

Analysis of ARP Spoofing Attacks and Their Impact on Local Network Security

Harshdeepsinh Chudasama¹

¹ Computer Engineering Department, Gandhinagar University, Gandhinagar, Gujarat, India

Abstract

The Address Resolution Protocol (ARP) is a foundational component of local area network communication, responsible for resolving IP addresses to MAC addresses within a subnet. Its stateless and unauthenticated design makes it susceptible to a class of attacks known as ARP spoofing, where a malicious actor transmits falsified ARP messages to corrupt the ARP caches of other devices on the same network segment. This paper presents a detailed analysis of ARP spoofing attacks, covering the protocol's inherent weaknesses, the step-by-step mechanics of cache poisoning, and the tools commonly used by attackers. It further investigates the impact of these attacks on LAN and public Wi-Fi environments, addressing data interception, session hijacking, and network availability disruption. A review of current mitigation techniques including Dynamic ARP Inspection (DAI), static ARP entries, and network monitoring is provided, along with an assessment of their practical effectiveness. The paper also discusses emerging trends such as machine learning-based anomaly detection and Software-Defined Networking as future directions for defending local networks against ARP-based threats.

Keywords: ARP spoofing, ARP poisoning, MAC address, man-in-the-middle attack, LAN security, Dynamic ARP Inspection, session hijacking, gratuitous ARP, network monitoring

1. Introduction

Every time a device on a local network wants to send a packet to another IP address on the same subnet, it must first discover the hardware address associated with that IP. This is the job of the Address Resolution Protocol (ARP), specified in RFC 826 and unchanged in its fundamental design since 1982 [1]. It is a simple, broadcast-based mechanism with no authentication, and that design has proven to be a persistent source of local network vulnerability.

ARP spoofing, also referred to as ARP poisoning, takes advantage of the fact that any device on a local segment can broadcast ARP replies claiming any IP-to-MAC binding, and most neighbouring devices will update their caches without verifying the claim. An attacker exploiting this behaviour can redirect traffic meant for a legitimate host, including the default gateway, through their own machine. The attack requires physical or wireless access to the same broadcast domain as the target, making it a local network threat by nature.

A successful ARP spoofing attack positions the attacker between two communicating parties, granting the ability to intercept credentials, inspect unencrypted traffic, or disrupt connectivity.

Tools like Ettercap and Arpspoof have lowered the skill threshold for executing such attacks, making them accessible well beyond the professional security community [3].

Despite being a well-documented problem, ARP spoofing remains relevant because the protocol itself has not been replaced or secured at the design level. Mitigations exist at the switch level through Dynamic ARP Inspection and at the host level through static ARP table entries, but both require deliberate configuration and are not universally deployed. Public Wi-Fi environments and many corporate LANs remain exposed as a result [6].

The sections that follow examine the ARP protocol mechanism, categorise the forms ARP spoofing takes in practice, describe the tools used, assess the real-world impact on local networks, and evaluate the mitigation landscape. The final section considers where detection and prevention are heading, given growing interest in machine learning-based traffic analysis and SDN-based ARP control.

2. Background and Context

2.1 How ARP Works

When a host needs to communicate with an IP address on its local subnet, it first consults its ARP cache, a short-lived table stored in memory that maps IP addresses to MAC addresses. If no entry is found, the host broadcasts an ARP Request frame to the entire subnet asking which device owns a particular IP address. The device with that IP replies directly with an ARP Reply containing its MAC address. The requester stores this mapping in its cache and uses it for subsequent frames [1].

The vulnerability lies in two design properties. First, ARP is stateless: a device will accept and cache an ARP Reply even if it never sent a corresponding request. Second, there is no authentication: any device on the segment can claim to own any IP address, and the receiver has no means to verify that claim. These properties reflect trade-offs made for simplicity and broadcast efficiency in a trusted environment [2]. In contemporary enterprise or public networks, they represent a meaningful attack surface that has been exploited in practice for over two decades.

