ARP-Scan Troubleshooting & Analysis Training Guide

Author: Easton Childress **Field:** Cybersecurity / Network Analysis **Format:** Professional Training Document **Version:** 1.0

Table of Contents

1. Introduction

This training guide provides a complete, professional workflow for diagnosing and resolving ARP-Scan vendor-file issues in Linux environments. ARP-Scan is a widely used tool for network discovery, but its accuracy depends heavily on the integrity of its vendor (OUI) files. When these files are malformed, duplicated, or shadowed by hidden versions, ARP-Scan may produce incorrect vendor names or fail to resolve them entirely.

This guide blends academic clarity with real operator methodology. It documents the full troubleshooting process, explains the underlying concepts, and provides hands-on exercises and red-team applications.

2. Understanding ARP & OUI Resolution

ARP (Address Resolution Protocol) maps IP addresses to MAC addresses on a local network. ARP-Scan uses this mechanism to identify active hosts and then resolves MAC prefixes to vendor names using OUI files.

Key Concepts

- ARP requests ask: *“Who has this IP address?”*
- The responding host returns its MAC address.
- The first three bytes of a MAC address identify the manufacturer (OUI).
- ARP-Scan uses vendor files to map these prefixes to vendor names.

Why This Matters

Accurate vendor resolution helps analysts:

- Identify device types
- Detect rogue or unauthorized devices
- Understand network composition
- Improve reconnaissance accuracy

3. Vendor File Architecture

ARP-Scan searches for vendor files in a specific order. Understanding this hierarchy is essential for diagnosing issues.

Search Order

1. Explicit --ouifile argument
2. Current working directory
3. /etc/arp-scan/
4. /usr/local/share/arp-scan/
5. /usr/share/arp-scan/
6. Built-in fallback table

Risk Levels

Location	Purpose	Risk
--ouifile	User override	High
Current directory	Local override	Medium
/etc/arp-scan/	System config	Low
/usr/local/share/arp-scan/	Custom installs	Medium
/usr/share/arp-scan/	Default vendor file	Low
Fallback	Compiled-in table	Minimal

Hidden or malformed files in high-priority locations often cause the most issues.

4. Common Failure Modes

During this project, multiple failure modes were reproduced and documented:

Examples

- Malformed OUI entries (missing tabs, incorrect spacing)
- CRLF vs LF line-ending conflicts
- Duplicate OUI prefixes causing ambiguity

- Extended OUI formats (MA-S, MA-M) not handled correctly
- Hidden vendor files in unexpected directories
- Incorrect fallback behavior

Each failure mode affects ARP-Scan's ability to resolve vendor names accurately.

5. Diagnostic Workflow

This workflow blends academic structure with real operator troubleshooting.

Step-By-Step Process

1. Validate ARP-Scan installation and interface selection
2. Enumerate all vendor file paths
3. Use strace to trace file access
4. Identify hidden or shadowed vendor files
5. Validate formatting (tabs, LF, no BOM)
6. Rebuild vendor directories from trusted sources
7. Run clean scans to confirm resolution

Example Command

Code

```
sudo strace -f -e openat arp-scan --interface=eth0 192.168.1.0/24
```

This reveals exactly which vendor files ARP-Scan is reading.

6. Case Study: Hidden mac-vendor.txt

Symptoms

- ARP-Scan produced repeated “Could not parse OUI” warnings
- Vendor names were missing or incorrect

Investigation

Using strace, a hidden file was discovered at:

```
Code:  /home/kali/arp-scan/mac-vendor.txt
```

Root Cause

The file contained malformed OUI entries with incorrect spacing and CRLF line endings.

Resolution

- Removed the hidden file
- Replaced it with the official IEEE OUI file

- Validated formatting
- Re-ran ARP-Scan to confirm clean output

This case study demonstrates the importance of understanding ARP-Scan’s vendor file search order.

7. Remediation Procedures

1. Remove corrupted vendor files

Code:
```
sudo rm -r /usr/local/share/oui
```

2. Rebuild clean vendor directory

Code:
```
sudo mkdir -p /usr/local/share/oui
```

3. Download trusted vendor file

Code:
```
sudo wget -O /usr/local/share/oui/oui.txt
\https://raw.githubusercontent.com/royhills/arp-scan/master/ieee-oui.txt
```

4. Validate formatting

- Use tabs, not spaces
- Use LF line endings
- No BOM or CRLF

5. Run clean scan

Confirm that vendor names resolve correctly.

8. Expected Clean Output

Example of correct ARP-Scan output:

Code:
```
192.168.1.10  AA:BB:CC:DD:EE:FF  Dell Inc.

192.168.1.11  11:22:33:44:55:66  Apple Inc.
```

No warnings. Accurate vendor resolution.

9. Hands-On Exercises

Exercise 1: Identify malformed OUI entries

Exercise 2: Use strace to trace ARP-Scan file access

Exercise 3: Rebuild vendor directory from scratch

Exercise 4: Validate vendor file formatting

Exercise 5: Perform a clean scan and document results

These exercises reinforce both academic understanding and operator-grade troubleshooting.

10. Red-Team Application Scenarios

ARP-Scan is valuable for offensive and defensive operations.

Use Cases

- Network footprinting
- Identifying device types quickly
- Detecting rogue or unauthorized devices
- Mapping IoT ecosystems and vendor distribution

11. Troubleshooting Checklist

- Validate ARP-Scan command and interface
- Check vendor file search order
- Use strace to detect hidden files
- Remove malformed or duplicate entries
- Download official IEEE OUI file
- Validate formatting (tabs, LF)
- Re-run ARP-Scan and verify output

12. Conclusion

This guide provides a complete, professional workflow for diagnosing ARP-Scan vendor-file issues. By understanding ARP fundamentals, vendor file architecture, and common failure modes, analysts can quickly identify and resolve problems that impact network discovery accuracy. This training guide is suitable for cybersecurity students, SOC analysts, and red-team operators who want to strengthen their troubleshooting and network analysis skills.