STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, February 2026, pp 1151–1161.
Published online in International Academic Press (www.IAPress.org)

Towards the application of Process Mining for analyzing Network Security

Salah-Eddine SAMIRI [1,*], Zineb LAMGHARI [2], Rachid FAKHAR [1]

[1] *LS2ME Laboratory, Sultan Moulay Slimane University of Beni Mellal, Polydisciplinary faculty of Khouribga, Morocco*
[2] *Department of Computer Science, Faculty of Sciences, Mohammed V University in Rabat, Rabat 10000, Morocco*

Abstract The objective of our study is to investigate the use of process mining algorithms in analyzing data collected from a Man-in-the-Middle (MitM) attack that we simulated in a controlled local network to detect malicious activities. After analyzing the data we found results that indicate that under normal network conditions the BPMN model displays standard interactions without duplicate or conflicting IP addresses. However, in the case of MitM attack, the model shows duplicate and conflicting IP addresses which was a real manipulation and disruption. These results highlight how process mining can improve forensic investigations into network security intrusions.

Keywords Process Mining, Cybersecurity, Network Security, Forensic, MitM attack

DOI: 10.19139/soic-2310-5070-2557

1. Introduction

For companies or individuals, cybersecurity has become important today because of the evolution of digital technologies and the increasing connectivity of networks. Attacks have become more complex and sophisticated, which represents major risks for information systems. In recent years, process mining has emerged as a new approach used to improve cybersecurity data analysis. Process mining is intended to extract knowledge from event logs that have been generated by information systems [2] and among the characteristics of this approach is to visualize, monitor and improve processes while providing precise information on the behaviors of the systems. by applying process mining techniques to cybersecurity, network administrators will be able to have a deep understanding of network interactions and identify cyber threats in real time.

In this paper, we applied process mining algorithms on the event logs of a simulated man-in-the-middle (MitM) attack in a local area network to evaluate the performance of process mining techniques in detecting malicious activities. What makes our approach different from other traditional cybersecurity tools is that the latter often rely on predefined signatures or anomaly detection algorithms, except that our approach uses process mining to analyze the event logs.

From the results of our study we can confirm that process mining is able to identify security vulnerabilities and improve forensic investigations through the analysis of event logs by giving a visual representation of attack paths that is easy to interpret, which is not the case with current methods.

The remainder of this paper is organized as follows: Section 2 provides a preliminary overview. Section 3 presents a review of the relevant literature. Section 4 introduces the proposed framework. Section 5 outlines the

*Correspondence to: Salah-Eddine SAMIRI (Email: samirisalah60@gmail.com). LS2ME Laboratory, Sultan Moulay Slimane University of Beni Mellal, Polydisciplinary faculty of Khouribga, B.P. 145, 25000 Khouribga, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

methodology employed in our case study. Section 6 presents the case study, followed by a discussion in Section 7, and Section 8 concludes the paper and suggests future research directions.

2. Preliminaries

This section we will cover process mining techniques, event logs, and process representation through mathematical models.

2.1. Process Mining Techniques

Process Mining is an analysis method based on the exploitation of event logs to analyze and improve business processes. Below are the main Process Mining techniques [13]:
Process Discovery: This technique consists of extracting the process model from an event log, identifying the relationships between activities and the actual flow of the process and the result is a process model P derived from an event log L:

$$P = D(L)$$

D: Represents the discovery algorithm used on the L event log.
Conformance Checking: This method is based on comparing the behavior recorded in the event log with a pre-established process model by evaluating the extent to which the observed behavior conforms to the expected behavior. The conformance score C is defined as follows:

$$C(P, L) = \frac{\text{Number of matching traces}}{\text{Total traces in L}}$$

Enhancement: This technique allows to optimize the existing process model by relying on additional information. The improved model P' is generated from an initial model P' and an event log L:

$$P' = E(P, L)$$

Where E represents the enhancement function.

2.2. Event Logs

Event logs are the source of information for process mining. An event log L is described as a set of traces, each representing a distinct case:

$$L = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

Each trace σ_i is composed of a series of events, where an event e is defined by its properties, typically including the activity name, timestamp, and resource:

$$e = (a, t, r)$$

where:
– a: activity,
– t: timestamp ,
– r: resource utilized during the event.