2.2 A Real-World Analogy

Imagine an office building where inter-department mail is sorted by desk nameplates. When a colleague wants to send a document to the Finance Manager, they look up the desk location on a shared board and drop the envelope there. Now suppose a malicious employee quietly swaps their own nameplate with the Finance Manager's. All mail intended for Finance now arrives at the attacker's desk. The attacker can read every document, copy it, and forward it to the real Finance Manager so nothing seems wrong. This mirrors an ARP spoofing-based man-in-the-middle attack: the directory (the ARP cache) has been corrupted, and the real owner of the address is bypassed without either party being immediately aware.

2.3 Historical and Research Context

The exploitability of ARP's unauthenticated design was discussed in early security research examining the trust assumptions baked into the TCP/IP protocol suite [2]. The release of the Dsniff toolkit in 2000 by Dug Song, which included the Arpspoof utility, brought practical ARP poisoning tools into mainstream security testing [3]. Ettercap followed, offering a more feature-rich environment for man-in-the-middle attacks over local networks, and was presented at Black Hat Europe in 2003 [8].

Subsequent academic work explored both attack refinements and defensive responses. Lootah, Enck, and McDaniel proposed TARP, a ticket-based ARP replacement that introduces authentication tokens to ARP exchanges, representing one of the more thorough attempts to address the protocol's trust deficit at the design level [4]. More recently, research has turned toward detection rather than protocol replacement, applying machine learning to ARP traffic features to distinguish poisoning attempts from legitimate cache updates [5].

3. Types of ARP Spoofing Attacks

3.1 Man-in-the-Middle via Bidirectional ARP Poisoning

The most operationally significant use of ARP spoofing is the establishment of a man-in-the-middle (MITM) position between two hosts on the same local segment. The attacker simultaneously sends forged ARP replies to the victim host and to the default gateway. The victim is told that the gateway's IP address maps to the attacker's MAC address; the gateway is told that the victim's IP maps to the attacker's MAC. With IP forwarding enabled on the attacker's machine, packets flow through the attacker transparently in both directions. Neither end host detects any disruption, while the attacker gains full visibility into the traffic stream [4].

This bidirectional poisoning is what distinguishes a full MITM from a simple redirect. It preserves the communication session, which makes the attack harder to notice and allows the attacker to remain in position for extended periods. The Ettercap tool automates maintenance of this state by periodically re-sending forged replies, since ARP cache entries expire and must be refreshed [8].

3.2 ARP Cache Poisoning for Targeted Traffic Redirection

Not every ARP spoofing attack aims at full traffic interception. In some cases, the objective is to redirect traffic destined for a specific host, such as an internal server, toward a machine the attacker controls. By poisoning only the victim's ARP cache entry for that specific server's IP, the attacker can capture login requests, serve a forged version of the application, or observe the data exchanged without positioning themselves between the victim and the gateway for all traffic. This more selective approach reduces the attacker's network footprint and may evade detection systems that look for broad ARP anomalies [4].

3.3 ARP-Based Denial of Service

ARP spoofing can be directed at network availability rather than data confidentiality. The most straightforward form involves poisoning a victim's ARP cache to associate the default gateway's IP address with a non-existent or invalid MAC address. When this happens, frames the victim

sends toward the gateway are never received by any real device, effectively disconnecting that host from the rest of the network without touching any hardware. The attack produces no visible error at the network layer, making it difficult to diagnose without directly inspecting the ARP table [6].

A broader form of this disruption involves flooding the segment with continuous gratuitous ARP announcements targeting multiple hosts simultaneously. Each announcement overwrites the gateway entry in the receiving host's ARP cache with a bogus mapping. If the rate of these spoofed packets exceeds the rate at which legitimate ARP traffic corrects the cache, multiple devices lose gateway connectivity in parallel. Tripathi and Hubballi documented this class of ARP-based availability attacks and analysed the conditions under which they succeed in both wired LAN and wireless environments [6].

