A Lightweight Real-time Framework for Detecting Rogue Switches in Wired Local Area Network

Vijay Bhuse, Vijay Prathap, Reddy Kanipakam and Xinli Wang
Grand Valley State University, Allendale, MI, USA

bhusevij@gvsu.edu
kanipakv@mail.gvsu.edu
wangx@gvsu.edu

Abstract: In today's digital landscape, securing wired network infrastructure is essential, particularly for Small and Medium Enterprises (SMEs) that often lack dedicated security personnel and resources. This project presents a fully automated, cost-effective method for detecting rogue switches, which are unauthorized devices that may be introduced by malicious insiders or external attackers to bypass network controls. Unmanaged (dumb) switches facilitate lateral movement, network sniffing, and long-term persistence, allowing adversaries to blend into legitimate traffic without detection. To counter this threat, the proposed system employs a correlation-based detection mechanism, supported by a three-layer validation model, to systematically verify infrastructure changes. The layered design reduces false positives and ensures accurate identification of unauthorized network modifications. Tailored for SME environments, this solution removes the need for manual inspection, offering a scalable, real-time response to rogue switch detection and mitigation.

Keywords: Rogue switches, Open vSwitch (OVS), Topology monitoring, ARP-scan, fping, MAC and ARP spoofing, SMTP alerts

1. Introduction

Wired Local Area Networks (LANs), though often perceived as inherently secure, are increasingly exposed to internal threats that bypass traditional perimeter defences. Among the most critical of these threats is the introduction of rogue switches, unauthorised Layer 2 devices silently connected to existing infrastructure. These unmanaged switches are typically plug and play, and lack visibility in network management systems.

Once connected, rogue switches enable adversaries to passively attach multiple devices, sniff traffic, or launch cyberattacks, all while blending into legitimate network activity. This allows attackers to bypass authentication controls, intercept sensitive data, and maintain persistent access within the network environment.

As enterprise networks grow in complexity, for SMEs with limited security infrastructure, the challenge of detecting such hidden threats becomes even more critical.

To address this, the proposed work introduces a structured rule-based detection framework that identifies rogue switches through layered validation focused on topology changes, device registration, and spoofing behavior. This real time, vendor neutral solution is scalable, lightweight, and does not require proprietary hardware, making it suitable for SMEs that need practical and reliable security mechanisms.

2. Related Work

Rogue switches in wired Local Area Networks (LANs) present a persistent challenge due to their passive nature and limited traceability. These unmanaged devices typically lack routable identities and do not participate in Layer 2 discovery protocols such as Link Layer Discovery Protocol (LLDP) or Cisco Discovery Protocol (CDP), rendering them invisible to conventional monitoring tools.

Several approaches have been proposed to infer their presence. Port-security controls such as BPDU Guard and PortFast are widely used to protect against unauthorized switches by disabling ports that receive unexpected Bridge Protocol Data Units (BPDUs) [1] (2024). However, these safeguards assume protocol participation. Unmanaged "dumb" switches neither emit BPDUs nor respond to LLDP or CDP, and can therefore forward traffic silently. In enterprise settings, comprehensive Network Access Control (NAC) commonly incorporates IEEE 802.1X [2] (2020) for port-based admission (EAP over LAN to a RADIUS backend), together with device profiling, posture assessment, and policy-based VLAN/ACL assignment. While NAC/802.1X is effective at deciding who may access a port and enforcing policy at admission time, it does not continuously validate post-admission physical changes or reveal hidden fan-out; a transparent unmanaged switch can still be placed behind an already authenticated port. The method proposed here is complementary; it supplies post-admission visibility that protocol-triggered controls and NAC do not provide.

