POLITECNICO DI TORINO

Master's Degree in ICT for Smart Societies

Master's Degree Thesis

Design and Development of a Cloud Monitoring System for Enhanced Network Security

Supervisor

Prof. Marco MELLIA

Prof. Enrico VENUTO

Candidate

Rasoul ADIM HAFSHEJANI

April 2023

Summary

With the increasing adoption of cloud computing and the growing reliance on networked infrastructure, improving cloud network security has become an increasingly vital concern. One critical aspect of network security is monitoring, which involves keeping track of the devices that are connected to the network and detecting potential security threats such as IP and MAC address duplications.

To address these challenges, this thesis focuses on providing a customized monitoring system. The system is designed to gather data from ARP tables of routers in a network. A specific script is used to export the data at regular intervals, and this exported data is then transferred to a monitoring server. The system analyzes the IP and MAC of each line of the files and stores this information in a table. The system's modular design also makes it easy to scale and extend, allowing users to add new data sources, such as additional databases or logs, and customize the dashboard to their specific needs.
To provide a comprehensive view of the network and its devices, the system integrates metadata from other databases. This table is created and updated by operators and includes valuable information about each IP and MAC such as the department in which the device is used, the department in which the user who works with this device works, the type of device, the type of operating system and etc. To accomplish this, the system replicates the metadata table from another database on a local database.
The system then matches the exported data from the ARP table with the metadata table and stores valuable information for each line in a specific database table. By integrating metadata from other databases, the system provides a comprehensive view of the network and its devices. The metadata allows the system to group connected devices by various parameters, such as departments, types of devices, and operating systems. This provides useful insights into the distribution of network devices across the organization.

Once the data is stored, the system processes it to extract various statistics that provide a comprehensive overview of the network's status. These statistics include

the total number of connected devices, the number and list of MAC and IP duplications, the number of corrupted lines in ARP tables, the number of connected devices in each department and etc. The system presents all statistics in various dashboards.

The dashboards include charts and graphs that enable users to visualize network activity over time and quickly identify potential security threats. Dashboards are important tools for network administrators to monitor their networks effectively. One of the significant benefits of these dashboards is that administrators can easily keep track of the number of devices connected to the network. This information can be helpful in determining the network's capacity and ensuring that it is not overloaded. Furthermore, administrators can analyze the trends and fluctuations in connected devices, which can help them make informed decisions about resource allocation, capacity planning, and detecting threats.
Another key benefit of the dashboard is that administrators can easily check the distribution of devices across different departments in real time. This feature can help them identify any issues with network traffic distribution and take corrective actions to optimize the network.

From a security standpoint, dashboards can also help administrators detect and respond to potential security threats proactively. For instance, administrators can use the dashboard to check the number and list of duplicated IP and MAC addresses on the routers. This information can be critical in identifying any malicious activity on the network, such as unauthorized access or data breaches. Additionally, administrators can monitor fluctuations in the number of duplications and incomplete ARP associations. These can be indicators of network attacks, and immediate action can be taken to prevent further damage.
With the help of this monitoring, administrators can easily set alerts for significant fluctuations in the values of different parameters and receive notifications on various media, such as email or telegram, to gain a better real-time insight into the network. This enables administrators to respond quickly to any potential issues, preventing them from escalating and causing downtime.

In summary, this thesis offers a valuable contribution to improving cloud network security by providing a customized monitoring system that focuses on detecting IP and MAC duplications. The system's integration with other databases and the ability to group devices by the departments provides a more comprehensive view of the network's status. The system's modular design makes it easy to extend and customize, and the dashboard's charts and graphs enable users to quickly identify potential security threats. Overall, this system represents a valuable tool for organizations seeking to improve their network security and reduce the risk of potential cyber threats.

Acknowledgements

I would like to express my gratitude to the Politecnico di Torino for offering an exceptional Master's degree program. I am also grateful to Edisu Piemonte for their financial support, which helped me to pursue my Master's degree.

I owe a debt of gratitude to Professor Marco Mellia for his expert guidance and mentorship during my thesis study. His professional supervision was invaluable, and I am grateful for the time and effort he invested in helping me develop my research. I would also like to thank Professor Enrico Venuto and the Cybersecurity team at Politecnico di Torino, particularly Gianluca, for their assistance and the facilities they provided, which were instrumental in the success of my thesis.

I want to extend my heartfelt thanks to my family, including my father, mother, and sisters, who have always supported me from afar and encouraged me to pursue big goals in life. I am indebted to my brother Ali, who stood by me during the challenging days of immigration and encouraged me to think big and pursue new and exciting opportunities.

I am grateful for my close friends, who have been a source of support and encouragement throughout this journey. In particular, I would like to express my sincere appreciation to my dear friend Arman, who supported me through my Master's degree journey and my personal life. His support helped me create wonderful memories that I will cherish for years to come.

Finally, I owe my deepest gratitude to my wife Sepideh, who inspired me to persevere through the difficult moments and stay motivated to pursue my dreams. Her unwavering belief in me, even when I doubted myself, gave me the strength to overcome any obstacle and achieve what I thought was impossible. Without her love and support, continuing this journey would not have been possible.

Rasoul Adim

Table of Contents

List of Figures VI

Acronyms VIII

1 Overview of cloud monitoring 1
- 1.1 Cloud Computing . . . 1
- 1.2 Cloud Monitoring . . . 3
  - 1.2.1 Security Monitoring and Threats . . . 4
  - 1.2.2 Network Devices Monitoring . . . 6
- 1.3 Customized ARP Tables Monitoring . . . 7

2 Networking Basics Review 9
- 2.1 Network Definition . . . 9
- 2.2 OSI Reference Model . . . 10
- 2.3 Client-Server Paradigm . . . 12
- 2.4 Transport Protocols Services . . . 12
- 2.5 IP Address . . . 13
- 2.6 MAC Address . . . 14
- 2.7 ARP Table . . . 14
- 2.8 Process of Updating the ARP Table . . . 15
- 2.9 IP and MAC duplication . . . 15
- 2.10 ARP spoofing . . . 16

3 Implementation 17
- 3.1 Polytechnic Network Scenario . . . 17
- 3.2 Data Preparation . . . 19
  - 3.2.1 ARP table Data . . . 19
  - 3.2.2 Metadata Database . . . 22
  - 3.2.3 Data Volume . . . 22
- 3.3 Data Processing . . . 23
  - 3.3.1 Scripts . . . 23

3.3.2 Data Cleaning . . . 25
3.3.3 Statistics Extraction . . . 27
3.3.4 Storing the Data . . . 29
3.4 Visualization . . . 31
3.4.1 Dashboards' Type . . . 31
3.5 Resource and Demand . . . 32

4 Result and Discussion 34
4.1 Statistics Dashboards . . . 34
4.1.1 Dashboard 1 . . . 35
4.1.2 Dashboard 2 . . . 43
4.2 Query Dashboards . . . 50
4.2.1 Dashboard 1 and 2 . . . 50
4.2.2 Dashboard 3 and 4 . . . 50
4.2.3 Dashboard 5 . . . 51
4.2.4 Dashboard 6 . . . 52
4.2.5 Dashboard 8 and 9 . . . 53
4.2.6 Dashboard 9 and 10 . . . 54
4.3 Alerts and Notification . . . 55

5 Conclusion 58

Bibliography 61

List of Figures

1.1 Overview of Cloud Computing Essentials 1
1.2 Cloud Monitoring Components by Service Models 3
1.3 Security Issues 5

2.1 Size-based Network Classification 10
2.2 Topology-based Network Classification 10
2.3 OSI Reference Model 11

3.1 Polytechnic Network Schema 18
3.2 Monitoring System Process 19
3.3 System Scripts 23
3.4 Database and Tables 30

4.1 Statistics Dashboard 1 35
4.2 Device Type-based Distribution 37
4.3 Device Class-based Distribution 38
4.4 Connected Devices based on Distinct MAC and IP 39
4.5 Connected Devices based on Distinct MAC and IP - Weekend . . . 40
4.6 Connected Devices based on Distinct MAC and IP - Long Period . 40
4.7 Corrupted and Incomplete Input Lines 41
4.8 Number of Equal Lines in the same chunk 43
4.9 Statistics Dashboard 2 44
4.10 Department-based Devices distribution 44
4.11 Operating System-based Devices distribution 46
4.12 Distribution of Devices in Departments over Time - Top 5 47
4.13 Total Duplicated MAC and IP Addresses per Chunk 48
4.14 Distinct Duplicated MACs and IPs 49
4.15 Query Dashboard - IPs with multiple MACs at the same chunk . . 51
4.16 Query Dashboard - Duplicated MACs at the same chunk 52
4.17 Query Dashboard - Equal Lines at the same chunk 53
4.18 Query Dashboard - Department-based Device distribution 53

4.19 Query Dashboard - ARP and Metadata Table data Anomalies detected based on IP 54
4.20 Query Dashboard - Search Information in the database based on IP 55
4.21 Alert and Notification Dashboard 56
4.22 Alert and Notification - Telegram Media Type 57

Acronyms

IaaS Infrastructure as a Service

PaaS Platform as a Service

SaaS Software as a Service

AWS Amazon Web Services

CPU Central Processing Unit

GCP Google Cloud Platform

IP Internet Protocol

MAC Media Access Control

DoS Denial-of-Service

DDoS Distributed Denial-of-service

IDS Intrusion detection systems

IPS Intrusion prevention systems

AI Artificial intelligence

ML Machine learning

LAN Local Area Network

WAN Wide Area Network

MAN Metropolitan Area Network

GAN Global Area Network

OSI Open Systems Interconnection

TCP Transmission Control Protocol

UDP User Datagram Protocol

ARP Address Resolution Protocol

DHCP Dynamic Host Configuration Protocol

VSS Virtual Switching System

LAG Link Aggregation Group

SCP Secure Copy Protocol

Rsync Remote Sync

SSH Secure Shell

SSD Solid State Drive

HDD Hard Disk Drive

VM Virtual Machine

Chapter 1

Overview of cloud monitoring

1.1 Cloud Computing

Cloud computing refers to the delivery of computing services including servers, storage, databases, networking, software, analytics, and intelligence over the Internet (the cloud) to offer faster innovation, flexible resources, and economies of scale. You can access these services from anywhere, at any time, on any device. Instead of maintaining and managing these resources on-premises, businesses and individuals can access them from a cloud provider's shared pool of resources[1]. This allows for increased flexibility, scalability, and cost savings since resources can be purchased on an as-needed basis.

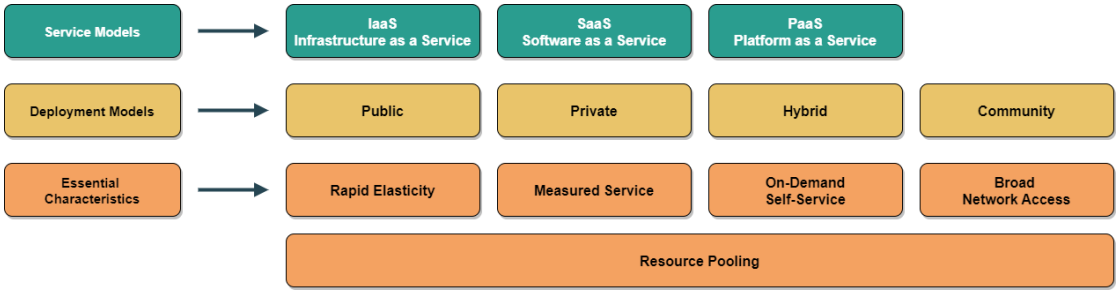

Figure 1.1: Overview of Cloud Computing Essentials

Cloud computing can be divided into three main service models[2]: (Figure 1.1)

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

IaaS provides access to virtualized computing resources. This includes physical servers, storage, and networking capabilities, which are made available to users on a pay-per-use basis[3]. PaaS provides a platform for developing, testing, deploying, and managing web applications and services[4]. SaaS delivers software applications over the internet. these applications are typically accessed through a web browser and can be used by businesses and individuals on a pay-per-use or subscription basis[5].
There are several different cloud deployment models (Figure 1.1), including:

- Public Cloud: Public clouds are owned and operated by a third-party provider and made available to the public over the internet. Examples include Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).
- Private Cloud: Private clouds are owned and operated by a single organization for their own use. They can be physically located on-premises or in a data center managed by a third-party provider.
- Hybrid Cloud: Hybrid clouds combine elements of both public and private clouds, allowing organizations to take advantage of the cost savings and scalability of public clouds while maintaining the control and security of a private cloud.
- Community Cloud: Community clouds are shared by multiple organizations with similar needs or goals. They can be operated by a third-party provider or a group of organizations.

The essential characteristics of cloud computing include: (Figure 1.1)

- Rapid elasticity: Computing resources can be quickly and easily scaled up or down as needed to meet changing demand.
- Measured service: Cloud providers track and measure the usage of computing resources, and users are typically only charged for the resources they consume.
- On-demand self-service: Users can provision computing resources, such as servers and storage, on an as-needed basis, without the need for human intervention.
- Broad network access: Computing resources can be accessed over a network, such as the internet, from anywhere and from any device.

- Resource pooling: Computing resources, such as servers and storage, are pooled together and dynamically allocated as needed.