3.4 Gratuitous ARP Exploitation

A gratuitous ARP is an ARP Reply sent by a device without a preceding request, typically broadcast at startup to announce the sender's IP-to-MAC mapping and prompt neighbouring devices to refresh their caches. While the mechanism serves legitimate purposes such as IP conflict detection and high-availability failover, it is straightforward to abuse. An attacker can craft and broadcast a gratuitous ARP claiming that the gateway's IP address belongs to the attacker's MAC. Because most operating systems accept unsolicited ARP Replies and update their caches accordingly, this approach can affect many hosts on the broadcast domain that receive the frame [1].

The attack requires no prior knowledge of existing sessions and no direct interaction with target devices. Sending a crafted gratuitous ARP broadcast at regular short intervals is enough to maintain a poisoned state, since legitimate cache refreshes and device reboots will periodically restore correct mappings that the attacker must overwrite again.

4. Techniques and Tools Used

4.1 Ettercap

Ettercap is a MITM framework designed for local network interception, first presented at Black Hat Europe in 2003 [8]. It is available in both graphical and command-line modes and automates the ARP poisoning workflow from host discovery through continuous cache maintenance. When a user selects two targets and initiates an ARP poison attack, Ettercap sends forged ARP replies to both targets at a configurable interval, associating each target's IP with the attacker's MAC address. It simultaneously enables kernel IP forwarding so that intercepted packets reach their intended destination after inspection. Built-in dissector plugins parse the contents of protocols including HTTP, FTP, SSH handshakes, and POP3, making it practical to observe credentials from unencrypted sessions.

Because Ettercap continuously re-sends spoofed ARP replies, the poisoned cache state persists even as entries expire. This persistence makes manual detection through periodic ARP table

inspection less reliable, since the table appears correct immediately after a legitimate refresh but is re-poisoned within seconds.

4.2 Arpspoof

Arpspoof, part of Dug Song's Dsniff toolkit released in 2000, is a focused command-line tool whose sole function is sending continuous spoofed ARP replies to a specified target [3]. Unlike Ettercap, it has no built-in sniffing or session manipulation capabilities, which makes it lighter and less conspicuous from a process and traffic standpoint. Penetration testers often prefer it for its simplicity and predictability.

A typical deployment involves running two Arpspoof instances in parallel: one poisoning the victim's cache with a false entry for the gateway's IP, and one poisoning the gateway's cache with a false entry for the victim's IP. With IP forwarding enabled at the OS level, this two-sided poisoning establishes a MITM position while preserving network connectivity for the victim.

4.3 Wireshark

Once a MITM position is established, Wireshark is commonly used to examine intercepted traffic. As a passive packet analyser, it captures all frames on the network interface and presents them through protocol dissectors that reconstruct application-layer sessions from raw frame data. HTTP requests and responses, FTP control channels, Telnet sessions, and similar unencrypted communications are rendered in readable form, with display filters available to isolate traffic by protocol, port, or endpoint address. The Wireshark project is maintained as open-source software and is widely used in both offensive testing and defensive network analysis [9].

From a defensive perspective, Wireshark is useful for identifying ARP poisoning in progress. Analysts can filter for ARP traffic and look for frames where the same IP address appears in replies from two different MAC addresses, or where a device is sending ARP Replies at an unusually high rate. Both patterns are consistent indicators of an active spoofing attempt and can prompt further investigation.

4.4 Step-by-Step Attack Execution

A complete ARP spoofing-based MITM attack typically follows a predictable sequence. The attacker first joins the target network, either physically via an Ethernet port or wirelessly by connecting to the access point, and uses a host discovery tool such as Nmap or Netdiscover to enumerate active IP addresses and identify the default gateway. Next, the attacker launches Arpspoof or Ettercap against the victim host and the gateway, beginning continuous delivery of forged ARP replies. IP forwarding is enabled on the attacker's machine so that captured frames are relayed correctly in both directions. Finally, Wireshark is used on the attacker's interface to record and inspect the traffic now flowing through the attacker. The setup is straightforward and requires no elevated privileges beyond local network access [3][8].