Anomaly-driven studies have looked at access-segment signals and Layer 2 controls in lab settings. Bhuse et al. [3] (2025) combined management plane telemetry with Dynamic ARP Inspection, DHCP Snooping, Root Guard,

and Port Security with Sticky MAC in a GNS3 and Cisco setup, and added a supervised logistic regression model. The results were strong for ARP spoofing, MAC flooding, and STP root manipulation, but the pipeline was evaluation-focused, dependent on specific vendors, and not run as an always-on alerting service. Bhuse and James [1] (2020) used SNMP broadcast-storm traps as the initial signal, confirmed activity with ARP probes and announcements, and traced offenders using the router ARP table and switch CAM tables. Quitiquit and Bhuse [2] (2022) watched link state and MAC table activity to infer downstream insertion. In practice, these methods can miss short-lived events because of polling intervals, assume uniform management access and credentials across devices, and require manual playbooks or vendor-specific tooling. They also provide limited correlation across signals after admission, which can increase false positives or delay detection when changes are subtle or brief.

Most existing strategies address individual behaviors or require specific infrastructure. They do not offer a unified, vendor-neutral process that adapts across network conditions, especially in resource-constrained environments. This paper introduces a lightweight, rule-based detection framework that continuously validates physical topology changes, verifies registered devices, and detects spoofing behavior in real time. The rules and the correlation logic are configurable, so organizations can tune thresholds, whitelist known infrastructure, and add or remove checks to match their topology and policy. By maintaining a simple topology and MAC–IP baseline, performing active discovery from a trusted host, checking ARP consistency to spot impersonation, correlating signals to reduce noise, and emailing administrators immediately when something new or suspicious appears, the framework offers a practical and scalable option for SMEs.

3. Problem Statement

Existing solutions for detecting rogue switches often rely on proprietary protocols, manual tracing, or SNMP-based infrastructure, which are not always practical in real-world SME environments. Unmanaged switches that do not participate in discovery protocols can silently forward traffic, bypassing traditional security controls. In practice, budget constraints often lead to the deployment of low-cost, Fast-Ethernet unmanaged switches, further reducing protocol visibility and complicating detection. There is a need for a lightweight, rule-based detection framework that can identify such devices in real time without relying on complex infrastructure or protocol-level visibility.

4. Proposed Solution

The proposed framework enables real-time detection of rogue switches in wired LAN environments through a layered validation approach. It incorporates topology monitoring, device verification, and ARP cache analysis to detect unauthorized links, unregistered devices, and address spoofing. A trusted baseline of active authorized switch-port mapping and as well as MAC–IP mapping for Compliant (authorized) devices, is maintained and periodically compared against live network data. Upon identifying anomalies, the system triggers alerts via an SMTP-based notification mechanism.

To ensure long-term accuracy and compliance with organisational network governance standards, baseline updates are strictly manual and subject to predefined security policies. This approach minimises false positives while preserving the integrity of the detection process.

5. Implementation

The framework was implemented using Ubuntu, Mininet, Open vSwitch, and Bash scripting to define detection rules and automate topology monitoring. OVS (Open vSwitches) were monitored to detect topology changes and unauthorised links, while tools such as fping and arp-scan were integrated for active scanning and host discovery. ARP cache analysis was used to detect MAC and IP spoofing attempts. Postfix was configured to handle SMTP-based alerting. This approach is vendor-independent, lightweight, and suitable for real-time rogue switch detection..

5.1 Topology Overview

The test environment was configured to simulate a realistic enterprise LAN structure using Mininet's tree topology. The network consists of one core switch (s1) connected to four access-layer switches (s2 to s5). Each access switch is linked to four hosts, resulting in a total of sixteen hosts (h1 through h16). Specifically, s2 connects to h1 to h4, s3 to h5 to h8, s4 to h9 to h12, and s5 to h13 to h16. This topology emulates a structured multi-segment LAN, enabling detailed analysis of rogue switch insertion, spoofing behaviour, and automated alert

generation across various zones. Each detection rule was tested individually to evaluate the accuracy and responsiveness of the framework.