1.2 Cloud Monitoring

Cloud monitoring refers to observing and tracking various aspects of cloud-based resources and services to ensure that they are operating correctly and efficiently [1]. This can include monitoring things like performance metrics, resource usage, availability, and error rates. Cloud monitoring is similar to traditional monitoring, but it is specifically designed to work with cloud-based resources and services. This means that it takes into account the unique characteristics and challenges of cloud computing, such as the dynamic nature of cloud environments, the need for scalability and elasticity, and the need to monitor distributed systems. Cloud monitoring tools serve the purpose of overseeing various elements within a cloud service. The specific scope of this monitoring can vary depending on the cloud service model[6]. As illustrated in Figure 1.2, different service models have distinct components that fall under the umbrella of cloud monitoring.

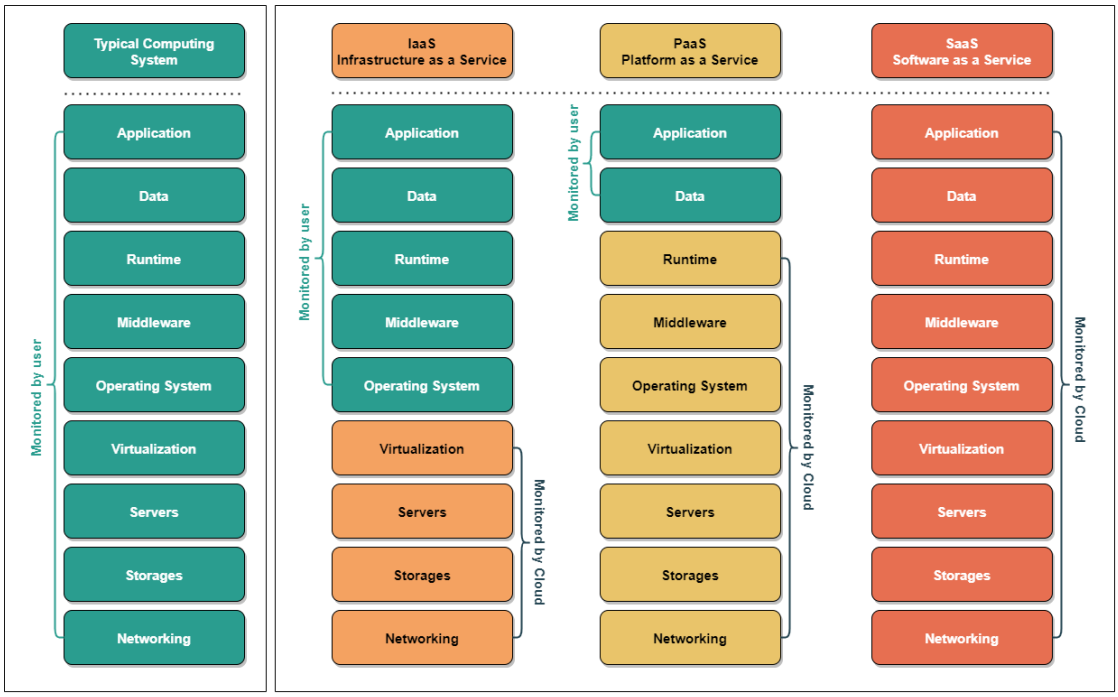

Figure 1.2: Cloud Monitoring Components by Service Models

The components monitored by cloud monitoring tools vary depending on the cloud service model, as demonstrated in Figure 1.2. In the case of Infrastructure

as a Service (IaaS), monitoring typically covers networking, storage, servers, and virtualization. For Platform as a Service (PaaS), in addition to these components, monitoring must also include the operating system, middleware, and runtime. Finally, in Software as a Service (SaaS), all components are monitored and managed by the cloud provider, allowing users to simply utilize the software without worrying about any underlying infrastructure. Cloud monitoring can also include a variety of tasks and processes [7], such as:

- Resource monitoring: This type of monitoring focuses on tracking the utilization of cloud-based resources, such as servers, storage, and networking. It can include monitoring metrics such as CPU usage, memory usage, disk space, and network bandwidth to ensure that resources are running optimally and to identify any potential performance bottlenecks. Resource monitoring can also include monitoring for events such as disk failures, network outages, and other issues that may affect the availability of resources.

- Application monitoring: This type of monitoring focuses on tracking the performance and availability of cloud-based applications and services. It can include monitoring metrics such as response time, error rates, and throughput to ensure that applications are running smoothly and to identify any potential issues. Application monitoring can also include monitoring log files and error messages to identify and troubleshoot any issues or errors.

- Security monitoring: This type of monitoring focuses on tracking security-related events and metrics, such as unauthorized access, data breaches, and compliance with security regulations and best practices. It can include monitoring for events such as failed login attempts, suspicious network traffic, and security alerts from firewalls and intrusion detection systems.

Cloud monitoring is an ongoing process, and the collected data should be analyzed regularly to identify any trends or patterns in the usage of resources and applications. This can help to identify potential issues before they become problems and allow for proactive measures to be taken to optimize performance and reduce costs. It also helps to ensure that the cloud-based infrastructure is secure and compliant with industry regulations and best practices[8].

1.2.1 Security Monitoring and Threats

Security monitoring in the cloud is the process of tracking and analyzing security-related events and metrics to ensure that the cloud-based infrastructure is secure and compliant with industry regulations [9]. This can include monitoring for suspicious network traffic, unauthorized access attempts, and other indicators of

potential security breaches. Some examples of security monitoring in the cloud include:

- Network traffic monitoring: This involves monitoring for unusual patterns of network traffic, such as high volumes of traffic from a single IP address or traffic that is originating from a known malicious source.
- Intrusion detection and prevention: This involves monitoring for attempts to gain unauthorized access to the cloud environment, such as through the use of stolen login credentials or by exploiting vulnerabilities in the system.
- File integrity monitoring: This involves monitoring for changes to important system files, such as configuration files, in order to detect any unauthorized modifications that may have been made.

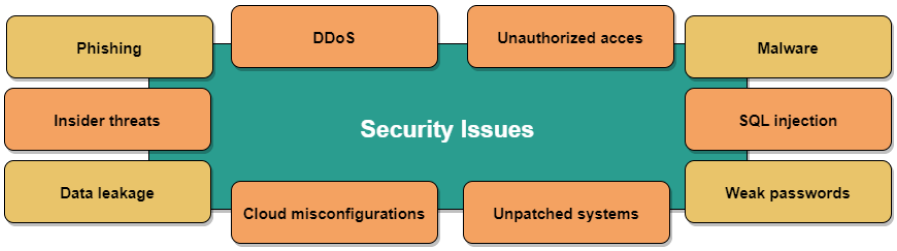

Figure 1.3: Security Issues

There are many different types of security issues that organizations must consider when securing their cloud environment (Figure 1.3). Some examples include:

- Phishing: Attempts to steal sensitive information by disguising oneself as a trustworthy entity in an electronic communication, like an email or message.
- DDoS (Distributed Denial of Service) attacks: Attempts to make a machine or network resource unavailable by overwhelming it with traffic from multiple sources.
- Unauthorized access: Attempts to gain access to a system or data without proper authorization.
- Malware: Malicious software like viruses, trojan horses, worms, and ransomware can infect systems and cause damage or steal data.
- SQL injection: Attempts to take control of a database by injecting malicious code into SQL statements.

- Weak passwords: The use of easily guessed or cracked passwords can make it easy for attackers to gain unauthorized access to systems and data.
- Unpatched systems: Outdated software and systems can contain known vulnerabilities that attackers can exploit to gain unauthorized access or launch attacks.
- Insider threats: Employees or contractors with access to cloud-based systems may misuse that access for malicious purposes, such as stealing sensitive data.
- Cloud misconfigurations: Inadequate security settings, lack of encryption, and other misconfigurations can leave cloud-based systems vulnerable to attack.
- Data leakage: Unauthorized transfer of data from a secure system to an unsecured system.

These are just a few examples of the many security issues that organizations must consider when securing their cloud environment. It's important to have comprehensive security monitoring in place to detect and respond to potential security threats and vulnerabilities in a timely manner.

1.2.2 Network Devices Monitoring

Monitoring network devices involves continuously monitoring and analyzing the security of these devices in order to detect and respond to potential security threats. Routers and switches are critical components of a network and are responsible for directing and managing network traffic. As such, they are often targeted by attackers looking to gain unauthorized access to a network or launch a denial-of-service (DoS) attack.
There are several key aspects of security monitoring of routers and switches, for example, Intrusion detection and prevention. This involves monitoring network traffic for signs of unauthorized access, such as hacking attempts or port scans. Intrusion detection systems (IDS) and intrusion prevention systems (IPS) can be used to detect and respond to potential security threats in real-time. Moreover, monitoring connected devices to network equipment is an essential aspect of maintaining the security and availability of the network. By keeping track of connected devices, organizations can gain valuable insights into the number of devices connected to the network, as well as identify any IP or MAC address duplications. Additionally, monitoring connected devices can also help to detect and respond to failed or incomplete connection attempts, which can be an indication of a potential security threat.
These statistics can help organizations to better understand their network environment, identify potential security vulnerabilities and take proactive measures to

prevent security incidents from occurring. By implementing monitoring of connected devices, organizations can ensure the confidentiality, integrity, and availability of data and systems, and protect their assets and reputation.

1.3 Customized ARP Tables Monitoring

Monitoring the Address Resolution Protocol (ARP) table is a powerful security measure that can help protect networks from a variety of threats. One of the primary benefits of monitoring ARP is that it allows for the detection of ARP spoofing attacks, which occur when an attacker sends false ARP messages on a network in order to map their own IP address to the MAC address of another device on the network. This can allow the attacker to intercept and modify network traffic, potentially stealing sensitive information or disrupting network operations. By monitoring the ARP table, network administrators can quickly detect and respond to these types of attacks. Another benefit of monitoring ARP is that it can help identify rogue devices on a network. For example, if an unauthorized device is connected to a network and is sending ARP messages, monitoring the ARP table can reveal its presence. This can be especially useful in environments with many connected devices, such as in large organizations or public spaces.

There are monitoring systems that monitor ARP tables. PRTG and Zabbix are comprehensive network monitoring tools that can monitor various aspects of networks, including ARP tables. With PRTG, administrators can:

- Monitor ARP cache entries: PRTG can monitor the ARP cache entries on network devices to ensure that each device has a unique IP address.
- Detect ARP spoofing: PRTG can detect and alert administrators to any ARP spoofing attempts, where an attacker tries to use a fake ARP message to associate their IP address with the MAC address of another device.
- Configure alerts: PRTG allows administrators to set up alerts for specific ARP-related events, such as the addition or removal of ARP entries, and can send notifications via email, SMS, or push notifications.

In our thesis, we have used the concept of monitoring the ARP table as a security measure to detect and prevent ARP spoofing attacks. We proposed a solution to detect ARP spoofing attacks by monitoring the ARP table and comparing it with a list of known devices. Our solution also provides a real-time alert to the network administrator if an attack is detected. In addition to using a commercial network monitoring tool, we have also implemented a custom ARP table monitoring system specifically designed for our environment. Our open-source system is connected to our metadata database and obtains information about IP addresses, departments,

devices type and operating systems and etc. This information is used to generate detailed statistics and provide a comprehensive view of our network. Our custom ARP table monitoring system is designed to accurately reflect our unique network configuration and provide real-time insights into the status of our ARP entries. With this customized system in place, we are able to quickly detect and resolve any issues related to IP and MAC duplication, ensuring that our network continues to operate smoothly and efficiently. This tailored solution offers greater flexibility and granularity in our network monitoring, providing us with a customized approach to managing and maintaining our ARP tables.

In conclusion, monitoring the ARP table is an essential security measure for protecting networks from a variety of threats. Its benefits include detecting ARP spoofing attacks and identifying rogue devices on the network, it can be a useful tool for securing networks in any environment.

Chapter 2

Networking Basics Review

2.1 Network Definition

A network is a group of interconnected devices, such as computers, servers, and routers, that can communicate and share resources with each other. Networks can be classified based on their size, purpose, and topology. Based on the size (Figure 2.1), networks can be classified as:

- Local Area Network (LAN) - A network that connects devices within a small geographic area, such as a single building or campus.
- Metropolitan Area Network (MAN) - A network that connects devices within a metropolitan area.
- Wide Area Network (WAN) - A network that connects devices across a larger geographic area, such as multiple buildings or cities.
- Global Area Network (GAN) - A network that connects devices around the world.

Based on purpose, networks can be classified as:

- Data networks - Networks used to transmit data, such as the internet and private intranets.
- Telecommunication networks - Networks used to transmit voice and video, such as telephone networks and cable TV networks.
- Control networks - Networks used to control and monitor devices, such as industrial control systems and the power grid

Based on topology (Figure 2.2), networks can be classified as:

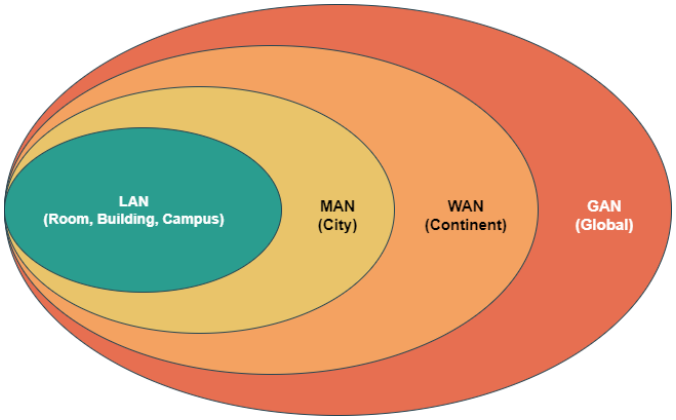

Figure 2.1: Size-based Network Classification

- Bus topology: A network where all devices are connected to a single cable.
- Star topology: A network where all devices are connected to a central hub.
- Ring topology: A network where devices are connected in a closed loop.
- Mesh topology: A network where devices are connected to multiple other devices.