2.3. Process Models

Process models are visual diagrams of processes based on event logs.
Petri Nets: A Petri net N is defined formally as a tuple (P, T, F), where:
– P: set of places
– T: set of transitions
– F: set of directed arcs connecting places and transitions
BPMN: Is a schematic notation for representing business processes.

$$G = (V, E)$$

where:
– V: set of vertices (activities, events, gateways)
– E: set of edges representing the flow between activities

3. Literature Review

Following the great development of information technology and network infrastructure, today there is a strong demand to secure these assets, which has prompted researchers to focus on the integration of process mining in cybersecurity to improve security measures. In this literature review we present several researches that have discussed the use of process mining in cybersecurity.

3.1. Process Mining in Cybersecurity

Van der Aalst et al. [1] used process mining to analyze security logs and discover unusual patterns that contain cyber threats and spot deviations in network behavior, which provides a useful basis for threat mitigation. Their results show that process mining can be useful for detecting and responding to cybersecurity incidents.

3.2. Process Mining in Network Security

Several researches have discussed the application of process mining in network security, especially in threat analysis and detection. To simplify the analysis of large amounts of intrusion detection alerts and discover attack strategies S. Alvarenga [6] applied process mining algorithms to intrusion detection system (IDS) alert records to have user-friendly visual models that are easy to interpret by network administrators. To identify unauthorized access patterns and detect security vulnerabilities in networks Zafar et al. [11] used discovery algorithms. According to the results of both researches, process mining could improve and facilitate the detection of security vulnerabilities and monitor network activities.

3.3. Conformance Checking

To classify event log traces that may indicate security vulnerabilities such as fraud, Fazzinga et al [12] presented a methodology based on process discovery to develop a vulnerability detection model and then classifies new traces via a conformance check.

3.4. AI-based intrusion detection

Intrusion detection systems (IDS) have seen significant development due to the integration of artificial intelligence which improves their performance in intrusion detection. Studies have discussed the benefit of using machine learning to improve the accuracy and clarity of IDS [15, 16]. To assess the effectiveness of AI algorithms in detecting network threats, Naseem et al [17] conducted a comparative study of AI models to identify their difficulties and overcoming them to enhance their effectiveness against threats. According to these researches, integrating AI into intrusion detection has become important to improve intrusion detection.

4. Methodology

To apply process mining techniques to network security analysis, we used a methodology divided into five steps (Figure 1). Each step was necessary to transform raw network data into information that could be used to detect and analyze network attacks.

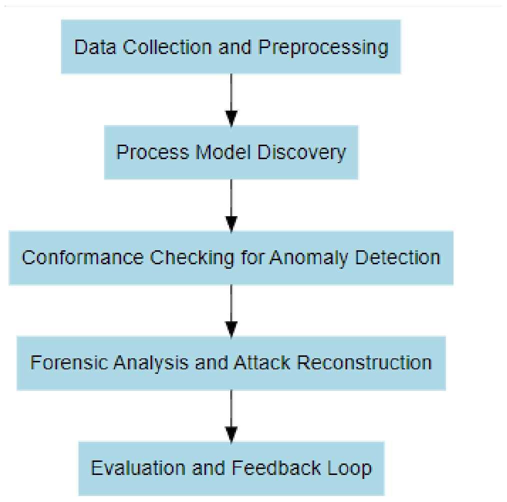

Data Collection and Preprocessing
Process Model Discovery
Conformance Checking for Anomaly Detection
Forensic Analysis and Attack Reconstruction
Evaluation and Feedback Loop