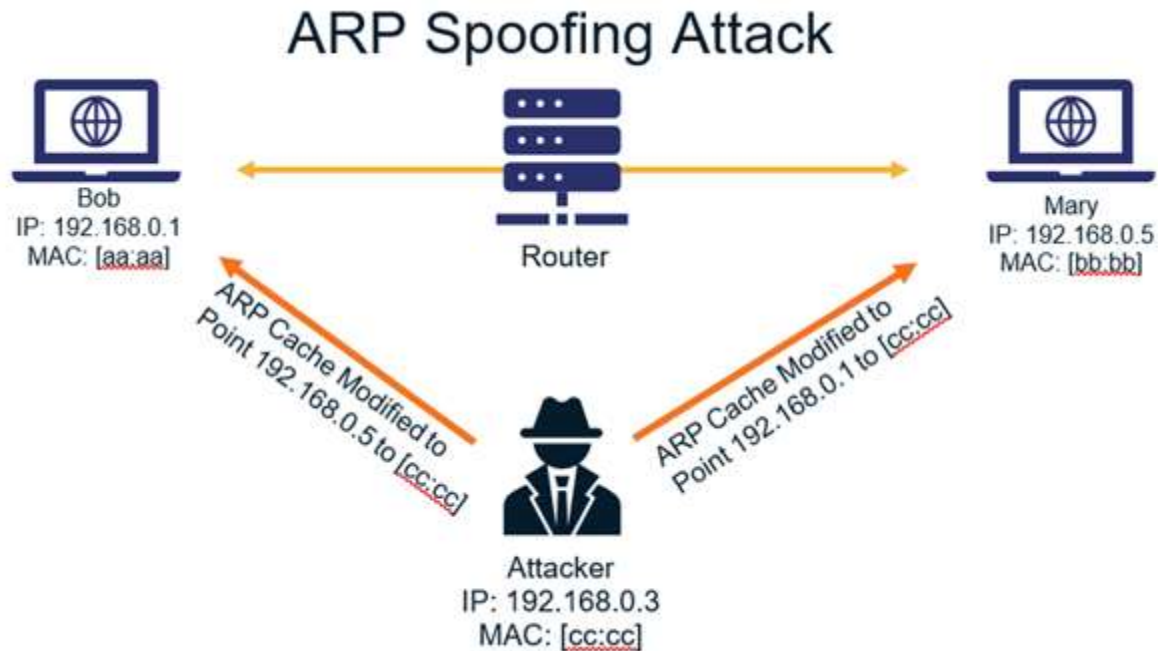


Figure 1: ARP Spoofing Attack Workflow in a Local Network (adapted from [10])

5. Impact of ARP Spoofing Attacks

5.1 Data Interception on LAN Environments

Within a corporate LAN, a successful ARP spoofing attack gives the attacker visibility into traffic exchanged between the poisoned hosts. Although HTTPS is widely used for external web traffic, internal communications frequently rely on unencrypted protocols. Database connection traffic, internal HTTP applications, network printer communications, legacy enterprise software using Telnet or FTP, and SMTP exchanges between internal mail servers may all be transmitted in plaintext. An attacker holding a MITM position can reconstruct these sessions from captured frames and extract credentials, business data, or configuration information [4].

The attack is largely transparent to the end hosts. Neither the victim nor the gateway experiences connection errors or obvious disruption under normal conditions. The only observable effect may be a slight increase in round-trip time as packets travel the additional hop through the attacker's machine, a difference that is usually too small to be noticed without deliberate measurement.

5.2 Session Hijacking

ARP spoofing provides the packet-level visibility needed to carry out session hijacking against active authenticated sessions. When a user has already logged into a web application and a session cookie has been issued, that cookie is included with every subsequent HTTP request. An attacker capturing this traffic can extract the session token and replay it in their own browser to impersonate the authenticated user, without needing the user's credentials. The attack targets the session layer

rather than the authentication mechanism, meaning that strong passwords and two-factor authentication at login do not mitigate the risk once an active session is in progress [4].