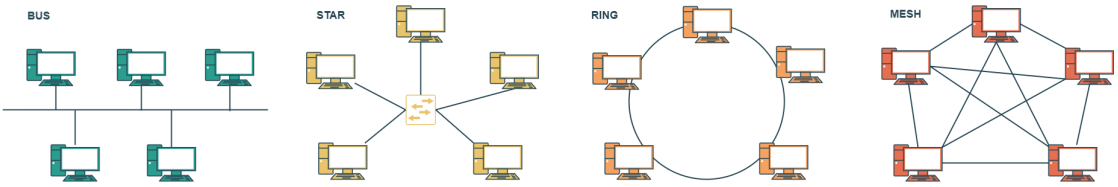

Figure 2.2: Topology-based Network Classification

Overall, a network allows devices to communicate, share resources, and exchange information.

2.2 OSI Reference Model

The OSI (Open Systems Interconnection) model is a conceptual framework that describes how different layers of a computer network interact with each other [10]. The OSI model as depicted in Figure 2.3 is divided into seven layers, each with a specific function:

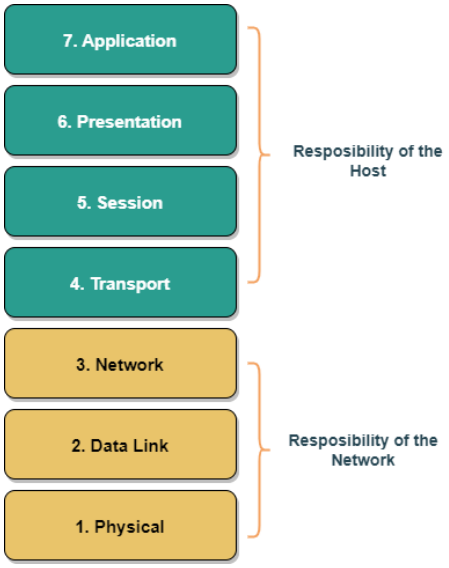

Figure 2.3: OSI Reference Model

1. The Physical Layer: This is the lowest layer of the OSI model and it is responsible for the physical connections between devices on a network. This layer includes the cables, connectors, and other hardware that make up the network.

2. The Data Link Layer: This layer is responsible for creating a reliable link between devices on a network. It uses MAC addresses to identify devices and it provides error detection and correction for data transmitted over the network.

3. The Network Layer: This layer is responsible for routing data packets between devices on different networks. It uses IP addresses to identify devices and it determines the best path for data packets to take.

4. The Transport Layer: This layer is responsible for ensuring that data is delivered reliably and in the correct order. It uses port numbers to identify different applications running on a device and it provides flow control and error recovery.

5. The Session Layer: This layer is responsible for establishing, maintaining, and terminating connections between devices on a network. It coordinates the flow of data between devices and it can be used to recover from errors or network failures.

6. The Presentation Layer: This layer is responsible for converting the data into

a format that can be understood by the application. It performs functions such as data compression, encryption and decryption, and data conversion.

7. The Application Layer: This is the highest layer of the OSI model and it is responsible for providing services to the user. This layer includes the application software and the user interface.

The OSI model provides a standard way of describing how different layers of a computer network interact with each other, it helps to understand the different functions and protocols that are used in networks and it also helps to troubleshoot and identify the problem in a network.

2.3 Client-Server Paradigm

The client-server paradigm is a model for distributed computing in which a client, the requesting device, sends a request to a server, the providing device, which then processes the request and sends back a response. In this model, the client and server have distinct roles and responsibilities:

The client is responsible for creating and sending requests to the server. It can be a computer, mobile device, or any other device that can initiate a network request.

The server is responsible for receiving requests from the client, processing them, and sending back a response. It can be a computer, a device with specialized hardware, or a virtual machine running on a cloud platform.

The client-server paradigm is used for a wide variety of applications, such as web browsing, email, file sharing, and online gaming. This model is scalable and robust, and it allows for the separation of concerns between the client and server.

2.4 Transport Protocols Services

Transport protocols are responsible for providing services to ensure that data is delivered reliably and efficiently between applications running on different devices in a network. The two main transport protocols are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Transmission Control Protocol (TCP) provides a reliable, stream-oriented service. It establishes a reliable connection between the sender and the receiver before any data is exchanged. It uses a three-way handshake to establish the

connection, and it guarantees that all data sent will be received by the other end and in the same order as it was sent.

User Datagram Protocol (UDP) provides a connectionless, datagram-oriented service. Unlike TCP, UDP does not establish a connection before exchanging data and does not guarantee that the data will be received by the other end. However, UDP is faster than TCP as it has a simpler header and fewer overhead.

TCP and UDP are used in different types of applications, depending on their requirements. TCP is typically used in applications that require a reliable and ordered delivery of data, such as file transfer, email, and web browsing. On the other hand, UDP is typically used in applications that require fast, low-latency communication, such as online gaming, voice-over IP (VoIP), and live streaming.

In summary, transport protocols are responsible for providing services that ensure that data is delivered reliably and efficiently between applications running on different devices in a network, such as TCP and UDP.

2.5 IP Address

An IP (Internet Protocol) address is a unique numerical label assigned to every device connected to a computer network that uses the Internet Protocol for communication. It is used in the Network Layer (Layer 3) of the OSI reference model. It serves two main functions: identifying the host or network interface and providing the location of the host in the network.

There are two versions of IP in use today: IPv4 and IPv6. IPv4 uses 32-bit addresses and can support about 4.3 billion devices. However, as the number of devices connected to the internet has grown rapidly, IPv4 addresses have become scarce. IPv6, on the other hand, uses 128-bit addresses and can support a virtually unlimited number of devices.
An IPv4 address is typically written as four numbers separated by dots, for example, 192.168.1.1. Each number can be from 0 to 255. An IPv6 address is written as eight groups of four hexadecimal digits, separated by colons, for example, 2001:0db8:85a3:0000:0000:8a2e:0370:7334.
IP addresses are used to identify devices on a network and to route data packets between devices. When a device sends data to another device, it sends the data to the destination device's IP address. The data is then routed through the network, using routers, to the destination device. In addition to identifying devices on a network, IP addresses can also be used to identify the location of a device. For example, the first three octets of an IPv4 address can be used to identify the network and the last octet can be used to identify the specific device on that

network.
In summary, IP addresses are unique numerical labels assigned to devices connected to a network, used to identify and locate them on a network, as well as routing data packets between them.

2.6 MAC Address

A MAC (Media Access Control) address is a unique identifier assigned to network interfaces for communications on the physical network segment. It is used to identify a specific device on a network and it is usually assigned by the manufacturer of the network interface card (NIC).
A MAC address is a 48-bit (6 bytes) address that is usually written in the form of 12 hexadecimal digits (0-9, A-F) separated by colons or hyphens. For example, a MAC address might look like this: 00:11:22:33:44:55.
Each device that connects to a network, such as a computer, smartphone, or router, has a unique MAC address. When a device sends data to another device on the same network, it sends the data to the destination device's MAC address. This allows the data to be sent directly to the correct device, rather than being broadcast to all devices on the network.
A MAC address is typically used at the Data Link Layer of the OSI model, which is responsible for physically delivering data between devices on a network. It also works as a low-level addressing system, it is used for communication between devices in the same LAN (Local Area Network), but it's not meant to be used for communication between devices in different networks.
The MAC address is a unique, hard-coded number that is assigned to a device's network interface card (NIC) during the manufacturing process and it cannot be changed. However, it is possible to use a technique called MAC address spoofing, in which a device can be configured to use a different MAC address. This can be used to hide the true identity of a device or to bypass security restrictions.
In summary, a MAC address is a unique identifier assigned to network interfaces for communications on the physical network segment, it is used to identify a specific device on a network and it is used for communication between devices in the same LAN.

2.7 ARP Table

ARP (Address Resolution Protocol) table is a database that is used by a network device to map a network layer address (such as an IP address) to a corresponding link-layer address (such as a MAC address). The ARP table is used to translate the IP addresses of devices on a network into their corresponding MAC addresses,

so that data can be sent directly to the correct device.
ARP table shows the IP addresses and corresponding MAC addresses of devices that have recently communicated with the device. The ARP table is usually stored in the memory of a device and it is a volatile table, which means that the entries in the table will be removed after a certain period of time if they are not used again. This is to ensure that the ARP table does not become too large and consume too much memory.
It is important to note that ARP can also be used for malicious purposes, such as ARP spoofing or ARP cache poisoning, which can be used to intercept or redirect network traffic.
In summary, the ARP table is a database used by a network device to map a network layer address to a link-layer address, it is used to translate the IP addresses of devices on a network into their corresponding MAC addresses, and it is stored in the device's memory with a limited size.

2.8 Process of Updating the ARP Table

The process of updating the ARP table involves the following steps:

1. A device needs to send data to another device on the same network segment.
2. The device checks its ARP table to see if it already has the MAC address corresponding to the IP address of the destination device.
3. If the ARP table contains the mapping, the device uses the cached MAC address to send the data directly to the destination device.
4. If the ARP table does not contain the mapping, the device sends an ARP request (broadcast) to all devices on the network segment, asking for the MAC address corresponding to the IP address of the destination device.
5. The device with the matching IP address responds to the ARP request with its MAC address, and the requesting device caches the mapping in its ARP table.
6. The requesting device uses the received MAC address to send the data directly to the destination device.

2.9 IP and MAC Duplication

IP and MAC duplication refers to the situation where two or more devices on a network have the same IP address or MAC address, respectively. This can cause

conflicts and result in communication issues, as the network may not be able to distinguish between the devices. It is important to ensure that each device on a network has a unique IP and MAC address to ensure proper communication and avoid conflicts. IP and MAC duplication can happen due to a variety of reasons, including:

Configuration errors: When manually configuring network devices, it is possible to accidentally assign the same IP or MAC address to two different devices.

DHCP server issues: If the Dynamic Host Configuration Protocol (DHCP) server is not configured properly, it may assign the same IP address to multiple devices.

Clone or spoofing: Attackers can use IP or MAC cloning or spoofing techniques to duplicate the IP or MAC address of another device on the network.

DHCP lease expiration: If a device that was assigned an IP address through DHCP is turned off or disconnected from the network, the DHCP server may assign the same IP address to a different device when it comes online.

DHCP server failure: If the DHCP server fails, it may temporarily assign the same IP address to multiple devices.

To avoid IP and MAC duplication, it is important to properly configure network devices, use reliable and secure DHCP servers, and monitor network activity and security for any suspicious activity.

2.10 ARP spoofing

ARP Spoofing is a type of cyber attack where an attacker manipulates the ARP cache of a target computer to associate the attacker's MAC address with the IP address of a legitimate device on the same network. As a result, any network traffic intended for the legitimate device is instead sent to the attacker's device.
The attacker can then use this information to intercept and modify network data, launch man-in-the-middle attacks, or even perform denial-of-service attacks by disrupting communication between the target device and the rest of the network.
Preventing ARP spoofing attacks involves using ARP security measures such as static ARP mapping, and ARP inspection, and implementing security technologies like firewalls and intrusion detection systems.
It's important to note that ARP spoofing attacks are highly effective due to the inherent trust relationship between devices on a network and the lack of encryption and authentication in the ARP protocol.

Chapter 3

Implementation

3.1 Polytechnic Network Scenario

This monitoring system was implemented for the Polytechnic University of Turin's network, which comprises all departments and campuses of the university. The network has 12 routers/switches that support layer three and are spread across different departments, including the main campus and remote sites.
Seven of these routers are located in remote sites, which can be within or outside the city of Turin. The routers used in these remote sites are Cisco 2960 or Cisco 3750 models, and their names are as below.

"l3-energycenter"
"l3-lingotto"
"l3-morgari"
"l3-settembrini"
"l3-trento"
"l3-valentino"
"l3-verres"

Figure 3.1 provides a high-level view of the network, with remote sites connected to the **"irf-vss"** router, which is a logical switch comprising two physical switches of HPE FlexFabric 5945. This router supports layer 3 and Virtual Switching System (VSS) with the HPE Comware operating system, allowing two FlexFabric 5945 switches to be combined to form a single logical switch.

Various departments, offices, and classrooms for example Departments DET, DAIUN, DENERG, and others, are part of this network, with each department connected to the **"irf-core2"** logical switch, which is a combination of three physical switches of HPE FlexFabric 5940 through VSS. The university labs are also connected to the **"irf-laib"** router, which is another logical switch with the same

configuration as "irf-vss."
These three logical switches are the distribution switches, which are connected to two core switches **"nexus1"** and **"nexus5"** through LAG links (link aggregation group). These core switches are Cisco Nexus 7009 Model. "irf-laib" is connected to core switches with a link of 80 Gbps, "irf-core2" with a link of 240 Gbps, and "irf-vss" with a link of 160 Gbps. The core switches are then connected to data center services and the internet, providing access to different services for departments, labs, and remote sites.
Therefore, there are 12 routers in total, including the core routers, in this network. This network includes only the wired connections of the university. However, the wired network is connected to another network, and the wireless network, which is another large network of the university in terms of connected devices, is not connected to this network.