Figure 1. Methodology

Data Collection and Preprocessing: Network traffic data is obtained from various monitoring devices such as intrusion detection systems, firewalls and software such as waresharke captures key details such as IP addresses, timestamps and communication protocols (e.g. ARP, TCP, UDP). Data is cleaned to eliminate noise, standardize log formats and deal with missing values. Event logs are then structured according to network sessions, with each session representing a case and actions representing activities.
Process Model Discovery: Discovery algorithms, such as the Inductive Miner algorithm, use a recursive divide-and-conquer approach to build robust, structured process trees by dividing event logs into smaller parts based on event dependencies and processing each part separately. The algorithm is capable of handling different workflow models including sequences, loops, exclusive choice XOR and AND parallelism considering data noise management mechanisms [14].
To create Petri net models from event logs we used the inductive Miner algorithm to analyze the event logs collected under normal conditions and during ARP Spoofing attack and then transformed the Petri net models into BPMN diagrams to have a clear visualization of activity flows in the network. However, in the preprocessing step we used other process discovery algorithms such as the α-algorithm and Heuristics Miner. The α-algorithm was used to derive Petri nets to identify causal links between events and Heuristics Miner was used to detect frequent sequences and dependencies in the logs. Through the use of these algorithms, we were able to create interpretable process models that help us identify deviations caused by ARP spoofing attacks.

Conformance Checking for Anomaly Detection: Anomalies such as IP and MAC address conflicts are detected by comparing the behavior of attack models with the models under normal condition. Additionally, metrics such as fitness and simplicity allow us to measure deviations.

$$A(P_{\text{baseline}}, L_{\text{observed}}) = \frac{\text{Conforming traces}}{\text{Total traces}}$$

Forensic Analysis and Attack Reconstruction: To determine the attack patterns it will be necessary to analyze the anomalies (IP and MAC address conflicts) that were detected during the Compliance Checking step.

Evaluation and Feedback: Model performance is assessed using metrics including processing time, accuracy, and false positive rates [13].
mathematical definitions:
- Accuracy:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Where TP : True Positives, TN : True Negatives, FP : False Positives, and FN : False Negatives.
- False Positive Rate (FPR):

$$FPR = \frac{FP}{FP + TN}$$

- Processing Time:

$$T_{\text{processing}} = \frac{\sum_{i=1}^{N} t_i}{N}$$

Where t_i :Time for the i-th process, N : Total number of processes.

5. Case Study: MITM Attack

5.1. *The Definition of MITM Attacks*

A man-in-the-middle (MITM) attack is a form of cyberattack in which an intruder surreptitiously monitors and tampers with the communication between a client and a server that believe they are communicating directly with each other [7]. This form of attack compromises the privacy of shared data, as the attacker has the ability to manipulate the information being transmitted. There are several forms of MitM attacks, including ARP spoofing, DNS spoofing, and SSL/TLS interception [8].

We chose the ARP spoofing attack because of its impact on information systems communication and network security.

5.2. *Setting Up the Environment*

To simulate and study the ARP spoofing attack, we used various software tools like Wireshark to capture network traffic [9], Arpspoof command to perform the attack, and ProM for process mining analysis. To perform the attack simulation, we deployed an Ubuntu virtual machine (VM) to conduct the attack. Additionally, using a smartphone, we shared Wi-Fi to establish a connection between the Windows PC and the VM. We started Wireshark on the VM to record the packets exchanged between the client and the server, and then used arpspoof [10] to spy on the network. Finally, we used ProM to examine and create process models from the filtered event logs.

One major challenge we faced was filtering out the relevant packets in Wireshark due to the high volume of background network activity, which made isolating attack data tricky. Another hurdle was keeping the event logs synced with the attack execution since timing mismatches occasionally led to inconsistencies in our analysis. To address these issues, we refined Wireshark's capture filters and performed multiple test runs to enhance the accuracy of our data.

5.3. *Ethical Considerations*

To ensure that no unauthorized access to sensitive information occurs, we conducted our study in a controlled test environment and the data collected during the study was limited to test situations, with no real user data involved. To protect confidentiality, all captured packets were sorted to eliminate sensitive details. Furthermore, in our study we followed the ethical guidelines for cybersecurity research and that ARP spoofing simulation was only used for analysis purposes and not for malicious purposes.

5.4. ARP Spoofing attack

The diagram represents the attack scenario we have carried out in this case study, where the attacker disrupts communication between client and server by manipulating ARP cache entries on both systems (Fig.2).

