests.
- Replied to the Victim's IP, as seen in a normal ping.

4.1.3 Victim

- Received ICMP Echo Replies from the Receiver without ever sending any pings.
- This confirmed that the spoofed ping had diverted replies toward it.

4.2 ICMP Redirect Attack

4.2.1 Attacker

- Crafted and sent ICMP Type 5 Redirect packets impersonating the Router.
- Intercepted traffic from the Victim after a successful redirect (VM case).

4.2.2 Victim

- Docker: Received the redirect but ignored it due to kernel configuration.
- VM: Accepted the redirect and updated routing table to route via Attacker.

4.2.3 Router / Target

- Router was spoofed and remained idle after the redirect.
- Target (8.8.8.8) received pings routed through the legitimate router or attacker depending on redirect success.

5 Defense and Countermeasures

- **Disable ICMP Redirects** on all hosts and routers:
  - `net.ipv4.conf.all.accept_redirects = 0`
  - `net.ipv4.conf.all.send_redirects = 0`
- **Implement anti-spoofing filters** using:
  - `iptables` rules (e.g., '–spoof-protect')
  - BCP 38 configuration on routers
- **Use encrypted protocols** like HTTPS, SSH to prevent eavesdropping.
- **Monitor ICMP behavior** via intrusion detection systems (IDS) that flag redirect or abnormal echo patterns.
- **Harden containers or VMs** by disabling unused network features and enforcing strict kernel policies.

6 Conclusion

This project demonstrates the practicality and risks of ICMP-based attacks in both modern and legacy systems. While newer platforms (e.g., Docker) enforce some restrictions by default, legacy systems (e.g., bare-metal VMs) remain vulnerable to ICMP Redirects and spoofed packets.

ICMP Ping Spoofing was consistently successful, showing how easily identity can be spoofed. ICMP Redirect, though partially blocked in containers, proved fully exploitable in VirtualBox environments. These experiments highlight the need for better kernel configurations, stricter firewall rules, and protocol-level security awareness.