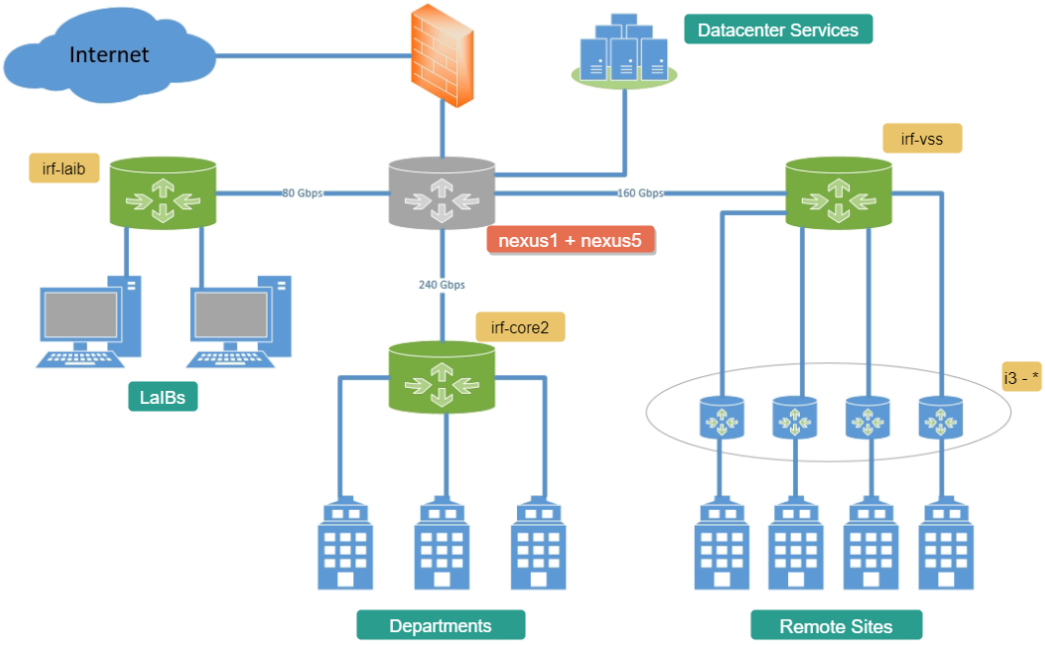

Figure 3.1: Polytechnic Network Schema

3.2 Data Preparation

The data fed into this system is a combination of two different sources. As Figure 3.2 depicts, the first source is the data extracted from the ARP table of the routers, which contains information about the IP and MAC addresses of devices connected to the network and the router names.
The second source is the metadata that is replicated from one of the Polytechnic databases, which holds all the information related to IP, MAC, user, device, and departments. This metadata provides additional context about the devices and users on the network, including their department affiliations. The data from these two sources is crucial for the system to accurately identify and manage devices and users on the network. The data comes into the system in different ways, depending on the source. The data from the ARP table is typically transmitted via network protocols, while the metadata from the Polytechnic database is typically replicated and updated periodically through database synchronization processes.

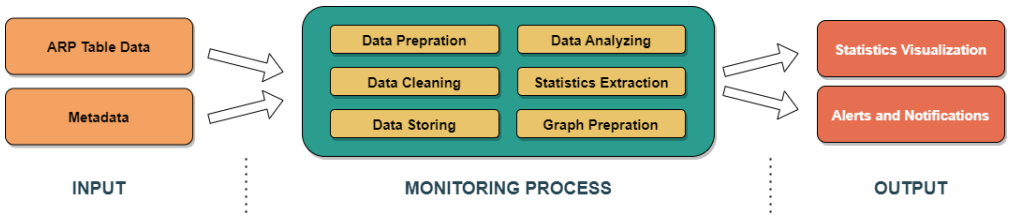

Figure 3.2: Monitoring System Process

3.2.1 ARP table Data

Data Extraction

The ARP table data is extracted using a script specifically designed for this purpose on a separate server. This script attempts to extract the content of the ARP table of each router by interval and store it in a separate file. To schedule this script to run automatically at specific intervals, it uses Crontab, a Linux command that schedules recurring tasks or commands to be executed automatically.
Crontab uses a special syntax to specify the time and frequency of the scheduled tasks, consisting of five fields representing the minutes, hours, day of the month, month, and day of the week. Users can set any combination of these fields to define the schedule of the job.
The main command used in this script is as below.

```
SNMPTABLE -r3 -t5 -v2c -m [MIBs-FILES] -CH -Cf " " -OOT
-Ln -c [SNMP-COMMUNITY] [IP] ipNetToMediaTable > [OUTPUT-FILES]
```

This command uses the SNMPTABLE tool to retrieve and display data from SNMP-enabled network devices in a tabular format. The ipNetToMediaTable is a table maintained by the ARP and NDP protocols that maps the IP addresses of devices on a network to their corresponding MAC addresses. It provides a list of all IP addresses on the network along with their corresponding MAC addresses.
SNMP is a protocol used to monitor and manage network devices such as routers, switches, servers, and printers. The SNMPTABLE command allows network administrators to easily retrieve information from SNMP-enabled devices and present it in a structured format for easy analysis and monitoring.

Other important parameters in this command include:

- [MIBs-FILES] : The required MIBs file
- [SNMP-COMMUNITY]: Specifies the SNMP community configured into the router to extract ARP tables
- [IP] : Specifies the IP address of the router
- [OUTPUT-FILE] : Specifies the output file where the data is stored

Data Transfer

Once the files from each interval are stored, all MAC addresses within the files are hashed using a specific algorithm for privacy reasons. Hashing is a process that takes an input (in this case, a MAC address) and produces an output of a fixed length, known as a hash value. This output is a unique representation of the original input, and it is not possible to determine the original input based on the hash value alone. By using this specific algorithm to hash the MAC addresses in the text files, the system ensures that any personally identifying information is protected and kept private.
These files are then transferred to the monitoring server using one of two solutions: SCP or Rsync. Both SCP and Rsync are command-line tools used for transferring files between remote and local systems and are commonly used in Linux and Unix-like operating systems.
SCP (Secure Copy Protocol) is a command that enables users to securely transfer files between local and remote systems using the SSH (Secure Shell) protocol. It encrypts data during transfer and requires authentication with a username and password or SSH key. SCP preserves file attributes such as timestamps, permissions, and ownership. It is useful for transferring files that are not too large or for one-time transfers.

On the other hand, Rsync (Remote Sync) is a command that synchronizes files and directories between local and remote systems. It uses an efficient algorithm that compares files and only transfers the differences between them, reducing transfer time and bandwidth usage. Rsync can be used for backups, mirroring, and general file transfers. It also preserves file attributes and supports copying files over SSH, making it secure.

Arp Table Files

The input ARP table data in the monitoring server is a collection of text files, one for each router. These files are obtained every fifteen minutes, resulting in a total of 12 files in each interval from the routers which have been described in section 3.1. The name of each file includes important information such as the chunk date and time, as well as the router name.

```
20230312_101501_l3-settembrini_no_MAC.txt
```

In the example above, the file name indicates that it contains data related to a specific chunk of information, which was recorded on March 12th, 2023 at 10:15:01, and came from the router named 'l3-settembrini' which is one of the remote sites (Figure 3.1). This information is kept in a database for later use in the analysis. Each file contains the contents of the ARP table for the corresponding router. The ARP table is a mapping of IP addresses to MAC addresses and is used for network communication. Each line in the file represents an entry in the ARP table, with the IP and corresponding MAC address being the important data for this project. This information will be used to analyze and understand the network communication patterns. It's important to note that due to privacy concerns, the MAC addresses in the text files received by the system are hashed using a specific algorithm.

```
110 cdf10ebf765f966d792f88f6beeda917b359945a 192.168.159.254 static
110 cdf10ebf765f966d792f88f6beeda917b359945a 192.168.160.254 static
111 02699eb9b6351e6df650c1ba0967e7aa035e8dc2 172.30.107.29 dynamic
111 46d73536bc221b56633a00a41eee52ae8a5696b8 172.30.107.30 dynamic
```

In the example above, we can see four lines from one of our input files. Each line contains four parts. The second and third parts are the ones we need in our system: the hashed MAC address and the real IP address, respectively. We store this data, along with the chunk time and router name, in our database for further analysis.

3.2.2 Metadata Database

The metadata table in one of the polytechnic databases is an important source of information that holds various details about devices used in the institution. This table contains a plethora of data, such as IP and MAC addresses, device names, department names, device types, device usage classes, operating system types and etc.
The various columns present in the metadata table play a crucial role in extracting valuable statistics related to the polytechnic's network infrastructure. For instance, the department column helps in identifying the number of devices connected from a particular department, providing insight into the department's technological needs. Similarly, the device type column allows the institution to keep track of the number of devices of a particular type, aiding in inventory management and budget allocation. Furthermore, the device usage class and operating system type columns provide additional information that can help identify trends or issues related to the usage of devices across the institution.
As mentioned earlier, this metadata table is another input to our system (Figure 3.2). We copy this table from one of Polytechnic's databases to a specific table on the local system's database using a replication process.

Database Replication

The data presented in the table is a replication of information from a source table located in one of the databases of the polytechnic. To ensure that the system has the latest version of data, the replication process takes place every early morning at 3:30 am when the workload is minimal and does not interfere with other processes on both sides.
Initially, the existing values in the local table are removed before establishing a new connection to the source SQL database. A 'SELECT' query in SQL is used to read all the new data, and these values are then put in a data frame. Finally, the data frame's content is written to the local table by an 'INSERT' query in MySQL local database. This process ensures that the data in the table is up to date and can be used for statistical purposes.

3.2.3 Data Volume

The data volume in this system is significant, as it receives a large number of text files on a regular basis. Specifically, every fifteen minutes, the system receives 12 text files from 12 different routers. This translates to 48 files every hour and 1152 files in a day. These files must be processed and stored in a database in order for the system to function properly.
The number of lines of data in each file can vary depending on the location of

the router and the time of day. During working hours, the number of connected devices is typically much higher than during other hours, so files captured during those times may contain more data. However, when considering all of the routers together, the system receives an estimated 1.5 to 4 million rows of data every day. This volume of data is important to take into account when designing the system. This is particularly important when it comes to real-time monitoring, as the system must be able to handle and process such a large amount of data in a timely manner. It is also important to note that this volume of data can put pressure on the storage, processing, and network of the system, therefore it is important to consider these factors in the system design as well.

3.3 Data Processing

3.3.1 Scripts

In order to efficiently handle all the necessary processes for the data received by the system, four different scripts have been developed. These scripts which are written in Python programming language, are responsible for different stages of the data handling process and work together to ensure that the data is properly cleaned, processed, and stored.

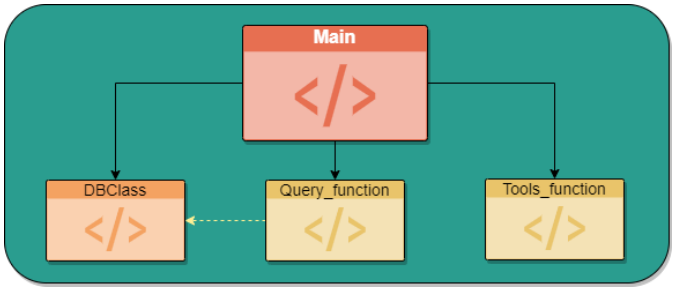

Figure 3.3: System Scripts

The first script, referred to as the **"main"**, plays a crucial role in the data handling process of the system. This script is responsible for several tasks including detecting new files, reading their contents, storing the data into the database, calculating statistics based on the input data, and inserting those statistics into a table.
The 'main' script starts by detecting new files that have been received by the system, it then reads all the content of these files and stores the data into the database. This script also calculates statistics based on the input data. This step is important in order to gain useful insights and understanding of the data. The calculated statistics are then inserted into a table. This step helps to provide a clear overview of the system and to monitor the performance and status of the

system.

As mentioned before, the "main" script is responsible for detecting new data and when new data is detected, it calls the **"tools_function"** script. The "tools_function" script is responsible for cleaning the data, specifically, it checks the input data for any corruption which is detailed in the section 3.3.2. This script has also other responsibilities to check certain parameters of input lines, including the format of IP addresses and the length of MAC addresses. These checks are essential for ensuring the accuracy and consistency of the data. Further details on these checks are provided in the data cleaning section, where their importance in the overall data cleaning process is explained.

The third script, **"query_functions"**, serves multiple important purposes in the overall system. Firstly, it is responsible for replicating the Metadata table, as detailed in section 3.2.2. This replication process is crucial for ensuring that the data is up-to-date and accurate, and the script is designed to perform this task efficiently and effectively. Additionally, this script is programmed to check whether the replication process has already been completed or not when the program starts from scratch. This is important because the program requires this data and cannot wait for the next replication process, which only happens once a day in the morning. By checking the replicated table, the script ensures that the data is readily available for use, and in case it is not, a manual replication will be done.
Another important task of the "query_functions" script is to manage the storage space used by the database. Specifically, it is designed to remove all rows of data in the 'router_log' table that are older than six months. This ensures that the database only stores data from the last six months, which is typically the most relevant and useful for analysis. By removing older data, the script frees up disk space and helps to optimize the performance of the database.

The **"DBClass"** script, also known as the Forth script, plays a crucial role in managing the database connection and performing various actions on the database such as 'Select', 'Insert', 'Update', and 'Delete'. This script functions entirely independently from the other scripts, which allows for flexibility and ease of use. In the event that the decision is made to change the database, for example from MySQL to SQL, there is no need to make changes to any other scripts. The only necessary action is to update the functions within the "DBClass" script to be compliant with the new database, ensuring that everything continues to work seamlessly. This design choice allows for easy maintenance and scalability of the overall system.