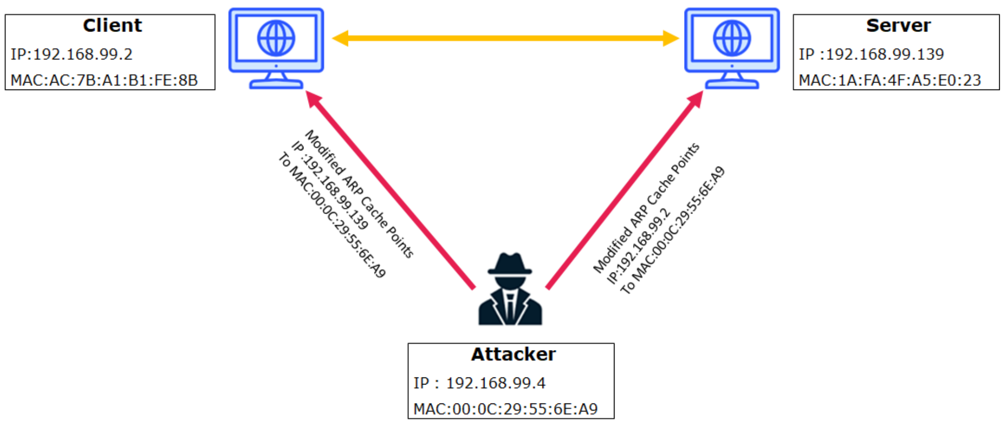
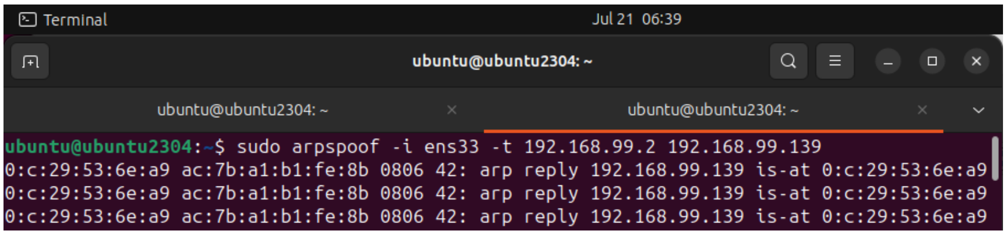


Figure 2. ARP Spoofing Attack

To execute the ARP spoofing attack, we kicked things off by running Wireshark to capture all network traffic on the interface [ens33]. This step allowed us to log every packet exchanged between the client and server for future analysis. Next, we launched the MitM ARP Spoofing attack using arpspoof as shown in (Fig. 3). This attack tricked both the client and the server into identifying the attacker's machine as the network gateway, which we accomplished with the following command:



```
ubuntu@ubuntu2304:~$ sudo arpspoof -i ens33 -t 192.168.99.2 192.168.99.139
0:c:29:53:6e:a9 ac:7b:a1:b1:fe:8b 0806 42: arp reply 192.168.99.139 is-at 0:c:29:53:6e:a9
0:c:29:53:6e:a9 ac:7b:a1:b1:fe:8b 0806 42: arp reply 192.168.99.139 is-at 0:c:29:53:6e:a9
0:c:29:53:6e:a9 ac:7b:a1:b1:fe:8b 0806 42: arp reply 192.168.99.139 is-at 0:c:29:53:6e:a9
```

Figure 3. Cmd: sudo arpspoof -i ens33 -t 192.168.99.2 192.168.99.139

This command effectively redirected the network traffic between the client (192.168.99.2) and the server (192.168.99.139) through the attacker's machine, allowing interception and potential modification of the communication. During the attack, Wireshark continued to capture the network traffic, providing detailed logs of ARP communications.

5.5. Filtering and Processing Relevant Events

Following the collection of event logs using Wireshark, which captured various protocols (DNS, ARP, TCP, HTTP, etc.), we focused specifically on the ARP protocol for our case study. Consequently, we filtered out only ARP transmissions for detailed analysis. To process the attack data, we have applied Process Mining techniques to create process models and carry out conformance checking to detect attacks and deviations. For this purpose, we employed the Inductive Miner algorithm to obtain Petri net models, which we then converted to BPMN models (Fig. 4).