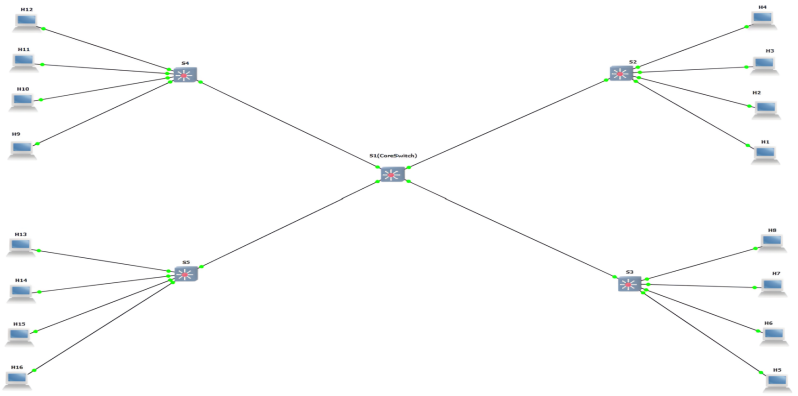

Figure 1a: Topology

5.2 Detection Rules Overview

Before introducing any rogue elements, the detection framework establishes a secure and trusted view of the network using three-layered rules. These rules work together to ensure that any unauthorised or malicious changes are promptly identified.

Rule 1: Topology Change Detection

Open vSwitch (OVS) is used to monitor the real-time status of switch ports. OVS listens to kernel-level network events such as cable insertions/removals, interface state changes, or system configuration updates. These events are reflected instantly in the OVS database and can be queried using ovs-vsctl show.

Once the network stabilises, a baseline snapshot of all switch-port mappings is taken and saved as baseline_topology.txt. This serves as a trusted reference, and the script periodically compares it against the current topology every few seconds to identify any unauthorised modifications (e.g., newly added ports from rogue switches to further connect multiple devices using a single legitimate switch port).

Open — baseline_topology.txt — ~/Desktop/Topology change script

```
1 Switch: s3
2   Port: s3-eth5
3   Port: s3-eth2
4   Port: s3-eth4
5   Port: s3-eth1
6   Port: s3-eth3
7 Switch: s2
8   Port: s2-eth2
9   Port: s2-eth4
10  Port: s2-eth1
11  Port: s2-eth3
12  Port: s2-eth5
13 Switch: s4
14  Port: s4-eth3
15  Port: s4-eth2
16  Port: s4-eth1
17  Port: s4-eth5
18  Port: s4-eth4
19 Switch: s1
20  Port: s1-eth4
21  Port: s1-eth2
22  Port: s1-eth3
23  Port: s1-eth1
24 Switch: s5
25  Port: s5-eth5
26  Port: s5-eth3
27  Port: s5-eth2
28  Port: s5-eth4
29  Port: s5-eth1
```

Figure 1b: Switch Active Ports (Baseline)

Rule 2: Device Registration and MAC Mapping

After establishing the baseline switch-port topology, the second rule focuses on identifying all active end devices and registering their corresponding MAC and IP addresses. This rule ensures that only legitimate hosts are recognised before the detection framework begins operation.

The system initiates a fast subnet-wide sweep using fping, which sends ICMP echo requests across the network to detect live hosts. Once responsive hosts are identified, tools like arp-scan or arping are used to send ARP

requests, forcing those hosts to reply with their MAC addresses. This process creates a reliable IP–MAC mapping for every connected device in the network.

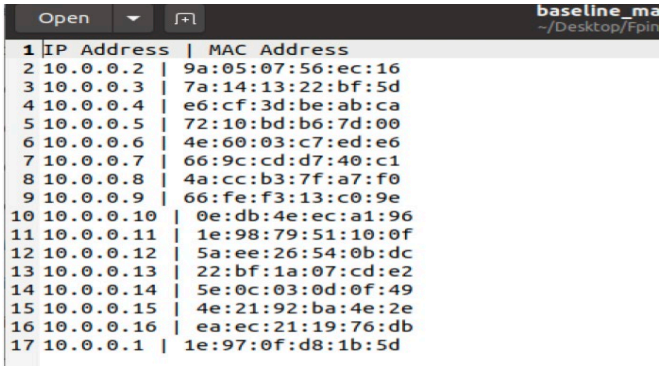

Figure 2: Trusted MAC–IP

To ensure trust and consistency, this scan is executed from a known legitimate host (h1), connected to access switch s2. Scanning from within a trusted network segment minimises the risk of spoofed replies or detection evasion by rogue nodes.