This risk is most pronounced for applications that transmit session cookies over HTTP rather than HTTPS, or that do not bind session tokens to specific client addresses. On networks where ARP spoofing is viable and such applications are in use; an attacker can harvest valid session tokens for multiple concurrent users.

5.3 Threats in Public Wi-Fi Environments

Public Wi-Fi networks present a particularly high-risk environment for ARP spoofing. In these settings, all connected devices share the same broadcast domain with no established trust relationship between them. There is typically no 802.1X authentication, no network access control, and no switch-level protection such as Dynamic ARP Inspection. An attacker who connects to the same access point as their targets can poison ARP caches across the wireless segment without requiring any special access or privilege [6].

The risk is compounded by user behaviour on these networks. People connecting to public Wi-Fi often access webmail, corporate VPNs with HTTP fallback, internal applications, and services that may not enforce strict HTTPS. Each of these creates an opportunity for credential or session theft under an ARP-based MITM attack, and many users are unaware that such attacks are feasible on shared wireless segments.

5.4 Network Availability Disruption

Poisoning a host's ARP cache with an invalid MAC address for the default gateway silently disconnects that host from external networks without triggering any obvious error condition. The host continues to operate normally for local traffic but cannot reach anything beyond the subnet. This type of targeted disconnection can be used to disrupt specific services, isolate a device from security monitoring systems, or create an outage that coincides with other malicious activity on the network [6].

In environments running time-sensitive processes, such as industrial control systems or networked medical equipment, connectivity disruptions caused by ARP cache corruption can carry consequences that are disproportionate to the simplicity of the underlying attack. The absence of visible error messages at the application layer further delays diagnosis and recovery.

6. Mitigation Strategies

6.1 Dynamic ARP Inspection (DAI)

Dynamic ARP Inspection is a switch-level security feature that validates ARP packets on untrusted ports against a DHCP snooping binding table before allowing them to pass. When a host obtains an IP address via DHCP, the switch records the IP-MAC-port binding. Subsequently, when an ARP Reply arrives on an untrusted port, the switch compares the claimed IP-MAC mapping against its binding table. If the mapping does not match a recorded entry, the packet is dropped

and an alert is logged. Trusted ports, typically uplinks to other switches or routers, bypass this validation [9].

DAI breaks the ARP spoofing attack at the network layer: forged ARP replies are dropped before reaching other devices, regardless of how frequently the attacker sends them. The mechanism is transparent to legitimate traffic and requires no changes on end hosts. Its main limitation is that it requires managed switches with DAI support, meaning it is unavailable in home networks and many unmanaged environments. Careful configuration is also needed to avoid inadvertently blocking legitimate gratuitous ARPs used by clustering software or virtual machine live migration [7].

6.2 Static ARP Entries

For a small number of critical devices such as default gateways, DNS servers, and domain controllers, administrators can configure static ARP entries in the ARP tables of hosts on the network. A static entry is a manually added IP-to-MAC mapping that the operating system will not overwrite in response to incoming ARP replies. Because the entry is fixed, forged ARP announcements targeting that IP-MAC pair have no effect on hosts where the static entry exists [7].

This approach works well for infrastructure devices whose MAC addresses rarely change. Managing static ARP entries across a large fleet of workstations is operationally demanding and does not scale in environments with dynamic assignment or frequent hardware changes. It is most practical as a targeted protection for the most critical mappings rather than a general-purpose solution.

6.3 Network Monitoring and Anomaly Detection

Monitoring ARP traffic in real time is a practical complement in environments where switch-level controls are absent or incomplete. Tools such as ARPwatch maintain a persistent record of IP-to-MAC mappings observed on the network and generate alerts when a mapping changes unexpectedly, when a new device appears claiming an existing IP, or when the same MAC address is associated with multiple IPs. These conditions are characteristic of ARP spoofing and can trigger incident response procedures [5].