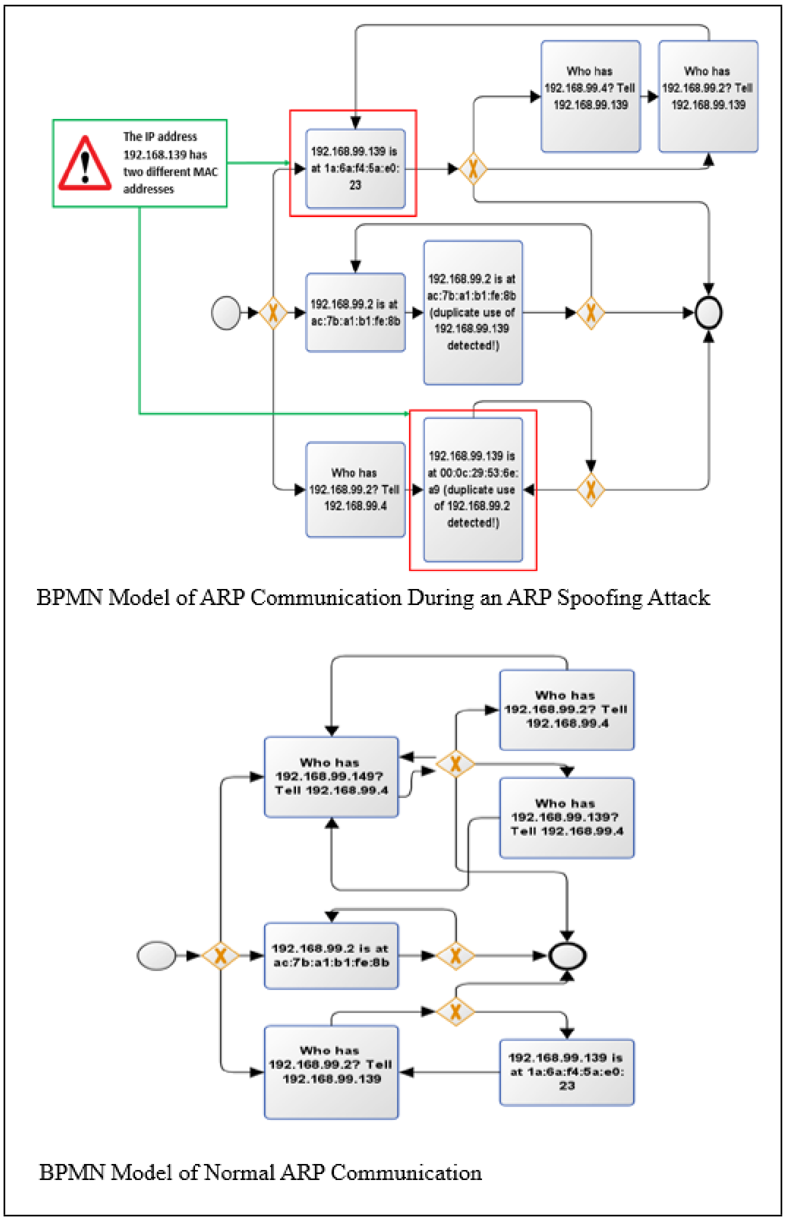

The IP address 192.168.139 has two different MAC addresses
192.168.99.139 is at 1a:6a:f4:5a:e0:23
Who has 192.168.99.4? Tell 192.168.99.139
Who has 192.168.99.2? Tell 192.168.99.139
192.168.99.2 is at ac:7b:a1:b1:fe:8b
192.168.99.2 is at ac:7b:a1:b1:fe:8b (duplicate use of 192.168.99.139 detected!)
Who has 192.168.99.2? Tell 192.168.99.4
192.168.99.139 is at 00:0c:29:53:6e:a9 (duplicate use of 192.168.99.2 detected!)
BPMN Model of ARP Communication During an ARP Spoofing Attack
Who has 192.168.99.2? Tell 192.168.99.4
Who has 192.168.99.149? Tell 192.168.99.4
Who has 192.168.99.139? Tell 192.168.99.4
192.168.99.2 is at ac:7b:a1:b1:fe:8b
Who has 192.168.99.2? Tell 192.168.99.139
192.168.99.139 is at 1a:6a:f4:5a:e0:23
BPMN Model of Normal ARP Communication