The resulting mappings are stored in a reference file named baseline_mac_ip.txt. This file acts as the foundation for all runtime MAC address verifications. During monitoring, any MAC address not found in this baseline is flagged as a potential rogue device.

The baseline can be updated when new, authorised devices are introduced. Each update must include the device's MAC address and its active switch port connection at the time of registration. For DHCP environments, detection keys are used for MAC-first matching. IP changes are tolerated, but MAC–IP conflicts and unknown MACs still trigger verification and alerts.

However, to ensure the security integrity of the framework, all baseline modifications must be manually approved in accordance with internal policies. This prevents accidental or malicious whitelisting of unauthorised devices.

During detection, any new MAC address not present in the baseline is treated with high priority, as it may indicate the presence of a rogue switch introducing new connected hosts. The severity of the alert increases if this MAC anomaly is accompanied by other signs of tampering, such as unexpected topology changes (Rule 1) or MAC/IP inconsistencies (Rule 3).

Rule 3: ARP and MAC Spoofing Detection

The third rule addresses spoofing attacks, where a malicious device may impersonate legitimate hosts by forging MAC or IP addresses. This layer is critical for identifying stealthy rogue devices that are connected behind rogue switches to perform man-in-the-middle (MITM) attacks. Such devices often bypass port-based controls or baseline MAC mapping checks established in Rule 2 by mimicking legitimate network behavior.

To detect these threats, the framework periodically retrieves ARP cache data from multiple legitimate hosts distributed across different access switches (e.g., h1 on s2, h5 on s3, h9 on s4, and h13 on s5). While this multi-host perspective ensures broader visibility within the wired LAN, primary preference is given to h1, as it actively performs network scans and communicates with other devices more frequently. As a result, its ARP cache is typically updated faster, making it the most reliable source for identifying MAC–IP inconsistencies and detecting spoofing behavior in real time.

The script scans for the following anomalies every 10 seconds:

- **MAC Spoofing**: The same MAC address appears under multiple IP addresses.
- **ARP Spoofing**: A single IP address mapping to multiple MAC addresses.

These patterns are direct, vendor-neutral signs of identity impersonation that a trusted host can check quickly. They stay reliable in DHCP environments because we match on the MAC first, so normal IP changes don’t create noise while real spoofing still shows up. They also reflect how attackers or a rogue downstream switch can manipulate the network through ARP poisoning or MAC cloning to corrupt ARP caches and switch CAM tables,

redirect traffic, hide additional devices, and bypass port-level controls and device-registration policy, which is exactly what our correlation and alerting are built to catch in real time.

Alerts from this rule are treated as high priority, especially when correlated with topology or device mapping anomalies from Rules 1 and 2. The scanning interval may be adjusted based on the network environment, particularly in cases where IP assignment changes occur frequently due to device reconnections.

5.3 Insertion of Rogue Switch and Device

To simulate a real-world network compromise, a rogue Open vSwitch (OVS) switch was introduced into the running Mininet topology after the legitimate setup was fully stabilised. This rogue switch was externally connected to the access switch s3, imitating a physical insertion into a live LAN through an unused network port.

A rogue device was also introduced and linked to the rogue switch using a virtual Ethernet connection. This setup replicates an attacker connecting an unauthorised switch along with one or more hidden devices in order to blend into the network without being immediately detected.

The insertion was performed from outside the Mininet environment to reflect a realistic scenario where an attacker interacts with the underlying system directly, without Mininet's awareness. This method allows the detection framework to validate whether it can identify unauthorised topology changes and hidden device activity introduced through unmanaged connections.