Monitoring Service

To ensure that the monitoring system runs continuously and automatically starts after each reboot, a service called **"monitoring-service.service"** has been created on the Linux server machine. This service is responsible for running the "main.py" file when the operating system boots up. By using this service, we can ensure that any interruption or reboot does not affect the monitoring system.

To start or stop the service manually, you can use the following command:

```
sudo systemctl start monitoring-service.service
sudo systemctl stop monitoring-service.service
```

To check the state of the system, you can use the command:

```
sudo systemctl status monitoring-service.service
```

If the service is running, you can see a Python process for the main.py file in the server's process status by using the below command on the server.

```
ps aux | grep main.py
```

3.3.2 Data Cleaning

Ensuring the integrity of the input data is an essential step before inserting it into the database. To achieve this, various control functions are applied to the input lines of each file. One example of this is the validation of the router name, MAC address, and IP address to ensure that they are not null values.
In addition, the format of the IP address is also checked to ensure it conforms to the appropriate standard. Here is a piece of code that checks the validity of IPv4 addresses.

```
import re
regex = "^((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.)
{3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])$"
if not (re.search(regex, ip_param)):
  lOperation = False
```

This code checks if a given string ip_param is a valid IPv4 address using regular expressions. The code first imports the re module, which provides support for regular expressions in Python. It then defines a regular expression pattern regex which matches a valid IPv4 address.
The regular expression pattern checks that the IP address consists of four octets

separated by periods, with each octet being a number between 0 and 255 inclusive. The regular expression uses the pipe | character to match one of several possible patterns for each octet, allowing for leading zeros to be omitted in single-digit octets. The code then uses the re.search() function to search for a match between the regular expression pattern and the ip_param string. If a match is not found, it sets the value of lOperation to False, which suggests that the given ip_param is not a valid IPv4 address.
The format of the MAC address cannot be controlled as it is hashed, but the length of the value is checked to ensure it is valid. These control functions serve to ensure that the input data is accurate and complete before it is added to the database, thereby maintaining the integrity of the data stored in the database.

Corrupted Lines

ARP tables are an important tool for detecting network attacks or attempts, and as such, corrupted or incomplete lines of data should not be removed. A corrupted or incomplete line caused by an incomplete ARP association refers to a line that contains invalid or incorrect data, such as a misspelled MAC address or a line that is missing either the IP address or MAC address.
These lines in an ARP table can be caused by various factors, such as failure to receive complete data during the ARP process, errors in network configuration, hardware issues, or software bugs. For example, a hardware failure in a network switch could cause corrupted data to be stored in the ARP table. Similarly, a software bug in a device's networking stack could cause incorrect data to be sent to the ARP table.
Corrupted or incomplete lines in an ARP table could potentially be a security issue because they can lead to incorrect or unauthorized communication between devices on a network. For example, if an attacker is able to corrupt an ARP table entry, they may be able to redirect network traffic to a malicious device, intercept sensitive information, or launch other types of attacks. Incomplete ARP table entries can result in devices not being able to communicate with each other, which could cause security issues such as denial-of-service (DoS) attacks or the inability to monitor network traffic effectively.
Here are some examples that have been identified in the input data.

```
151060845 000000000000 130.192.186.78 ?
151060845 000000000000 130.192.186.87 dynamic
151060845 000000000000 130.192.186.116 dynamic
151060845 000000000000 ?  ?
151060845 ?  ?  ?
151060845 ?  ?  ?
```

```
?  ?  130.192.2.73 dynamic
?  ?  130.192.2.111 dynamic
?  ?  ?  dynamic
?  ?  ?  dynamic
```

The script handles this by saving all corrupted lines in separate files. For example, when an input file is received from a specific route, the system checks the data, and if any corrupted lines are detected, it creates a new file with the same name but marked as corrupted, and stores those corrupted lines in the new file for further investigation. This allows the system to generate statistics and derive useful information, such as an increase in incomplete ARP requests, from these files.

3.3.3 Statistics Extraction

During the process of ARP table analysis, several statistics are derived to gain insight into the network situation. Here's a brief explanation of each of the statistics that are obtained from input data for each chunk of 15-min data.

Count of all devices: This statistic gives the total number of devices that have communicated on the network and are present in the ARP table.

Number of distinct MAC addresses: This statistic gives the total number of unique MAC addresses found in the ARP table. This can be useful in identifying devices that have multiple MAC addresses.

Number of distinct IP addresses: This statistic gives the total number of unique IP addresses found in the ARP table. This can be useful in identifying devices that have multiple IP addresses.

Number of corrupted lines in each chunk: This statistic gives the number of lines in each chunk that are corrupted or invalid. This can be useful in identifying issues with the network devices, security threats, or the protocol itself.

Number of duplicated IP addresses: This statistic gives the number of IP addresses that are associated with more than one MAC address (repetitive or distinct) in the ARP table. This can be useful in identifying potential IP address conflicts.

Number of duplicated MAC addresses: This statistic gives the number of MAC addresses that are associated with more than one IP address (repetitive or distinct) in the ARP table. This can be useful in identifying potential MAC address conflicts.

Number of equal lines in the routers: This statistic gives the number of identical lines found in the ARP tables of different routers.

Number of IPs with multiple MACs: This statistic gives the number of IP addresses that are associated with more than one distinct MAC address in the ARP table. This can be useful in identifying potential issues with network devices or misconfigurations.

Number of MACs with different IPs: This statistic gives the number of MAC addresses that are associated with more than one distinct IP address in the ARP table. This can be useful in identifying potential issues with network devices or misconfigurations.

In the example below, the query calculates the number of identical lines among all input lines in each chunk. Therefore, the variable "df_count_equal_line_in_chunk" contains a value that indicates the number of rows that are identical across all routers in the ARP table for each chunk. This value will be stored in the specific column of a table which will be explained in the section 3.3.4.

```
df_count_equal_line_in_chunk = db_object.select(
(SELECT date_time, ip_address , mac_id,
COUNT(mac_id) AS count_equal,
GROUP_CONCAT(router_name SEPARATOR ; )
FROM router_log
WHERE date_time = ' + full_date + ''
GROUP BY ip_address, mac_id
HAVING count_equal > 1 and COUNT(ip_address) > 1)
AS count_equal_line_in_chunk,

date_time, count(*) AS count_equal_line',
where=None,
orderBy=None,
groupBy='date_time' )
```

The SELECT function on db_object is defined in the DBClass script (detailed in section 3.3.1) as shown below, and it receives these values as input.

```
def select(self, table, columns, where=None, orderBy=None,
groupBy=None, limit=None)
```

Overall, these statistics can help network administrators to identify issues with network devices, misconfigurations, potential IP or MAC address conflicts, and issues with the network topology or routing protocols. All these statistics are visualized in time series in different graphs.

3.3.4 Storing the Data

Once new files for a new chunk of data are detected by the script, raw data will be checked by the cleaning process. Once the cleaning process is done, the raw data is stored in the database. This allows for easy access and manipulation of the data in the statistics extraction process.
Derived statistics are also stored in the database for each chunk of data. This information can be used to provide valuable insights and visualizations of the data, such as the number of devices connected over time or the number of IP and MAC addresses duplication and etc.
All of the data storage for this project is being done in a MySQL database that is specifically designed for this purpose. The database contains several tables, each with a specific purpose which is described in the following section.

Database

In this thesis, SQLite was initially used as the database to store and manage the data. The script was compatible with this database, but after importing a certain amount of data, the database crashed. As previously mentioned, the volume of data is huge and the database needs to be able to handle this volume of data. Therefore, as a second choice, MySQL was used as a more powerful alternative to SQLite.
As MySQL is a more powerful and robust database management system, that can handle large and complex data sets with high performance and efficiency. This change enabled the script to process and store a large amount of data without crashing and also allowed for better performance and faster query execution.

Additionally, applying functions such as indexing specifically for different queries, was crucial to optimize the database and increase the speed of queries, this way the data can be easily searched, filtered, and visualized.

Tables

In general, Three tables are created to store the data in the database. The first table is called **"router_log"** and it stores all the cleaned data imported from the routers. It keeps information for each line of the ARP tables of the routers, including the date and time of the associated chunk of data, the MAC address, the IP address, the router's name, and some information about that Ip derived from the metadata table such as type and class of the device, departments that device is used, device name and type of operating system. This table allows for easy access and manipulation of the raw data in the future.

The second table is called **"statistics_log"** and it stores all the statistics that are

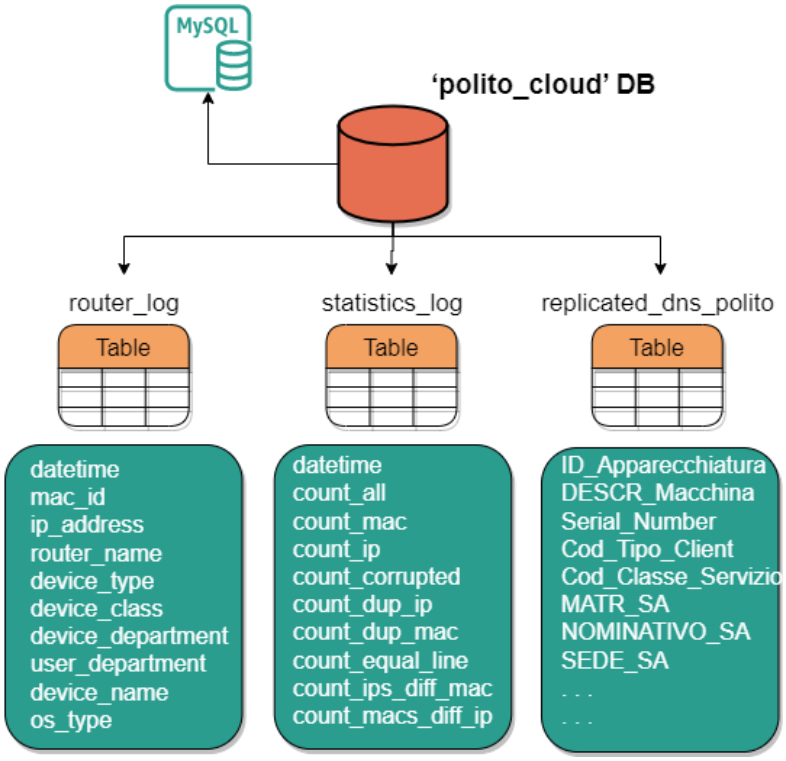

Figure 3.4: Database and Tables

derived by the script from the input data. This table keeps information for each chunk of data including the date and time of the chunk, the count of all devices, the number of distinct MAC addresses, the number of distinct IP addresses, the number of corrupted lines in each chunk, the number of duplicated IP addresses, the number of duplicated MAC addresses, the number of equal lines in the routers, the number of IPs with multiple MACs and the number of MACs with different IPs.
This table provides valuable insights into the data, such as the number of devices connected over time. It also keeps track of corrupted lines, duplicated IPs and MAC addresses and equal lines among the routers, which can help to understand the data better and spot any issues or problems.

The third table, **"replicated_dns_polito"**, is a destination table for the data that is replicated from the metadata table of the polytechnic database. This table has 53 columns, and the values in these columns are updated once a day during the replication process. In other words, this table is designed to store a copy of the metadata data from the polytechnic database, and this copy is updated on a daily basis to ensure that the data is accurate and up-to-date.

3.4 Visualization

When the statistics are derived and stored in the database, it is important to be able to visualize them in a clear and understandable way. This allows for easy analysis and interpretation of the data, which can help to identify trends, patterns, and potential issues. One of the most popular and powerful tools for data visualization is Grafana.
Grafana is a well-known and widely used open-source data visualization tool that can be used to create a variety of different visualizations. It is particularly well-suited for visualizing time series data, making it an ideal choice for monitoring and analysis. One of its key features is its great compatibility with MySQL, which makes it an excellent choice for visualizing statistics stored in a MySQL database.

To visualize the statistics, some dashboards are created on Grafana. These dashboards are used to present the data in an easily understandable way, making it easy to identify trends and patterns. The dashboards can be configured to show real-time data, which allows for easy monitoring and analysis of the statistics as they change over time.
Grafana also offers a wide range of options for customizing the look and feel of the visualizations and dashboards. This allows you to tailor the visualizations to suit your specific needs, and to make them as clear and informative as possible. Additionally, Grafana has the capability of setting alerts, which can notify users when certain conditions are met, such as when a value exceeds a certain threshold which has been used for some statistics.

3.4.1 Dashboards' Type

In general, two types of dashboards are created: the first one is called a **statistic** dashboard, and the second one is a **query-based** dashboard. For some statistics, as mentioned in previous sections, these statistics can be derived in the process of analyzing the input data, and the results for each chunk are stored in the database. In statistic dashboards, Grafana retrieves those statistics and presents them in different visualization formats for different periods of time. As the system receives new data at each chunk interval, these dashboards provide real-time monitoring of the statistics.
For other types of statistics, the system cannot derive them during the process of analyzing input files. This is because we need to derive the values by comparing different parameters on different chunks of data. Therefore, firstly, the data should be stored in the database, and then the system can extract statistics based on the values of parameters on different chunks of data. That's why query-based dashboards exist. In these dashboards, different queries are provided according to the dashboard, and these queries receive information such as the time period or

required IP address or MAC address and use these input values in the queries to provide statistics. For example, we can retrieve the list of equal lines in the ARP table or see the list of IPs that have been seen with different MAC addresses. All these dashboards and the statistics have been presented in Section 4.