Figure 4. BPMN Models of ARP Communication

5.6. Results

Analysis of the ARP logs collected using Wireshark and processed with the ProM process mining tool allowed us to generate BPMN process models related to the MitM attack on the ARP protocol. The BPMN diagram (Fig. 4) represents several key interactions between IP addresses and MAC addresses collected during the ARP spoofing attack.
1) Initial ARP Request: The request Who has 192.168.99.2? Tell 192.168.99.4 represents a legitimate ARP query seeking the MAC address of IP:192.168.99.2.
2) Attacker's ARP response: The response 192.168.99.139 is at 1A:FA:4F:A5:E0:23 (duplicate use of 192.168.99.2 detected!) shows an incorrect MAC address attached to the IP address 192.168.99.139, indicating an attempt to

redirect traffic.
3) IP Address Collision: The entry 192.168.99.2 is at AC:7B:A1:B1:FE:8B (duplicate use of 192.168.99.139 detected!) indicates that the target machine IP: 192.168.99.2 detects an IP address collision, a clear sign of an ongoing ARP spoofing attack.
4) Unusual ARP Responses: The responses 192.168.99.139 is at 00:0C:29:55:6E:A9 and 192.168.99.139 is at 1A:FA:4F:A5:E0:23 show different MAC addresses for the same IP:192.168.99.139, indicating the attacker's attempt to impersonate the gateway.
In this context, the incorrect information refers to the MAC addresses associated with legitimate IP addresses that have been altered by the attacker to redirect network traffic through their machine.
1) For the IP address Server[192.168.99.139] the legitimate response should be 192.168.99.139 is at 00:0C:29:55:6E:A9 indicating the correct MAC address corresponding to the IP address 192.168.99.139. However the attacker issues a falsified response of 192.168.99.139 is at 1A:FA:4F:A5:E0:23, which displays an incorrect MAC address for 192.168.99.139. By doing so, the attacker misdirects traffic meant for the legitimate device at 192.168.99.139 towards their machine.
2) In the case of IP address Client [192.168.99.2], an abnormal and repeated response appears: 192.168.99.2 is at AC:7B:A1:B1:FE:8B (duplicate use of 192.168.99.139 detected!) shows that the target machine detects an incorrect MAC address associated with IP:192.168.99.2, which can further disrupt network communication. As a result, traffic that should be destined for IP:192.168.99.2 may be misrouted or incorrectly redirected, exacerbating the interference caused by the attacker.

In the absence of an attack, (Fig. 4) illustrates the normal behavior of ARP interactions, which follow a standard pattern without showing any duplications or IP address conflicts. In a typical ARP interaction, the process begins with an ARP request, such as Who has 192.168.99.2? Tell 192.168.99.4. This request is sent out by a device on the network, querying for the MAC address associated with the IP address 192.168.99.2. The device making the request is identified by its IP address 192.168.99.4. Following the request, the appropriate response is issued by the device that owns the IP:192.168.99.2, 192.168.99.2 is at AC:7B:A1:B1:FE:8B. This response provides the MAC address AC:7B:A1:B1:FE:8B of the device with IP:192.168.99.2, allowing the querying device to map the IP address to the correct MAC address on the network.
In order to better understand the performance and effectiveness of the process mining approach in detecting the ARP spoofing attack, we present the following metrics:

Size of Data Used:
The dataset used for this analysis contains 2,000 ARP packets captured using Wireshark during the simulated MitM attack. These packets include both legitimate ARP communication and spoofed responses from the attacker.

True Attacks Identified:
Out of the 2,000 ARP requests captured, 150 packets were identified as representing true attack attempts based on the inconsistencies detected in the ARP responses (e.g., duplicate IP/MAC addresses and multiple conflicting MAC addresses for the same IP).

Detection Rate:
Our process mining tool, ProM, was able to successfully detect 85% of the true attacks based on the anomalies in the BPMN model. These anomalies, such as duplicate IP addresses or MAC address collisions, were key indicators of the ongoing ARP spoofing attack.

Missed Attacks:
Despite the high detection rate, 15 true attacks were missed. These missed attacks typically involved subtle changes in the ARP responses, which the current model did not flag as anomalies. This points to potential areas of improvement in the detection algorithm, such as adjusting thresholds for anomaly detection or incorporating additional features of ARP communications for better sensitivity.