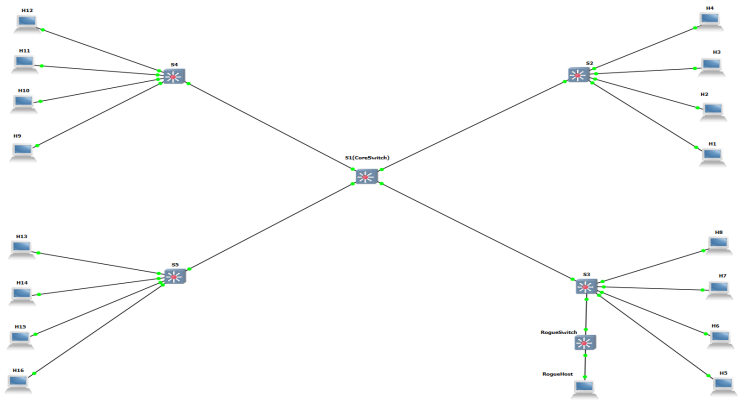

Figure 3: Topology after adding Rogue switch and device

5.4 Detection Based on Topology Changes

Once the framework is deployed, the topology monitoring script runs in the background and periodically checks for any new ports. When an unauthorized change, such as a rogue switch connection, is detected, the system immediately raises an alert. The screenshots below capture a real detection scenario where a new unauthorised port was identified.

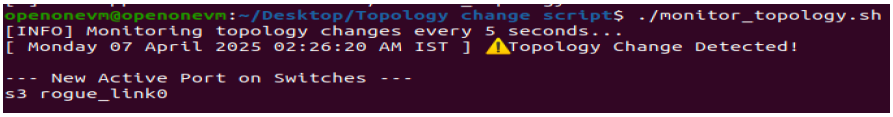

Figure 4: Real-Time Topology Change Detection Triggered

The script logs the moment a new port (rogue_link0) became active on switch s3, indicating a possible unauthorized link.

Figure 5: Email Alert Sent Upon Detection

An automated email alert was immediately triggered via SMTP, notifying the administrator of the detected rogue switch connection. The alert includes the exact timestamp and the specific switch-port details where the anomaly was observed, enabling precise identification and investigation.

5.5 Detection Rule Based on Unregistered Devices

Additionally, the second detection rule was running alongside the topology monitoring script. It prioritises MAC address verification. If a new, unregistered MAC address is detected, it is immediately flagged as rogue, regardless of its IP address. This approach is particularly important in DHCP-based networks, where IP addresses may change frequently but MAC addresses remain stable and unique identifiers.

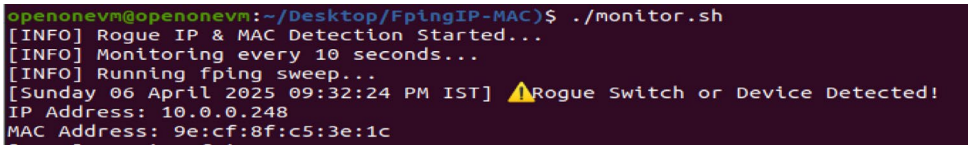

Figure 6: Rogue Device Detected via MAC Mapping

This screenshot shows the terminal output from the MAC detection script. A device with MAC address 9e:cf:8f:c5:3e:1c was identified at IP 10.0.0.248. Since this MAC was not listed in the baseline, it was flagged as a rogue device.

