

Journal Scientific and Applied Research, Vol. 31, 2026
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

Original Contribution

EXPERIMENTAL SIMULATION AND ANALYSIS OF LAYER 2 ATTACK VECTORS: ARP/DNS SPOOFING AND MITM

Denis I. Zvezdelinov[1], Zdravko Y. Kuzmanov[2]

[1]Department of Communication and Computer Engineering and Security Technologies, Faculty of Technical Sciences, Konstantin Preslavsky University of Shumen, Shumen 9712, 115 Universitetska St., e-mail: d.zvezdelinov@shu.bg
[2]Department of Communication and Computer Engineering and Security Technologies, Konstantin Preslavsky University of Shumen, Shumen 9712, 115 Universitetska St., e-mail: z.kuzmanov@shu.bg

Abstract: *This report examines the security of the Data Link Layer (Layer 2) of the OSI model, focusing on fundamental vulnerabilities in core local communication protocols. It addresses the concept of "implicit trust" inherent in the design of the Address Resolution Protocol (ARP) and the Domain Name System (DNS), which enables Man-in-the-Middle (MITM) attacks. The report analyzes the theoretical mechanisms of ARP cache poisoning and DNS spoofing. Network traffic was intercepted and subjected to forensic analysis through controlled simulations in an isolated laboratory environment. Key packet-level Indicators of Compromise (IoCs) were identified, including duplicate MAC addresses, an anomalous frequency of unsolicited ARP responses (Gratuitous ARP), and unusually low DNS resolution response times. In conclusion, architectural and software-based protective measures are outlined, including the implementation of Dynamic ARP Inspection (DAI), Port Security, and encrypted DNS protocols (DoH/DoT).*

Key words: *ARP Spoofing, DNS Spoofing, Man-in-the-Middle (MITM), Data link (Layer 2), Network forenscics, Indicator of compromise (IoCs), Dynamic ARP Inspection.*

1. Introduction

The network architecture of modern local area networks (LANs) relies on protocols designed during the early stages of the Internet's development, when functionality and interoperability were the primary priorities rather than protection against malicious actions [7]. As a result, fundamental protocols at Layer 2 (Data Link Layer) and Layer 3 (Network Layer) of the OSI model operate on the principle of implicit trust. The lack of built-in authentication and cryptographic verification mechanisms renders local networks vulnerable to internal attack vectors. One of the most critical issues in IPv4 networks is associated with the operation of the Address Resolution Protocol (ARP, RFC 826) [6]. Since ARP does not support origin verification for messages, any host in the local segment can send spoofed ARP responses, altering the MAC-to-IP mapping tables of other devices. Executing so-called ARP

cache poisoning allows a node to covertly redirect traffic between a client and the default gateway through itself, thereby becoming a Man-in-the-Middle (MITM). Once positioned as a forwarding node, the attacker gains the ability to perform passive eavesdropping (sniffing), modify data in real time, or redirect application traffic using subsequent techniques such as DNS spoofing. This compromises the confidentiality and integrity of even those systems that rely on external security infrastructure (such as perimeter firewalls or IDS/IPS).

Experimental modeling enables complex network processes and attack mechanisms to be reproduced under controlled conditions, allowing theoretical assumptions to be compared with observable system behavior and experimentally obtained data [9].

The primary objective of this report is to investigate the anatomy of Layer 2 attacks by comparing theoretical models of ARP and DNS redirection with real network artifacts generated during their analysis. Through Deep Packet Inspection (DPI) of captured .pcap datasets, specific anomalies and detection rules for MITM activity are defined, alongside corresponding hardening strategies for the local network infrastructure.