Intrusion detection systems such as Snort include rule sets designed to detect ARP anomalies. Custom rules can alert on gratuitous ARP floods, duplicate IP-MAC mappings, or ARP Reply rates that exceed normal baselines for a given host. Monitoring alone does not prevent the attack but meaningfully reduces the window during which an attacker can operate undetected, giving network teams time to investigate and respond.

6.4 Network Segmentation and VLAN Isolation

Since ARP operates within a single broadcast domain, its reach is bounded by that domain's boundaries. Dividing a network into VLANs at the switch level creates separate broadcast domains, each with its own ARP scope. An attacker on one VLAN cannot send ARP frames

directly to devices on another VLAN; inter-VLAN traffic is routed rather than switched, at which point ARP is not involved in forwarding decisions. This does not eliminate the risk within a VLAN but limits the scope of any single spoofing attempt [7].

For wireless environments, client isolation is the equivalent control. When enabled on an access point, client isolation prevents wireless clients from exchanging frames directly with each other, routing all inter-client traffic through the access point instead. This removes the direct layer-2 visibility that ARP spoofing depends on, making the attack impractical on properly isolated guest or public Wi-Fi networks.

7. Future Trends and Challenges

7.1 Machine Learning-Based ARP Anomaly Detection

Threshold-based ARP monitoring tools are effective at catching straightforward spoofing attempts but can struggle in environments with high legitimate ARP traffic variability, producing false positives that reduce operator confidence over time. Machine learning offers a more adaptive approach. Research by Ullrich et al. demonstrated that features extracted from ARP traffic, including reply-to-request ratios, the rate of new IP-MAC mappings per host, and the frequency of mapping changes, can be used to train classifiers that distinguish poisoning attempts from legitimate ARP behaviour with improved accuracy over rule-based systems [5].

Results in controlled evaluations have shown measurable improvements in detection performance, though accuracy on live enterprise traffic with its irregular patterns and long tail of edge cases remains an open area of research. Practical deployment requires labelled training data from known ARP attack scenarios and periodic retraining as network conditions evolve. The approach is best understood as a complement to switch-level controls rather than a replacement for them.

7.2 SDN-Based ARP Management

Software-Defined Networking architectures centralise network intelligence in a controller that maintains an authoritative view of the network topology. In an SDN environment, ARP handling can be shifted from the distributed broadcast model to a centralised proxy: when a host sends an ARP Request, the SDN controller intercepts it, consults its topology database, and replies with the verified MAC address. No ARP broadcast reaches the wider segment, and no device on the network can inject a forged reply into that exchange [7].

This architectural approach addresses the problem at its root rather than detecting or blocking attacks after the fact. Several SDN controllers and OpenFlow-based implementations include ARP proxy modules for exactly this reason. The practical barrier is deployment scale: SDN adoption has grown steadily but remains concentrated in data centre and campus backbone environments. For edge and access layer networks, traditional ARP broadcast behaviour remains the norm, and the vulnerability persists with it.

7.3 Persistent Challenges in Legacy and Unmanaged Environments

Advances in detection and prevention are meaningful for organisations that can invest in managed switching infrastructure, modern network architecture, and dedicated security operations. A significant share of deployed networking equipment does not meet these criteria. Unmanaged switches without DAI support, access points without client isolation, and embedded devices running firmware that cannot be updated are common in small business, industrial, and residential environments. In these settings, ARP spoofing is as viable today as it was two decades ago.

This is primarily a deployment and economic challenge rather than a technical one. The defences needed to protect against ARP spoofing are well understood and have been available for many years. Achieving consistent deployment across the full range of network environments, not only in organisations with dedicated security teams, remains the most significant practical gap.