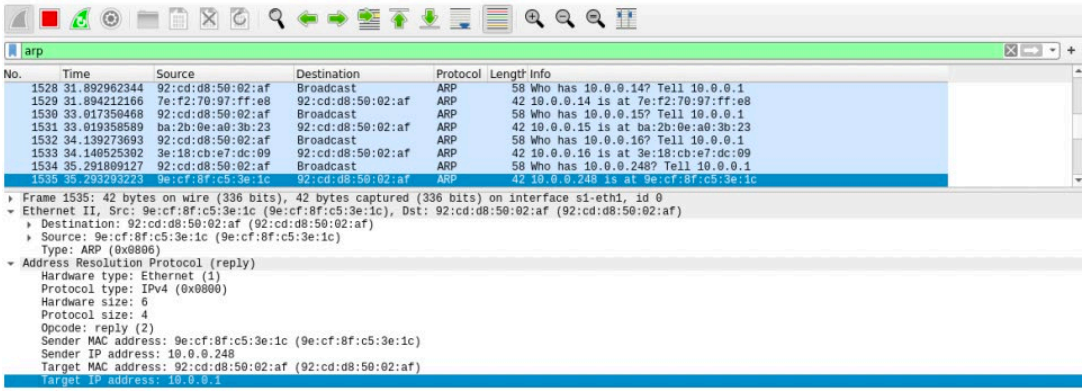

Figure 7: Wireshark Traffic Captured Near H1 host

This ARP reply packet confirms the MAC–IP mapping observed by h1 during baseline or live scan verification. In this frame, IP address 10.0.0.248 is associated with MAC address 9e:cf:8f:c5:3e:1c. Such entries are collected and compared against the trusted MAC–IP baseline in Rule 2 to identify unregistered or unauthorised devices. Any deviation from expected mappings triggers an alert, helping enforce device-level integrity within the LAN.

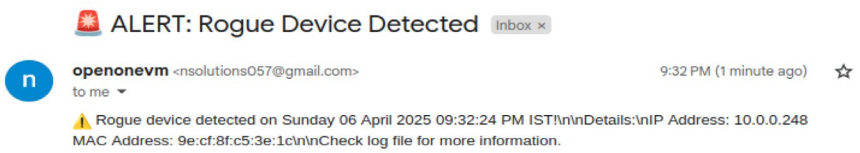

Figure 8: Email Alert for Topology Change

An automated email alert was immediately triggered via SMTP, notifying the administrator of the detected rogue switch connection. The alert includes the exact timestamp and identifies the affected port (rogue_link0) on switch s3 as observed by Rule 1. Additionally, Rule 2 raised a high-priority alert due to the appearance of an unregistered MAC address, further confirming the presence of a rogue device introduced via the rogue switch.

5.6 Detection Rule Based on MAC and ARP Spoofing

While topology changes and unregistered devices can expose many rogue setups, a determined attacker may still evade detection by impersonating known devices. This rule adds a critical layer of protocol-level inspection to uncover such spoofing behaviour.

5.6.1 MAC spoofing detection

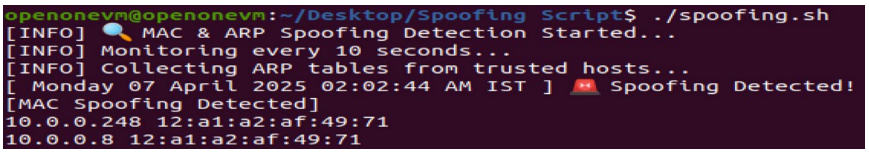

Figure 9: Spoofing Detection Triggered via MAC Conflict

In this scenario, a rogue device connected behind a rogue switch spoofed the MAC address of an already registered device. The detection script identified that the same MAC address (12:a1:a2:af:49:71) was simultaneously associated with two different IP addresses (10.0.0.248 and 10.0.0.8). Since this MAC was part of the trusted baseline, Rule 2 did not flag the device. However, Rule 3 successfully identified this as a MAC spoofing incident, which is an impersonation pattern commonly introduced by rogue switches to stay hidden. Given that these attack patterns are relatively common at Layer 2, there remains some scope for downstream evasion, which our Rule 3 is designed to mitigate.

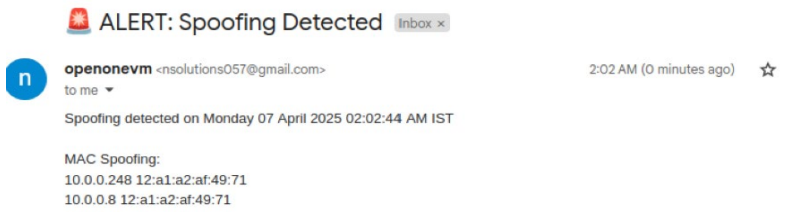

Figure 10: Email Alert Notifying Mac Spoofing Detection

An SMTP-based email alert was generated automatically, detailing the spoofed MAC address, the conflicting IPs, and the timestamp. This confirms that MAC spoofing can bypass earlier rules but still be caught by protocol-level analysis.