2. Theoretical model of the attacks

The vulnerabilities enabling Man-in-the-Middle (MITM) attacks in local area networks stem directly from the architectural specifications of the underlying protocols. Since the primary goal of protocols like ARP and DNS was to ensure fast and seamless communication, authentication mechanisms were left to the higher layers. This section examines the theoretical foundation of two of the most common local attack vectors. The Address Resolution Protocol (ARP, RFC 826) performs the critical role of mapping logical IP addresses (Layer 3) to physical MAC addresses (Layer 2). To optimize network traffic, every device maintains a local table – the ARP Cache – where it temporarily stores learned mappings between IP and MAC addresses. The fundamental flaw of ARP lies in its stateless nature. Devices automatically accept and process incoming ARP responses (ARP Replies), even if they have not previously sent a corresponding ARP request (ARP Request). This mechanism, known as Gratuitous ARP (unsolicited ARP), was originally designed to rapidly update network tables when a network card or IP address changes. An attacker abuses this functionality through the following mechanism: generating a Gratuitous ARP Reply to the client device (the victim), claiming that the IP address of the default gateway is associated with the attacker's MAC address. Simultaneously, the attacker sends an ARP Reply to the router, claiming that the victim's IP address corresponds to the attacker's own MAC address. The operating systems of both the victim and the router update their ARP tables without additional verification. As a result of this bidirectional "poisoning", all traffic between the victim and the gateway now physically traverses the attacker's network interface. To avoid causing a Denial of Service (DoS) and remain undetected, the attacker enables IP forwarding on their machine, passing traffic along to its real destination. After successfully positioning themselves as a Man-in-the-Middle, the attacker gains full access to all unencrypted requests from the victim, including those to the Domain Name System (DNS). DNS is the protocol responsible for translating human-readable domain names (e.g., example.com) into corresponding IP addresses. Traditional DNS queries are sent in cleartext over UDP (port 53) in accordance with the RFC 1035 specification [5]. Since UDP is a connectionless protocol and standard DNS lacks cryptographic signatures, the client accepts the first response received that contains a matching Transaction ID. The DNS Spoofing attack (or local-level DNS Cache Poisoning) is executed as follows: thanks to ARP Spoofing, the attacker "sees" the victim's DNS query (for instance, for the domain of a popular bank or social network) before it leaves the local network. The attacking machine immediately generates a forged DNS response before the real DNS server (located on the Internet) can return the legitimate

one. In this spoofed response, the Address Record (A Record) for the requested domain is replaced with the IP address of a web server controlled by the attacker. The victim's browser receives the response and, suspecting no deception, loads the fake web page. Because the URL in the browser bar appears fully legitimate (the victim typed the correct name), the attack is exceptionally difficult for an end user to visually recognize. By combining these two techniques (ARP and DNS Spoofing), the attacker constructs a complete compromise chain – first seizing control of the physical data flow (Layer 2), and then manipulating logical redirection (Layer 7) to guide the victim toward malicious infrastructure.

3. Experimental setup and tooling

To practically validate the theoretical models of Layer 2 attacks, a series of controlled experiments was conducted. To comply with ethical and legal standards for penetration testing, the simulations were executed entirely within an isolated laboratory subnet (192.168.8.0/24), the architecture of which was described in detail in previous reports. The use of a physically segmented environment ensures that malicious packets (such as Gratuitous ARP) do not affect external production networks. To execute the attack and perform its subsequent network analysis, the devices in the setup are:

• Gateway (GL.iNet Beryl AX): The network router serves as the egress point to external networks (the Internet) and is responsible for resolving legitimate DNS queries forwarded by clients. Within the scope of the experiment, its IP address is 192.168.8.1.

• Target / Victim (Client Device): A standard workstation connected to the local network and configured to receive its network settings dynamically (via DHCP) from the gateway. For testing purposes, its IP address is 192.168.8.100. The device generates standard user traffic, including DNS queries and HTTP/HTTPS sessions.

• Primary Software Tool (Ettercap): To automate the attack, Ettercap [10] is used – an established open-source software for protocol analysis and executing MITM attacks in local networks. Ettercap features built-in modules (plugins) for executing ARP Poisoning and DNS response spoofing in parallel, while automatically enabling IP forwarding (ip_forward) to prevent connection disruption for the victim.

• Analytical and Monitoring Node (Monitor): This is the most critical component for network forensics purposes. Unlike standard scenarios where either the attacker or the victim records traffic (which can lead to skewed results due to local packet processing), an impartial external observer is used in this setup.

• Equipment: A Raspberry Pi 4B single-board computer, connected to the network via a passive network tap (Throwing Star LAN Tap Pro) physically positioned between the switch and the gateway.

• Configuration: The Raspberry Pi interfaces operate in pure promiscuous mode without an assigned IP address (Stealth Mode). This guarantees that the recording station remains completely invisible to Ettercap during the Network Discovery phase and cannot be targeted by ARP poisoning.

• Software: tcpdump is used to capture all bidirectional physical traffic and record it into a raw .pcap file for subsequent Deep Packet Inspection (DPI) using Wireshark [2].