8. Conclusion

ARP spoofing is a local network attack with a straightforward execution path and a well-understood set of consequences. The absence of authentication in the ARP protocol is a design choice inherited from a different era of networking, and it has not been corrected at the protocol level. On any local segment where an attacker has access, devices that rely on ARP for address resolution are at risk of having their traffic intercepted or their connectivity disrupted.

Effective defences are available. Dynamic ARP Inspection, when deployed on managed switches, stops spoofed replies at the network layer. Static ARP entries protect critical IP-to-MAC mappings from being overwritten. Continuous monitoring with tools like ARPwatch narrows the window during which an attack can go unnoticed. VLAN segmentation and wireless client isolation reduce the broadcast domain that ARP spoofing requires. The challenge is not identifying what works but ensuring these controls are consistently deployed across diverse and often unmanaged environments.

Looking ahead, machine learning-based detection and SDN-based ARP proxy architectures offer meaningful improvements over current approaches, providing greater adaptability and the potential to address the problem structurally rather than reactively. Until such approaches see wider deployment, ARP spoofing will remain a relevant threat on local networks, and the case for applying existing mitigations more broadly remains strong.

References

- [1] D. C. Plummer, "An Ethernet Address Resolution Protocol," Request for Comments RFC 826, Internet Engineering Task Force (IETF), Nov. 1982. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc826>
- [2] S. M. Bellovin, "Security Problems in the TCP/IP Protocol Suite," ACM SIGCOMM Computer Communication Review, vol. 19, no. 2, pp. 32-48, Apr. 1989. [Online]. Available: <https://doi.org/10.1145/378444.378449>
- [3] D. Song, "Dsniff: A Collection of Tools for Network Auditing and Penetration Testing," Monkey.org, 2000. [Online]. Available: <https://monkey.org/~dugsong/dsniff/>

- [4] W. Lootah, W. Enck, and P. McDaniel, "TARP: Ticket-Based Address Resolution Protocol," *Computer Networks*, vol. 51, no. 15, pp. 4322-4337, Oct. 2007. [Online]. Available: <https://doi.org/10.1016/j.comnet.2007.05.007>
- [5] M. Ullrich, J. Lucke, and T. Zseby, "Evidence for IPv6 and ARP Spoofing Attack Detection Using Machine Learning," in *Proc. IEEE 44th Conference on Local Computer Networks (LCN)*, Osnabrueck, Germany, Oct. 2019, pp. 286-294. [Online]. Available: <https://doi.org/10.1109/LCN44214.2019.8990783>
- [6] N. Tripathi and N. Hubballi, "Exploiting ARP for Efficient Denial of Service Attacks," in *Proc. IEEE 13th International Conference on Advanced Communication Technology (ICACT)*, Seoul, South Korea, Feb. 2011, pp. 1151-1156. [Online]. Available: <https://ieeexplore.ieee.org/document/5745952>
- [7] A. Bruschi, A. Lanzi, and M. Monga, "A Defense Against ARP Spoofing Attacks Using Anti-ARP," in *Proc. ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Taipei, Taiwan, Mar. 2006, pp. 163-172. [Online]. Available: <https://doi.org/10.1145/1128817.1128841>
- [8] A. Ornaghi and M. Valleri, "Ettercap: A Multipurpose Sniffer/Interceptor/Logger for Switched LANs," presented at *Black Hat Europe*, Amsterdam, Netherlands, 2003. [Online]. Available: <https://www.blackhat.com/presentations/bh-europe-03/bh-eu-03-valleri.pdf>
- [9] Cisco Systems, "Configuring Dynamic ARP Inspection," *Cisco IOS Software Configuration Guide*, Cisco Press, San Jose, CA, USA, 2020. [Online]. Available: <https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/dynarp.html>
- [10] Shang Suru, "ARP Spoofing Demonstration Repository," GitHub, [Online]. Available: <https://github.com/shangsuru/arpspoof>

Corresponding author: Harshdeepsinh Chudasama
E-mail address: 230101105015@gandhinagaruni.ac.in