5.6.2 ARP spoofing detection

Although the device registration process in Rule 2 relies on MAC address verification, there still exists a narrow window for ARP spoofing, particularly when a rogue device impersonates a legitimate IP address with a different MAC. These cases are rare due to strict MAC filtering in Rule 2, but they can still occur when attackers reuse known IPs to bypass registration logic.

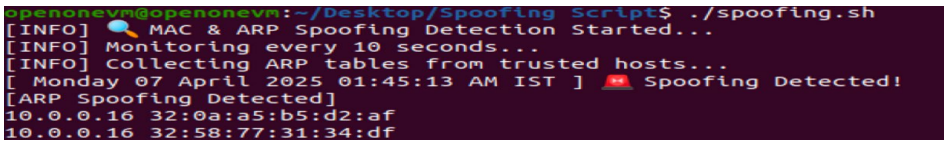

Figure 11: ARP Spoofing Detection Triggered

In this example, the spoofing script detected that the same IP address 10.0.0.16 was being claimed by two different MAC addresses: 32:0a:a5:b5:d2:af and 32:58:77:31:34:df. This is a typical indicator of an ARP spoofing attack, where the rogue host attempts to intercept traffic intended for a legitimate device.

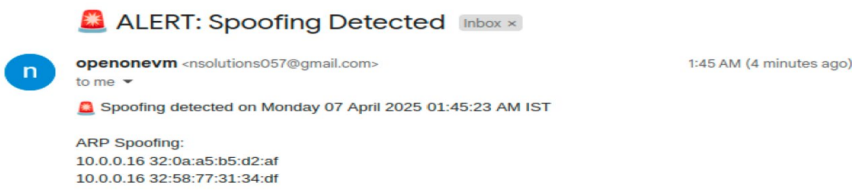

Figure 12: Email Alert Notifying ARP Spoofing Detection

An immediate alert was generated and sent to the administrator, showing the conflicting MAC addresses for the same IP. This helps in detecting impersonation attempts even when attackers try to reuse legitimate IPs.

5.7 Alert Prioritisation Logic

In dynamic environments, where legitimate devices may occasionally be moved or reconnected, a single topology change detected by Rule 1 is assigned medium priority by default. However, if that change is immediately followed by a MAC address mismatch (Rule 2) or an IP to MAC inconsistency (Rule 3), the alert level is escalated to high. When all three rules are triggered within the same observation window, the framework flags the event as critical, as it strongly suggests that a rogue switch is being used to connect multiple devices using a single legitimate switch port, indicating a highly suspicious or malicious activity. This layered prioritisation helps reduce false positives while ensuring timely and appropriate responses to compound threats.

6. Conclusion

This paper introduces a lightweight, real-time framework for detecting rogue switches in wired Local Area Networks by correlating topology monitoring, MAC address validation, and protocol-level spoofing analysis. The approach enables reliable identification of unauthorised switch insertions and downstream rogue devices through independent yet complementary detection rules, ensuring timely validation of network integrity. Targeted at small and medium-sized enterprises, the framework is vendor-neutral, cost-effective, and deployable without specialised hardware. Experimental evaluation confirms effective detection of topology anomalies, MAC spoofing, and ARP-based attacks, with automated email alerts supporting rapid administrative response. Future work will focus on visual topology dashboards, SIEM integration, and enhanced cross-rule correlation to further reduce false positives and strengthen post-admission network visibility.

Ethics declaration: Ethical clearance was not required for the research.

AI declaration: AI tools were not used the creation of this paper.

References

Bhuse, V., Kanipakam, V.P.R., Patil, R. and Wang, X., 2025, June. Detecting Rogue Switch and Device Behaviour Using Network Anomalies in LAN. In European Conference on Cyber Warfare and Security (pp. 42-51). Academic Conferences International Limited.