This distributed architecture makes it possible to capture the exact incident timeline – from initial scanning and poisoned ARP packet injection to the final redirection of the victim's DNS queries – providing an ideal dataset of digital evidence.

4. Practical execution of the attack

The practical execution of the attack proceeds in three sequential phases: initial network mapping (Network Discovery), launching duplex ARP cache poisoning (ARP Cache Poisoning), and activating the DNS response spoofing module (DNS Spoofing). The entire procedure was conducted from the attacking node (192.168.8.50) using both the command-line and graphical interfaces of the Ettercap tool. To execute a precisely targeted attack without crashing the entire segment, the attacker must first identify the active IP and MAC addresses in the 192.168.8.0/24 subnet. Before initiating any redirection, packet forwarding at the kernel level must be enabled on the attacking machine. Otherwise, intercepted communication will be dropped at the attacking node, and the victim will immediately lose Internet connectivity:

- sudo sysctl -w net.ipv4.ip_forward=1

By running Ettercap in text mode or via the graphical module, the tool sends ARP requests to all possible addresses in the 192.168.8.1–192.168.8.254 range. As a result, a list of discovered active hosts (Host List) is generated:

- 192.168.8.1 – MAC: 00:11:22:AA:BB:CC (GL.iNet Beryl AX Gateway)
- 192.168.8.100 – MAC: 44:55:66:DD:EE:FF (Victim / Client Computer)

Once the targets are identified, the attack vector is defined. For full bidirectional traffic redirection, Ettercap must simultaneously poison both the gateway’s cache and the victim’s cache. Target assignment (Targeting):

- Target 1: 192.168.8.1 (Router IP address)
- Target 2: 192.168.8.100 (Victim IP address)

Executing ARP poisoning:

- sudo ettercap -T -q -i eth0 -M arp:remote /192.168.8.1// /192.168.8.100//

Parameters: -T (text mode), -q (suppress verbose output), -i eth0 (network interface), -M arp:remote (activate MITM via ARP with remote traffic redirection).

Injection Mechanism: Ettercap continuously (at intervals of a few seconds) sends forged Gratuitous ARP packets:

- To the victim (192.168.8.100): "IP 192.168.8.1 is at MAC 99:99:99:88:88:88" (the attacker's MAC address).
- To the gateway (192.168.8.1): "IP 192.168.8.100 is at MAC 99:99:99:88:88:88" (again, the attacker's MAC address).

From this moment, all traffic between the client device and the external network physically flows through the eth0 interface of the attacking node, which transparently forwards it while acting as a Man-in-the-Middle. Following successful positioning at Layer 2, the attacker proceeds to name manipulation at Layer 7 (Application Layer) using the built-in dns_spoof plugin. The redirection rule is defined in the DNS spoofing configuration file. A target domain is selected (e.g., secure-bank.lab or wildcard *.lab), which is to be redirected to a local rogue web server running on the attacker's machine (192.168.8.50):

- # /etc/ettercap/etter.dns
- secure-bank.lab A 192.168.8.50
- *.secure-bank.lab A 192.168.8.50

Activating the DNS plugin. The plugin is launched withing the context of the established MITM session:

1. sudo ettercap -T -q -i eth0 -P dns_spoof -M arp:remote /192.168.8.1// /192.168.8.100//

Response simulation:

2. The victim opens a browser and navigates to http://secure-bank.lab.

3. The device sends a standard UDP DNS query (port 53) for the domain's address record.

4. Ettercap intercepts the request in real time, suppresses/outpaces the response from the real DNS server, and immediately returns a forged UDP packet containing IP address 192.168.8.50.

5. The victim's browser receives the spoofed response and loads the rogue phishing page hosted by the attacker without displaying any network error warnings.

5. Network analysis and indicators of compromise (IoCs)

Following the execution of the experimental attack, the raw mitm_capture.pcap file collected by the independent analytical station (Raspberry Pi 4B) was subjected to detailed forensic analysis. Because the capture was performed via a passive physical tap (Throwing Star LAN Tap Pro), the generated dataset contains a complete and unmodified copy of all traffic flows that traversed the physical segment during injection. By applying Deep Packet Inspection (DPI) in Wireshark and TShark, specific network anomalies and Indicators of Compromise (IoCs) accompanying the execution of ARP and DNS Spoofing were identified. Analysis of the data link layer reveals clear deviations from standard ARP protocol behavior (RFC 826), characteristic of cache poisoning. Under normal operating conditions, ARP replies are sent primarily in response to specific ARP requests. During timeline analysis of mitm_capture.pcap, cyclical sending of unsolicited replies (Gratuitous ARP Replies) is observed at regular time intervals (approx. 2 seconds).