3.5 Resource and Demand

The system has been set up on a virtual machine (VM) with the following specifications. The CPU model used is Intel's Xeon E5-2640 v4, which operates at a base clock speed of 2.40GHz.The VM is configured with 4 CPU cores, which allows for the efficient processing of tasks. In terms of memory, the system has been allocated 16 GB of RAM, providing ample space for running applications and services. The disk capacity of the VM is 160 GB, providing sufficient storage space for data and system files.
The VM's hard disk is a Solid State Drive (SSD), which offers significantly faster read/write speeds compared to a traditional hard disk drive (HDD). This makes it an excellent choice for running a database, as it can handle a large number of read/write operations quickly and efficiently.
The operating system installed on the VM is Ubuntu 20.04, which is a popular distribution of Linux known for its stability and ease of use, and compatibility with required libraries of Python that have been used in scripts.
With these specifications, the system should be able to handle a variety of tasks effectively and efficiently. As previously mentioned, the system is dealing with a large amount of data that is imported in 15-minute chunks. This presents a challenge in terms of resource demand, as there are different phases that need to be considered. The first phase is importing, analyzing and storing of the data, and the second phase is visualizing and querying the data.

The first phase, which involves importing and analyzing and storing the data, is a critical phase because real-time monitoring is needed. Therefore, it is important that this phase is completed as quickly as possible. With the given VM described above, the process of detecting, reading the new input lines, analyzing, and storing raw and statistical data into the database takes about 1 minute and 20 seconds. This amount is measured in the morning, which is one of the periods when the system experiences a high number of connected devices. A high number of devices means a high number of rows in the ARP table and high demand for resources to analyze them. Of course, in other periods, the system is under less load, and this time is less than one minute. This time value shows that the system can handle a lower chunk interval such as 10-min or 5-min, as all the input files are analyzed in a short amount of time.

The second phase, which involves visualizing and querying the data, is also important. When querying the data for a long period of time, such as several months, it is crucial that the queries are resolved quickly. Otherwise, it will negatively impact the ability to effectively visualize the data. It should be considered that the system and the database are under load for the first phase, and the second phase also puts a burden on the system and database. Some queries require joining multiple tables to retrieve the desired result, while others have a lot of arguments that need to be checked to retrieve the result. All of these factors make the process heavy, and they need to be considered in the system.
To improve the performance of the system, the database has been configured appropriately. Indexing the tables of the database was one of the useful tunings that really make the process of queries faster. Indexing depends on the queries and the parameters that need to be considered in that query. For example, if a query is checking many values of two or three parameters, a separate indexing has been set specifically for that query. These optimizations have helped to reduce the query resolution time, allowing the system to handle the heavy workload more efficiently.

Chapter 4

Result and Discussion

The system receives data from routers' ARP tables at regular intervals and processes it to derive various statistics. As soon as the system receives new files, it begins to process them. The system goes through all the lines of the new files and derives statistics, which are then stored in a database. These statistics are then presented in various time series charts on different dashboards on the Grafana platform. In general two types of dashboards are created which are explained in the following sections in detail. Additionally, the system includes an alert system that enables users to monitor the data and keep track of any issues that may arise. In this section, the results obtained from this system are presented in detail and discussed, providing users with an in-depth understanding of the data. The statistics derived from the system can provide valuable insights into network performance and help identify any issues that need to be addressed.

4.1 Statistics Dashboards

The monitoring system is designed to observe statistics for each chunk of data. This means that for every interval, a set of values for these statistics are stored in the database. To present this information in a more visually appealing and user-friendly way, two separate statistics dashboards have been created. These dashboards contain different charts and information, as shown in Figure REF. The purpose of these dashboards is to provide users with a quick and easy way to view the important statistics related to the processed data, which can help them make informed decisions and identify trends or anomalies. By providing these dashboards, the system can help users to better understand the behavior of the data and make more informed decisions based on the insights provided by the statistics.

4.1.1 Dashboard 1

The first dashboard depicted in Figure 4.1 provides several key pieces of information.

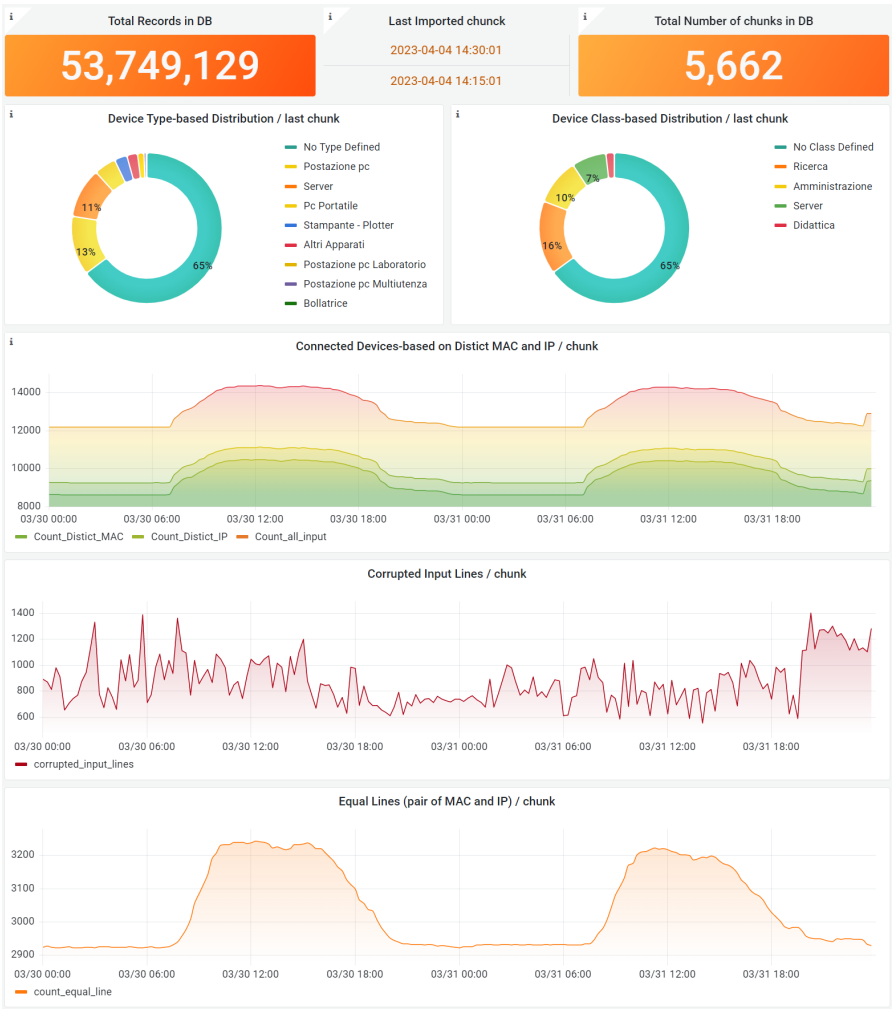

Figure 4.1: Statistics Dashboard 1

The Total Number of Records in the database refers to the number of rows stored from the ARP table of the routers. As of the moment when this figure was taken, there were approximately 53 million rows of data stored. Every day, approximately 1.5 million new rows of data are added to the database, although this number may vary depending on a working day. The system has been configured to store ARP table values for six months, meaning that it is expected to store almost 270 million rows of data during that time period.

The Last Imported Chunk shows the date and time of the most recent interval of data that the system has received and processed. This real-time monitoring allows users to see the latest data as soon as it is available.

The Total Number of Chunks in the database indicates the number of data intervals that have been imported into the system. In this case, there are 5662 chunks of data stored, which means that there are corresponding statistics values for each of these intervals.
The system is set up to receive ARP table data every 15 minutes. This means that we'll have four chunks of data every hour, and 96 chunks in a day. Unlike the total number of records in the database, there's no limit on the number of chunks, and the system will keep all the chunk information. This information includes all the columns of the 'statistics_log' table, which is explained in section 3.3.4.
The system is capable of handling shorter time intervals, since it can process and analyze input lines for each chunk in just 1 minute and 20 seconds. Therefore, the system can handle intervals as short as 10 minutes or 5 minutes, depending on the administrator's needs. For example, for some statistics, the variation of values isn't meaningful in 5-minute intervals, and collecting information at that interval would be a waste of space. However, for other statistics such as the number of corrupted lines, which provides valuable insight into network security, a shorter interval would be useful.

The Device Type-based Distribution chart which is depicted in Figure 4.2 shows the number of devices of each type that are connected to the routers. For example, the chart may indicate that 13 percent of the devices are PCs and 11 percent are Servers. This information is obtained from a metadata table that is replicated from another database. The system checks the detected IP addresses from input files of the ARP tables against this table and extracts other relevant information, such as the device type, which is then written into the system database. By doing this, the system is able to provide statistics on device types and other factors.
This chart serves as a valuable tool for administrators as it allows them to gain insights into the usage patterns of connected devices. By analyzing the data presented in the chart, administrators can easily identify the most commonly used type of device on their network. As time passes, the chart can also help administrators identify changes in device usage patterns. For example, if the number of connected fixed PCs is observed to be increasing, administrators can prioritize infrastructure changes or enhancements related to this device type. This information can be used to inform decisions related to network management and optimization, with a focus on improving the network's ability to accommodate and support the most commonly used and/or changing device types.

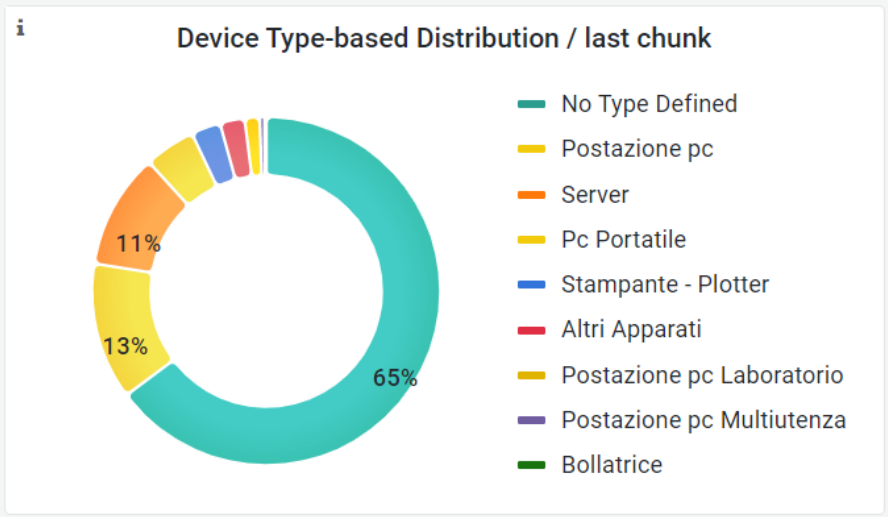

Figure 4.2: Device Type-based Distribution

The data presented in this text shows that a high percentage of devices are categorized as "No Type Defined". This can occur for two reasons: either the metadata table for this column has a "NULL" value, or the IP address being considered does not yet exist in the metadata table and has not been entered by the operator. While this chart provides administrators with a view of the distribution of devices across departments, it would be more precise if the metadata table were fixed and empty rows were completed for related columns. By addressing these issues and completing the metadata table, administrators can gain a more accurate understanding of device usage patterns and distribution across the network. This information can then be used to make more informed decisions related to network management and optimization, leading to a better user experience and improved network performance.

The Device Class-based Distribution chart provided in Figure 4.3 shows the percentage of devices for each class of device that has been defined in the metadata table. The classes include research, server, administration, didactic, and "No Type Defined". For the last imported chunk of data, which is shown in the chart, approximately 16% of devices were classified as research, 10% were administration devices, 7% were servers, and 2% were related to didactic.

The chart mentioned provides valuable insights to administrators by allowing them to understand the distribution of devices across different classes. By analyzing the data presented in the chart, administrators can easily identify which device classes have more connected devices and make informed decisions related to network management and optimization. For example, if the research class has a higher number of connected devices, administrators can use this information to prioritize

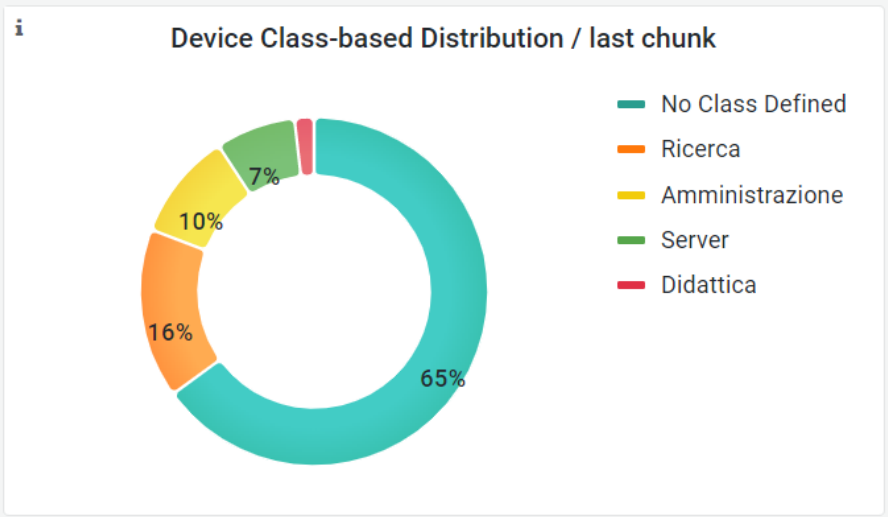

Figure 4.3: Device Class-based Distribution

infrastructure improvements or replacements related to this class. Additionally, the chart can help administrators to understand which device specifications should be more considered in the event of an upgrade or replacement. By considering the device specifications that are most commonly used across different classes, administrators can ensure that new devices are optimized to meet the needs of the network and the users.
As we can see from the chart, a high percentage of devices are categorized as "No Type Defined". This can occur for two reasons, as explained in the previous chart: either the related column on the metadata table has a "NULL" value, or the IP address being considered does not yet exist in the metadata table. By addressing these issues and completing the metadata table, administrators can gain a more accurate understanding of the classes of devices on the network and their distribution.