False Positives:
The number of legitimate ARP response was 20, that marked as suspicious. These cases were mainly caused by abnormal but valid network configurations that led to BPMN model anomalies. This needs to filter out false positives by fine-tuning the process model to distinguish between legitimate network activity and attacks. The results we found confirm that process mining is effective in identifying the main ARP spoofing attacks, there are areas where the model can be improved to reduce missed attacks and false positives.

5.7. Conformance checking

To compare the fitness and simplicity metrics between the normal case and the ARP spoofing attack case, we used log replay on the Petri net plugin for compliance checking. For the normal case, the fitness score was 0.9166, which indicates that most of the event logs closely followed the expected process, with few deviations and the simplicity metric shows that the process model is relatively simple, with fewer states and transitions, making it easier to understand and more consistent with normal ARP communication flow.
In the attack case, the fitness score fell to 0.8599, indicating more deviations between actual events and the expected process, and simplicity was also affected, as the number of states and transitions increased significantly due to the additional abnormal events generated by the ARP spoofing attack. This made the model more complex to follow.
The conclusion of the fitness and simplicity measures evaluation proves that the ARP spoofing attack reduced the alignment of the event logs with the expected process (lower fitness), and also made the process model more complicated and less simple (lower simplicity).

6. Discussion

Our study aims to detect malicious activities of ARP Spoofing attack through event log analysis using process mining. By analyzing BPMN model generated during the attack we observed a series of duplicate and conflicting IP addresses, which are clear signs of malicious manipulation. This proves that the attacker injected incorrect information into the network and the injection led to repeated ARP requests that disrupted the communication between the server and the client. Additionally, detecting MAC address conflicts was a critical sign because we know that ARP spoof corrupts ARP tables.
In the conformance checking analysis we compared two metrics, fitness and simplicity under normal conditions and during the ARP spoofing attack and as a result the ARP spoofing attack made the process model more complex and less simple (lower simplicity) and also reduced the alignment between event logs and the expected process (lower fit).
The results found can help network administrators in the future to configure automated alerts based on the metrics of conformance checking to trigger alerts and implement corrective measures and also thanks to the visual presentation of BPMN models that allows us to identify IP and MAC address conflicts can facilitate the diagnosis for network administrators.
To evaluate our approach we compared its behavior with other methods for detecting ARP spoofing attacks. Table 1 summarizes the results of the comparison, Taking into account various aspects such as detection accuracy, false positive rate, processing time, adaptability, and traceability.

Table 1. Comparison of Methods for Detecting ARP Spoofing Attacks

Method	**Detection Accuracy**	**False Positive Rate**	**Processing Time**	**Adaptability**	**Attack Traceability**
Proposed (Process Mining)	85%	Low	Moderate	High	High
Signature-Based	75%	Moderate	Fast	Low	Low
Statistical Analysis	80%	High	Fast	Low	Low
Machine Learning	92%	Low	High	Moderate	Moderate

Table 1 highlights the strengths and weaknesses of the process mining approach compared to other methods. For example, for the detection accuracy metrics the process mining method (85%) exceeds that of signature-based and statistical methods because it is able to do dynamic analysis of ARP event logs. However, machine learning approaches also guarantee high accuracy, but require large computational resources and training data.

The table also shows the advantage of the process mining approach in terms of adaptability and attack traceability. Unlike other methods, process mining interactively displays variations in network processes, providing exploitable information to network administrators. On the other hand, its average processing time indicates that it is more suitable for post event forensic analysis rather than real-time detection. This comparative study demonstrates the strength of process mining in detecting complex and evolving attack patterns such as ARP spoofing.

7. Conclusion

This paper presents a study on processing MITM attack event logs using process mining techniques to evaluate its effectiveness in detecting malicious activities. In this study we were able to identify security vulnerabilities such as conflicting IP and MAC addresses from the process models which help us to detect the ARP spoofing attack. From the results obtained, we believe that process mining techniques can improve cybersecurity measures, since most information systems are capable of generating event logs. If we can exploit this information, we will be able to improve incident response and strengthen network defenses. Although integrating process mining into forensic investigations can be an effective tool to improve the protection of critical infrastructures, this method has some weaknesses such as processing time and occurrence of false positives, these weaknesses make it less suitable for real-time detection and less reliable in some applications. These issues highlight the need to improve both speed and accuracy. Our future research will focus on anomaly detection using machine learning, particularly unsupervised models such as autoencoders and clustering techniques to improve process mining models. The integration of real-time monitoring via advanced network traffic analysis will be favored to better identify protocol anomalies, synchronization deviations and unauthorized access patterns.