• Capture filter in Wireshark:

• arp.opcode == 2 && !arp.isgratuitous == false

Registered Behavior: The attacker's MAC address (99:99:99:88:88:88) periodically sends ARP replies to the broadcast address or directly to the victim and gateway, claiming ownership of IP addresses 192.168.8.1 and 192.168.8.100. The primary artifact unequivocally confirming ARP Spoofing is the appearance of the same MAC address associated with two different IP addresses within the same Layer 2 broadcast domain.

Table 1. ARP poisoning process and gateway MAC address spoofing

Time (s)	Source MAC	Source IP	Message	Status
10.001234	00:11:22:AA:BB:CC	192.168.8.1	192.168.8.1 is at 00:11:22:AA:BB:CC	Real
10.512390	99:99:99:88:88:88	192.168.8.50	192.168.8.50 is at 99:99:99:88:88:88	Attacker
11.102341	99:99:99:88:88:88	192.168.8.1	192.168.8.1 is at 99:99:99:88:88:88	Poisoned

Wireshark automatically flags this anomaly through its Expert Info subsystem module:

• System Anomaly Alert: Duplicate IP address configured / MAC address configured with multiple IP addresses. Detection Filter: arp.duplicate-address-detected

Following the takeover of Layer 2, the injection of spoofed DNS responses leaves distinct application-layer traces resulting from the attacker's physical and network proximity to the victim. When the victim sends a DNS query for an external domain (e.g., secure-bank.lab), a legitimate response from a remote DNS server on the Internet requires geographical routing across multiple hops, taking between 15 ms and 80 ms. Because the attacker resides on the same local segment and intercepts the query locally, the spoofed response from Ettercap reaches the client machine in less than 1.5 ms.

Filtering and RTT extraction via TShark:

• tshark -r mitm_capture.pcap -Y "dns.flags.response == 1" -T fields -e frame.time_delta -e dns.qry.name -e dns.a

Result: All manipulated DNS responses exhibit unnaturally low latency values (frame.time_delta < 0.002s), which serves as a clear indicator of Local Packet Injection.

Legitimate DNS responses originating from authoritative DNS servers contain dynamic TTL caching values (e.g., 300s, 3600s). Interception of the spoofed response reveals that Ettercap injects static or hardcoded TTL values (e.g., fixed 0s or 60s), designed to prevent prolonged caching that would hinder the attacker's ability to flexibly manage the attack. Based on the network analysis performed, the identified artifacts are categorized for automated detection in SIEM and IDS environments:

Table 2. Indicators of compromise during real-time MITM attack execution

Category	Indicator of Compromise (IoC)	Interception Method / Signature
Layer 2 (ARP)	Sudden spike in Gratuitous ARP replies	ARP broadcast packet monitoring: > 5 packets/sec from a single MAC.
Layer 2 (ARP)	ARP cache modification without preceding request	Comparison of gateway MAC address against an initial static baseline list.
Layer 2 (ARP)	MAC address overlap for Gateway and Host	arp.duplicate-address-detected filter in PCAP dataset.
Layer 7 (DNS)	DNS response with RTT < 2ms for external FQDN	Calculation of elapsed time between dns.id request and corresponding response.
Layer 7 (DNS)	Duplicate DNS responses (Race Condition)	Presence of two responses with identical DNS Transaction ID (first spoofed, second delayed legitimate).

6. Defense and prevention strategies

The results of the practical experiment and subsequent network analysis confirm that traditional Layer 2 and Layer 7 protocols (ARP, unencrypted DNS) are extremely vulnerable in the absence of additional security controls. Protecting local network infrastructure against Man-in-the-Middle (MITM) attacks requires a defense-in-depth approach encompassing switch hardware configurations, cryptographic protocols, and network segmentation. Because ARP cache poisoning attacks operate at the data link layer, their effective mitigation must be implemented directly on network switches.

• Dynamic ARP Inspection (DAI) [3, 8]: A critical security feature supported by managed switches (such as the Zyxel GS1200 series used in the laboratory topology or enterprise-grade equipment). DAI relies on the DHCP Snooping binding database, which monitors and records valid mappings between IP address, MAC address, VLAN, and physical port upon issuing a DHCP lease. [8] When an ARP response arrives at a port, the switch

cross-references it with the DHCP Snooping database. If the forged ARP packet contains a MAC address that does not match the IP address registered for that port, the switch immediately drops the packet and blocks the port.