The Connected Devices Based on Distinct IP and MAC chart displays three values for each chunk of data. As we can see in Figure 4.4, The First two values represent the number of distinct IP and MAC addresses, respectively. These values exclude duplicates and corrupted lines since they are observed after detecting and removing incomplete or erroneous lines. The other value represents the total input lines, which includes corrupted and duplicated lines from the ARP table of the router. It means that this value includes all the input lines from ARP table of the routers and this is the reason for being higher than the other two values.
This figure presents a time series of values for a two-day period. Upon examining the chart, one immediately notices the variance between different data points. Specifically, by comparing the total number of input lines to the number of distinct

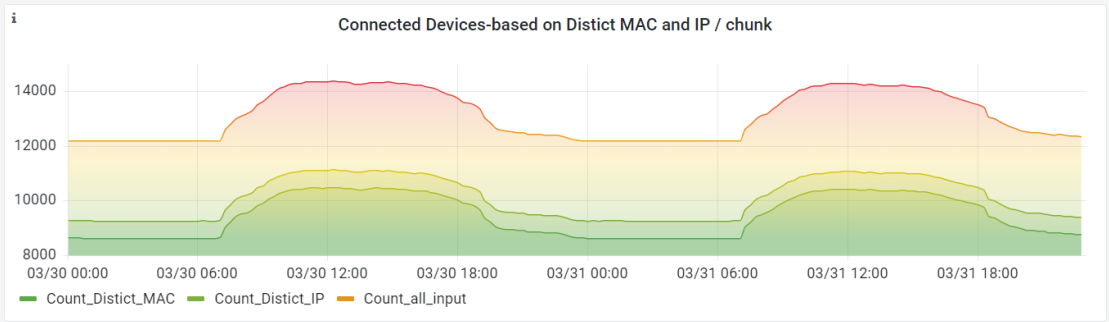

Figure 4.4: Connected Devices based on Distinct MAC and IP

IP and MAC addresses, it becomes apparent that there are always duplicated and corrupted lines present. However, it is noteworthy that the difference between these values has remained relatively consistent. Therefore, any significant fluctuation in this difference should be considered a significant event, indicating a substantial change in the system. This information can be used by administrators to identify potential issues and address them proactively, ensuring optimal network performance and user experience.

One observation from this time series is that the system exhibits consistent behavior throughout the day. At night, the number of connected devices remains relatively stable, with only minor fluctuations. However, it is important to note that the number of devices does not tend to approach zero during these periods, as there are many servers, network equipment, and research devices that remain connected at all times. The minor changes in the number of devices, typically an increase or decrease of one or two, may be due to device or network service restarts during the night.
In the morning, starting at 7 am, the number of connected devices begins to increase as people arrive at their offices and connect their devices to the network. The chart indicates that the highest number of connected devices occurs between 12 pm and 2 pm, after which the number starts to decline again. The lowest number of connected devices is typically observed during the night.

Another important consideration is the behavior of the system during weekends. Figure 4.5 shows that the number of connected devices during this period remains stable. For example, on the weekend of March 18th and 19th, the number of connected devices remained at the lowest level observed during each night. This suggests that there is little or no activity on the network during weekends, which may be useful information for network administrators.

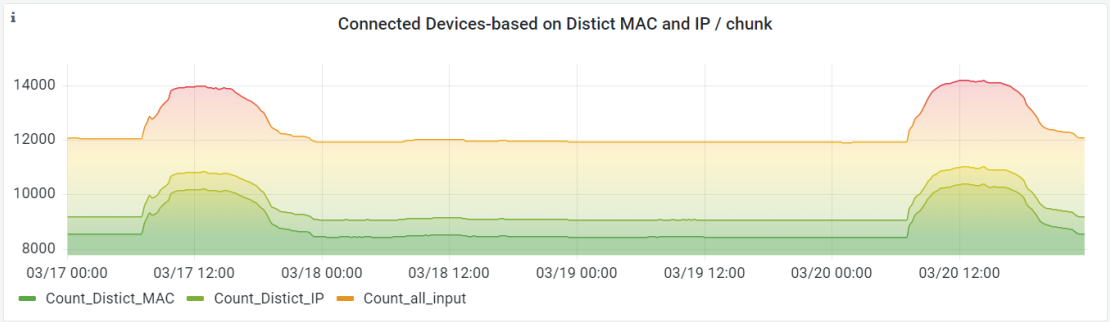

Figure 4.5: Connected Devices based on Distinct MAC and IP - Weekend

Another observation from this time series can be the examination of values and trends over a long period. Figure 4.6 presents all existing values in the database at the time the figure was taken, which shows the trend of connected devices during this period. The repetitive behavior of day and night and weekends is obvious, but what is also important is the slight increase in the number of connected devices over this period.
The total number of devices on the first day of this figure was under 14,000, but this value increased during the period shown, with the system detecting more than 14,000 devices in the last days of the figure. This demonstrates that the number of connected devices is increasing. Of course, a longer period would provide a better understanding of this trend. Such information helps administrators to forecast the trend and anticipate future needs. For instance, based on this amount of increase, one could estimate the number of connected devices likely to be present by the end of the year.

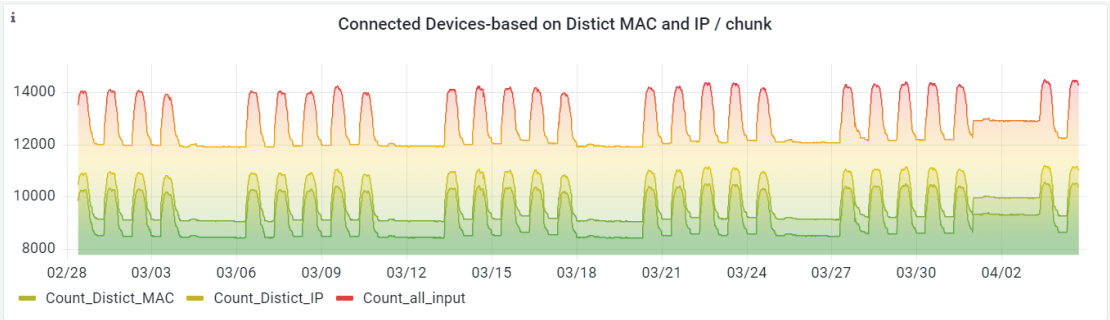

Figure 4.6: Connected Devices based on Distinct MAC and IP - Long Period

Monitoring these values is crucial as it enables the security team to keep track of the number of connected devices. Any significant changes in these values, as well as the differences between them, should be closely monitored as they may indicate

potential issues that need to be addressed.

The Number of Corrupted and Incomplete Lines in each chunk of data is another chart in the first statistic dashboard. As explained in section 3.3.2, there are lines in each input chunk that are either incomplete or corrupted for various reasons. Figure 4.7 shows the fluctuation of corrupted or incomplete input lines in the ARP table over a four-day period, including the weekend (March 18th and 19th). As we can see, there is a lot of fluctuation in this value, with some peaks such as 2,430 and 2,403 during this period. Even on weekends when the system experiences the lowest number of connected devices, it can be observed that these corrupted or incomplete ARP associations are being generated by devices that are already connected. Therefore, it is important to investigate the root cause of this issue to understand why it is happening.

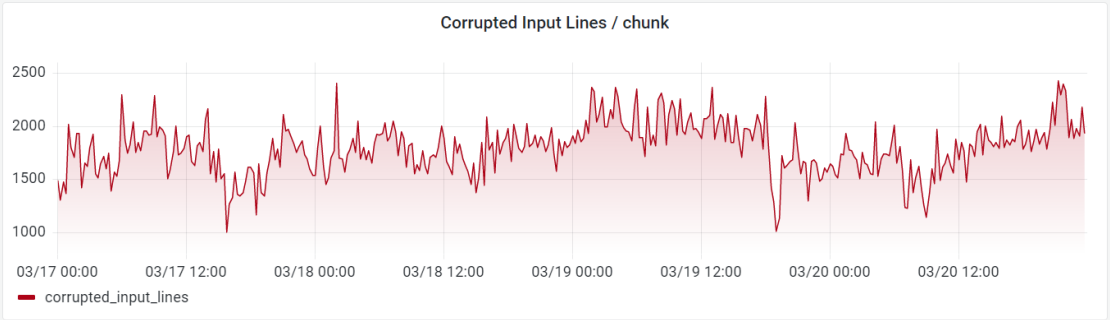

Figure 4.7: Corrupted and Incomplete Input Lines

As explained briefly in section 3.3.2, Incomplete lines in an ARP table can occur for various reasons. One common reason is that the device associated with the IP address is not currently active on the network. This can happen if the device is powered off or disconnected from the network. When a device is powered off or disconnected from the network, it no longer communicates with other devices on the network. In order for a device to be included in an ARP table, it must first communicate with the network and register its MAC address with the router. When a device is powered off or disconnected from the network, it no longer sends or receives network traffic, which means that the router has no way of learning or updating its MAC address in its ARP table. As a result, the ARP entry for that device becomes incomplete, as the router has an IP address for the device, but does not have the corresponding MAC address.
Another reason for incomplete entries is that there may be issues with the network such as network congestion or malfunctioning network hardware preventing it from properly registering its MAC address with the router. In case of network congestion, high levels of network traffic can cause packets to be dropped or delayed, which may prevent a device from properly registering its MAC address with the router.

In case of malfunctioning network hardware, faulty network hardware, such as a malfunctioning network adapter or cable, can cause intermittent connectivity issues that prevent a device from properly communicating with the router and registering its MAC address in the ARP table.
Another reason can be software-related issues. This includes a variety of cases such as a firewall or security software, Network configuration software, Network drivers, and software that generates traffic such as port scan software. Some firewalls or security software may block the transmission of ARP requests or responses, which can prevent a device from properly registering its MAC address with the router. Some network configuration software may misconfigure network settings or cause network connectivity issues. Outdated or malfunctioning network drivers can cause intermittent connectivity issues and prevent a device from properly registering its MAC address with the router.
A port scanning software scans a network to identify which devices are connected to the network and which ports on those devices are open. This process involves sending packets to a range of IP addresses and ports on the network, which may cause some devices to become overwhelmed with traffic or block incoming packets altogether. When a device is unable to respond to ARP requests due to the high volume of traffic caused by a port scanning software, the router may not be able to learn the MAC address of the device, resulting in an incomplete entry in the ARP table. Similarly, if a device blocks incoming traffic from the port scanning software, the router will not be able to learn the MAC address of the device, again resulting in an incomplete entry in the ARP table.
This chart and its corresponding value are critical for evaluating network performance and security. By monitoring this value, network administrators can identify the underlying causes of issues that may arise in the network. Additionally, the value is essential for the security team, as it enables them to quickly detect and respond to any potential attacks.

The Total Number of Equal Lines is the last chart of this dashboard that reports the total number of equal lines detected in each chunk. Equal lines refer to instances where one IP and its corresponding MAC address are identified multiple times across different routers. As we can see from the chart 4.8, the total number of equal lines exhibits a consistent pattern over the course of the day. During the night, this value is the lowest and remains relatively stable. However, starting from 7 am in the morning, at the beginning of the workday, the value begins to increase. The chart shows that the total number of equal lines reaches its highest point around 12-2 pm, and then starts decreasing again.
This chart is also important since it reports these lines. The presence of equal lines in the ARP table is more likely to be related to the misconfiguration, network topology, routing configuration, and traffic patterns, or the use of virtual networks connected to different routers. In a virtual network, multiple virtual machines

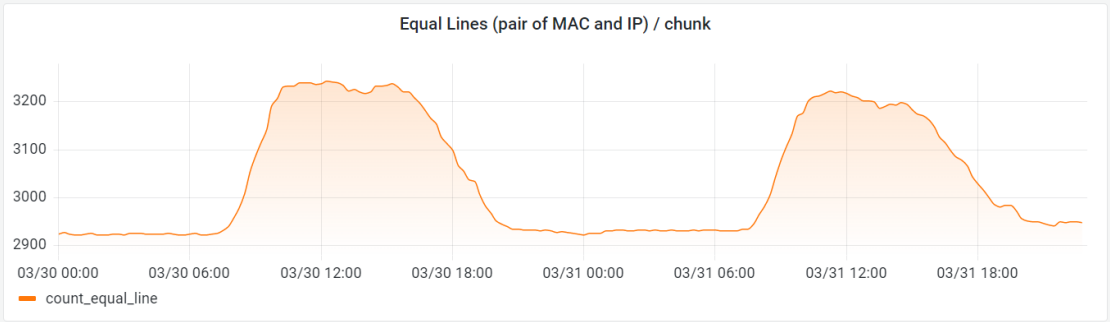

Figure 4.8: Number of Equal Lines in the same chunk

(VMs) may share the same IP address, and the physical host that hosts the VMs may have multiple physical network interfaces that are connected to different physical switches or routers. When multiple physical network interfaces are used to connect the same virtual network to different physical switches or routers, the ARP table on each router may have multiple entries for the same IP address, each learned from a different interface or next-hop router. This can result in equal lines in the ARP table of some routers, especially when the network is configured to use Equal-Cost Multipath (ECMP) routing or load balancing.
The total number of equal lines during the night remains almost stable, which could suggest that the virtual network is a reason behind this value. However, since the value increases during the day when the number of connected devices (usually by users) is higher, there may be other reasons at play, as previously mentioned.