Cisco (2024) Cisco Discovery Protocol (CDP). Cisco Learning Network. Available at: https://learningnetwork.cisco.com/s/article/cisco-discovery-protocol-cdp-x (Accessed: 18 March 2025)

Rigney, C., Willens, S., Rubens, A. and Simpson, W. (2000) RFC 2865: Remote Authentication Dial In User Service (RADIUS). IETF. Available at: https://datatracker.ietf.org/doc/html/rfc2865 (Accessed: 11 February 2025)

Droms, R. (1997) RFC 2131: Dynamic Host Configuration Protocol (DHCP). IETF. Available at: https://datatracker.ietf.org/doc/html/rfc2131 (Accessed: 12 February 2025)

Cisco (2024) Link Layer Discovery Protocol (LLDP). Cisco Learning Network. Available at: https://learningnetwork.cisco.com/s/article/link-layer-discovery-protocol-lldp-x (Accessed: 18 March 2025).

Cisco (2025) Spanning Tree Protocol (STP). Cisco Systems. Available at: https://www.cisco.com/c/en/us/tech/lan-switching/spanning-tree-protocol/index.html (Accessed: 19 March 2025)

Bhuse, V., Kalafut, A. and Dohn, L., 2019. Detection of a Rogue Switch in a Local Area Network. In The Fourteenth International Conference on Internet Monitoring and Protection. ICIMP.

Prins, K. and Bhuse, V., 2018, June. Forced vacation: A rogue switch detection technique. In European Conference on Cyber Warfare and Security (pp. 390-399). Academic Conferences International Limited.

Quitiqut, T. and Bhuse, V., 2022, March. Utilizing Switch Port Link State to Detect Rogue Switches. In International Conference on Cyber Warfare and Security (Vol. 17, No. 1, pp. 272-278).

Bhuse, V. and James, V., 2020, June. Detecting a Rogue Switch using Network Automation. In ECCWS 2020 19th European Conference on Cyber Warfare and Security (p. 26). Academic Conferences and publishing limited..

Bhuse, V., Vellaboina, Y. and Wang, X., 2024, June. Enhancing Network Security: Rogue Switch Detection and Prevention in Local Area Network. In European Conference on Cyber Warfare and Security (Vol. 23, No. 1, pp. 66-73).

Udhayan, J. and Thilagam, P.S. (2021) 'A survey on MAC address spoofing attacks and their detection techniques', International Journal of Security and Networks, 16(1), pp. 1–10.

IEEE (2020) IEEE Std 802.1X-2020: Port-Based Network Access Control. IEEE Standards Association. Available at: https://standards.ieee.org/ieee/802.1X/7345/ (Accessed: 8 November 2025).

Trusted Computing Group (2017) TNC Architecture for Interoperability Specification, Version 2.0, Revision 13. Trusted Network Connect Work Group. Available at: https://trustedcomputinggroup.org/wp-content/uploads/TCG_TNC_Architecture_v2.0r13.pdf (Accessed: 11 November 2025).

Kazienko, P., Choraś, M. and Kozik, R. (2020) 'Network layer threats and mitigation techniques: A study of ARP spoofing in LANs', Journal of Cyber Security and Mobility, 9(2), pp. 129–146.

Linux man-pages (2024) arping(8) – Linux manual page. Available at: https://linux.die.net/man/8/arping (Accessed: 4 March 2025).

Haardt, R. and Oetiker, T. (2023) fping Documentation. Available at: https://fping.org (Accessed: 1 March 2025).

GNU (2022) Bash Reference Manual. Available at: https://www.gnu.org/software/bash/manual/bash.html (Accessed: 2 March 2025).

Robbins, A. (2023) The GNU Awk User's Guide. Free Software Foundation. Available at: https://www.gnu.org/software/gawk/manual/gawk.html (Accessed: 28 February 2025).

GNU Diffutils (2022) Diffutils Manual. Available at: https://www.gnu.org/software/diffutils/manual (Accessed: 26 February 2025).

Mininet (n.d.) Mininet network emulator. Available at: https://mininet.org (Accessed: 15 February 2025)

Open vSwitch (n.d.) Open vSwitch: Production quality, multilayer virtual switch. Available at: https://www.openvswitch.org (Accessed: 17 February 2025).

Wireshark (n.d.) 'Wireshark Network Protocol Analyzer'. Available at: https://www.wireshark.org (Accessed: 20 February 2024).