• Port Security: Limits the number of MAC addresses that can be learned on a single physical switch port. If an attacker attempts to generate multiple fake MAC addresses from a single port, the port automatically enters an error-disabled state (err-disable) and drops the connection.

• Static ARP Entries: On critical systems (e.g., database servers or isolated industrial networks), the association between the IP and MAC address of the default gateway can be hardcoded statically in the operating system:

- Setting a static ARP entry for the gateway in Linux/Windows:
- arp -s 192.168.8.1 00-11-22-AA-BB-CC
- Limitation: This method is difficult to scale in dynamic enterprise networks with a large number of client devices.

Protection against DNS response spoofing is achieved by transitioning from unencrypted UDP queries to protocols featuring cryptographic authentication and channel encryption:

• DNSSEC (DNS Security Extensions) [1]: Adds digital signatures to DNS records. When a client issues a query, the authoritative DNS server returns a cryptographic signature (RRSIG). Even if an attacker injects a spoofed response via MITM, the client's validation mechanism discards the packet due to the absence of a valid zone private key.

• DNS over HTTPS (DoH, RFC 8484) [4]/ DNS over TLS (DoT, RFC 7858): Encapsulates DNS traffic within an encrypted TLS tunnel (over ports 443 or 853). This renders it impossible for Ettercap or any other tool to read intercepted requests in cleartext or inject spoofed responses at Layer 2/7.

Architectural Control Measures:

• VLAN Segmentation (IEEE 802.1Q): Dividing the local network into isolated virtual subnets (e.g., separate VLANs for users, IoT devices, management environments, and guests) restricts the scope of broadcast domains. An ARP Spoofing attack launched within one VLAN cannot affect devices located in another VLAN.

• IEEE 802.1X Implementation (Network Access Control): Requires authentication of every newly connected device at the physical port level via a RADIUS server before granting any access to the network infrastructure.

7. Conclusion

The research conducted within this report confirmed that the structure of Layer 2 and the core protocols of the IPv4 stack contain critical vulnerabilities arising from the blind trust model. Through the simulation of ARP and DNS Spoofing, it was demonstrated how easily an insider attacker can seize control of the physical data flow and manipulate application services. At the same time, forensic analysis of the collected .pcap datasets proved that Man-in-the-Middle attacks generate distinct Indicators of Compromise (IoCs) – anomalous Gratuitous ARP reply intensity, duplicate MAC addresses, and unnaturally low response times for DNS answers. Recognizing these signatures through passive monitoring, alongside the timely deployment of hardware defense mechanisms (such as DAI and encrypted DNS), guarantees a high degree of resilience for local network environments against modern cyberattacks.

References

[1] Arends, R., Austein, R., Larson, M., Massey, D., & Rose, S. (2005). DNS Security Introduction and Requirements. RFC 4033, Internet Engineering Task Force (IETF).

[2] Bejtlich, R. (2013). The Practice of Network Security Monitoring: Understanding Incident Detection and Response. San Francisco.

[3] Convery, S. (2004). Network Security Architectures. Indianapolis: Cisco Press.

[4] Hoffman, P., & McManus, P. (2018). DNS Queries over HTTPS (DoH). RFC 8484, Internet Engineering Task Force (IETF).

[5] Mockapetris, P. (1987). Domain Names - Implementation and Specification. RFC 1035, Internet Engineering Task Force (IETF).

[6] Plummer, D. C. (1982). An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware. RFC 826, Internet Engineering Task Force (IETF).

[7] Stallings, W. (2021). Computer Security: Principles and Practice (4th ed.). Pearson.

[8] Cisco Systems. (2021). Configuring Dynamic ARP Inspection (DAI). Catalyst Security Configuration Guide. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750/software/release/12-2_55_se/configuration/guide/3750scg/swdynarp.html

[9] Dimanova, D. “Theory of Modeling.” Journal Scientific and Applied Research, Vol. 4, 2013, pp. 106–111. [Online]. Available: https://doi.org/10.46687/jsar.v4i1.87.

[10] Orru, A., & Valleri, A. (2022). Ettercap: A suite for man in the middle attacks. [Online]. Available: https://www.ettercap-project.org/.