REFERENCES

1. W. van der Aalst, Ed., Berlin, Heidelberg. *Process Mining: Data Science in Action*, Springer, 2016, pp. 3–23. doi: 10.1007/978–3–662–49851–4–1.
2. IBM, *What is Process Mining?*, Accessed: Jul. 20, 2024. [Online]. Available: https://www.ibm.com/topics/process-mining.
3. ISACA *Understanding and Implementing a Culture of Cybersecurity*, ISACA. Accessed: Jul. 20, 2024. [Online]. Available: https://www.isaca.org/resources/news-and-trends/industry-news/2019/understanding-and-implementing-a-culture-of-cybersecurity.
4. Kissel, R, *Glossary of Key Information Security Terms*, NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.IR.7298r2.
5. J. T. Force *Security and Privacy Controls for Information Systems and Organizations*, National Institute of Standards and Technology, NIST Special Publication (SP) 800-53 Rev. 5, Dec. 2020. doi: 10.6028/NIST.SP.800-53r5.
6. S. Alvarenga, B. Bogaz Zarpelão, S. Barbon Junior, R. Miani, and M. Cukier, *Discovering Attack Strategies Using Process Mining*, 2015. doi: 10.13140/RG.2.1.4524.4008
7. IEEE Xplore, *A Survey of Man In The Middle Attacks*, IEEE Journals and Magazine . Accessed: Jul. 20, 2024.[Online]. Available: https://ieeexplore.ieee.org/abstract/document/7442758.
8. B. Bhushan, G. Sahoo, and A. K. Rai, *Man-in-the-middle attack in wireless and computer networking — A review*, in 2017 3rd International Conference on Advances in Computing,Communication and Automation (ICACCA) (Fall), Sep. 2017, pp. 1–6. doi: 10.1109/ICACCAF.2017.8344724
9. U. Banerjee, A. Vashishtha, and M. Saxena, *Evaluation of the Capabilities of WireShark as a tool for Intrusion Detection*, International Journal of Computer Applications, vol. 6, no. 7, pp. 1–5, Sep. 2010.
10. *How to Perform an ARP Poisoning Attack*, Accessed: Jul. 20, 2024. [Online]. Available: https://www.stationx.net/how-to-perform-an-arp-poisoning-attack/.
11. Zafar, M. R., Saleem, S., Hussain, W., and Ullah, K *A Review on Internet of Things (IoT): Security and Privacy Requirements and Solutions*, (2018). . IEEE Access, 6, 11785-11795.
12. B. Fazzinga, F. Folino, F. Furfaro, L. Pontieri, *Combining Model- and Example-Driven Classification to Detect Security Breaches in Activity-Unaware Logs*, Springer International Publishing,Cham, pp. 173–190, 2018.
13. W. M. P. Van Der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Busi-ness Processes*, Berlin, Heidelberg: Springer, 2011. doi: 10.1007/978-3-642-19345-3.

14. I. Nuritha dan E. R. Mahendrawathi, *Structural Similarity Measurement of Business Process Model to Compare Heuristic and Inductive Miner Algorithms Performance in Dealing with Noise*, Procedia Computer Science, vol. 124, pp. 255-263, 2017.
15. Sowmya T. Mary Anita E.A, *A comprehensive review of AI based intrusion detection system*, https://doi.org/10.1016/j.measen.2023.100827.
16. Naseem Khan, Kashif Ahmad, Aref Al Tamimi, Mohammed M. Alani, Amine Bermak, Issa Khalil, *Explainable AI-based Intrusion Detection System for Industry 5.0: An Overview of the Literature, associated Challenges, the existing Solutions, and Potential Research Directions*, arXiv:2408.03335v1 [cs.CR] 21 Jul 2024.
17. Yudhir Gala, Nisha Vanjari, Dharm Doshi, Inshiya Radhanpurwala, *AI based Techniques for Network-based Intrusion Detection System: A Review*, 2023, IEEE ISBN:978-93-80544-47-2.