4.1.2 Dashboard 2

Figure 4.9 displays the second statistics dashboard, which provides various information.

The Department-based Distribution chart presents the number of devices that are connected to the network from different departments. This chart displays the distribution of connected devices in each department for each data chunk, and it provides valuable information for administrators. Figure 4.10 depicts this distribution in the last chuck when the figure was taken. As we can see the chart shows that IT Department has 895 connected devices, while Departments DET and DIATI have 851 and 351 connected devices respectively, based on the latest data. For instance, administrators can use this chart to monitor the departments with the highest active users, which may generate more traffic than other departments. It is also useful to identify departments that require improvement or priority attention in terms of network infrastructure.

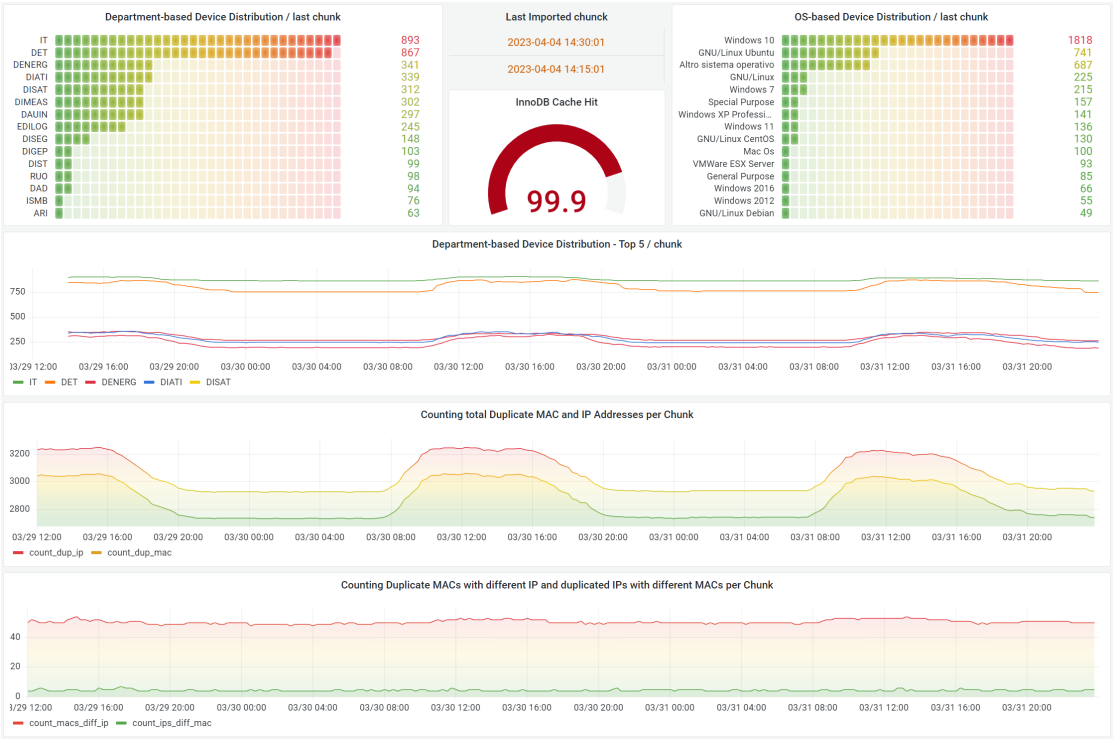

Figure 4.9: Statistics Dashboard 2

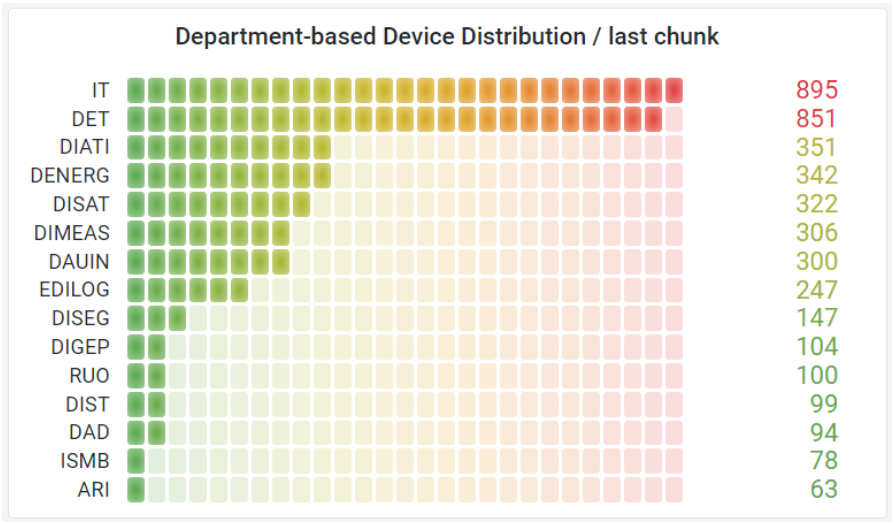

Figure 4.10: Department-based Devices distribution

Moreover, this chart enables administrators to observe different network aspects for each department. Any significant fluctuations in the number of connected devices

for each department should be monitored and checked. Since all departments have a relatively stable and predictable number of devices that they use every day, any deviation from this pattern should be investigated to ensure optimal performance. Another aspect can be controlling the connectivity of each department. For example, If the IT department or any other department that is always visible in this chart is not displayed in the chart for the latest data chunk, this may indicate a connectivity problem within that department. Thus, this chart helps control the connectivity of departments across different campuses.
The department value we need is obtained from the metadata table. When analyzing the ARP table for IP and MAC addresses, each IP address is checked in the metadata table to retrieve the corresponding department value. This value is then taken and entered into another table along with additional information for that IP and MAC address.
It's important to note that some IPs do not have a value or have a "NULL" entry for the department column in the metadata table. This results in a high number of entries showing "No Department Defined" in the chart. As a result, we did not include this value in the chart since the difference between it and the next highest department value was significant and would have made it difficult to visualize the distribution accurately.
However, if we were to address this issue in the metadata table by adding values for all the IPs, we would be able to create a more precise chart. This would allow us to better understand and make informed decisions based on the data.

The Operating System-based Distribution chart represents the diversity of operating systems installed on the connected devices in the latest imported chunk. This chart and its associated statistics also are generated by combining metadata information from the metadata table and input ARP table files.
Figure 4.11 shows the number of connected devices based on the operating system for the latest imported chunk at the time the figure was taken. The chart presents a variety of devices, and we can see that the most commonly used operating system among connected devices is Windows 10, with a count of 1,822. Linux Ubuntu is the second most used with 742 devices. This chart provides valuable information on connected devices at a different level.

For example, we can observe the distribution of devices between Linux and Microsoft-based systems. Another important piece of information that we can obtain from this chart is the number of devices running old operating systems. For instance, we can see that 217 devices are running Windows 7, which is an outdated version that Microsoft no longer supports. As a result, Microsoft does not release new security and fixing patches for this version, making it a vulnerability and a potential target for security threats and attacks.
Similarly, the chart can help administrators identify vulnerabilities in specific operating systems, such as Mac OS. For example, if a vulnerability is detected in a

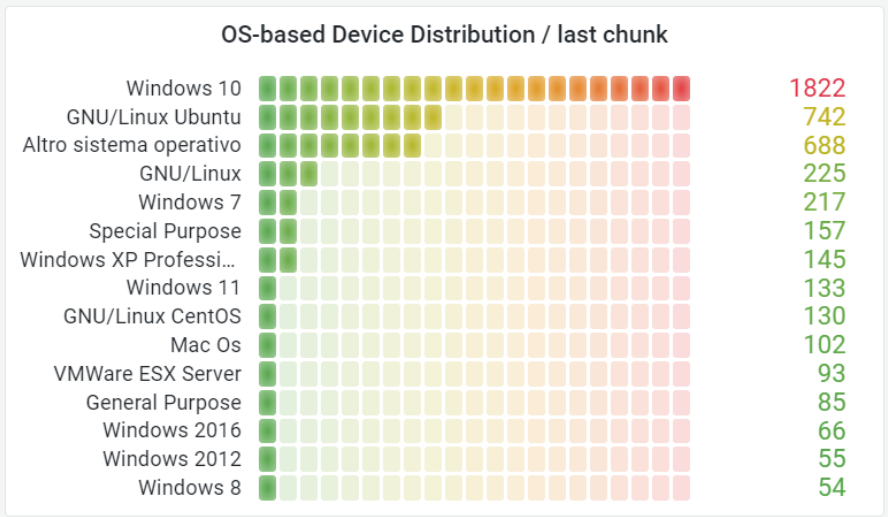

Figure 4.11: Operating System-based Devices distribution

particular version of Mac OS, this chart can help administrators quickly determine the number of systems running that operating system and take immediate action to patch or address the issue.
The chart in Figure 4.11 is a valuable resource for administrators to identify potential vulnerabilities in connected devices, particularly those running outdated operating systems. It can help them plan and execute timely updates and patches to improve system security and prevent potential security threats and attacks.
Like the previous chart (Department-based Device distribution), this chart also does not include devices with an undefined operating system to improve the visualization and distribution of devices. The metadata table's required column for the type of operating system contains NULL values for many IP addresses. By completing these values with the appropriate operating system type, administrators can create a more accurate and precise chart, providing them with better insights for decision-making.
By ensuring that all devices' operating systems are correctly identified and included in the chart, administrators can gain a more comprehensive understanding of the distribution of connected devices and their operating systems. This, in turn, can enable them to make better-informed decisions regarding system updates and patches, helping to ensure better overall system security.

The Distribution of Devices in Departments over Time is the next chart as depicted in Figure 4.12 which features a time series chart that displays the number of connected devices from the top five departments with the most connected devices. This information is important because it helps administrators understand how device usage is distributed across the organization and identify potential areas for

improvement or optimization.

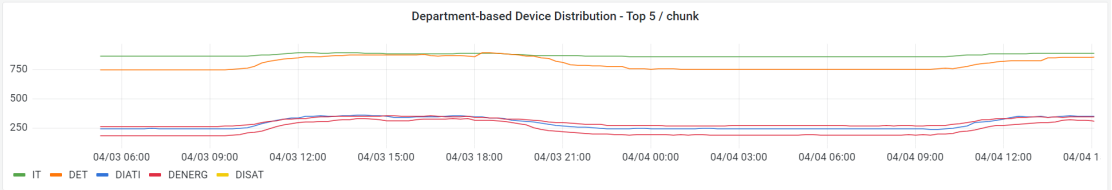

Figure 4.12: Distribution of Devices in Departments over Time - Top 5

To generate the data for this chart, the system queries the database in the background each time the dashboard is opened. The purpose of this query is to determine the top five departments based on the number of connected devices they have. The system then takes the results of those queries and feeds them into the time series chart, which displays a dynamic view of the connected devices over time. In other words, the chart will update automatically as new data becomes available, giving administrators a real-time view of device usage.
It's worth noting that each chunk of data on the time series chart represents a different value for each department. For example, the chunk value representing the IT department may be greater than the chunk value representing the DET department, indicating that the IT department has more connected devices. This visual representation of the data can be very helpful for identifying trends and patterns in device usage.
At the time the figure was taken, the top five departments displayed in the chart are IT, DET, DIATI, DENERG, and DISAT. However, this may change over time as device usage patterns shift or new departments come online. By providing a dynamic view of the data, the time series chart helps the administrators stay up-to-date with the latest trends and make data-driven decisions based on the most current information available.
Real-time monitoring is important, but long-term trend and pattern monitoring can provide valuable insights to administrators as well. By tracking the growth of devices in each department over time, administrators can identify patterns in device usage and make informed forecasts about future infrastructure needs. This can include predicting the need for additional network throughput and other necessary resources to support continued growth. By understanding these patterns, administrators can proactively plan for future demands and ensure that the organization is well-prepared for both near and distant future needs.
The reason why only the top five departments are included in this chart is due to the heaviness of the query required to generate the data. The query attempts to retrieve the number of connected devices for each department in each time chunk, by summing the devices associated with that specific department. The query

repeats this process for other chunks until the specified time period in the Grafana dashboard is fully covered. This process is also done for all other departments. If the time period is long, it can be very challenging for the database and server to handle the query. Therefore, only five departments are considered in this chart to avoid any potential problems with the system's processing and analysis of real-time files and other processes.
However, since the administrator may need to see the trend of other departments besides these five, a separate Query dashboard has been created for this purpose. In that dashboard, the administrator can select the desired department and view the trend and pattern for their desired time period. This dashboard is explained in section 4.2.4 and one example is depicted in Figure 4.18.

The Total Duplicated MAC and IP Addresses chart represents the total number of duplicated MAC and IP addresses for each chunk of data. When an IP address is detected multiple times across different routers, we refer to it as a duplicated IP, regardless of the MAC address. In the case of duplicated IPs, there are two possible scenarios for the MAC address. The first scenario is when the MAC address is the same for all replicated IPs, resulting in equal lines in the statistics. These equal line statistics can be found in Statistic Dashboard 1 in section 4.1.1 and it is depicted in Figure 4.8.
The second scenario is when the MAC address for those duplicated IPs is not the same, indicating that one IP has been detected with multiple MAC addresses. The total number of duplicated IP addresses, including both scenarios, is shown in Figure 4.13.
Additionally, the statistics for duplicated MAC addresses show the total number of duplicates, including MAC addresses that belong to the same IP address as well as MAC addresses that have multiple IPs associated with them.

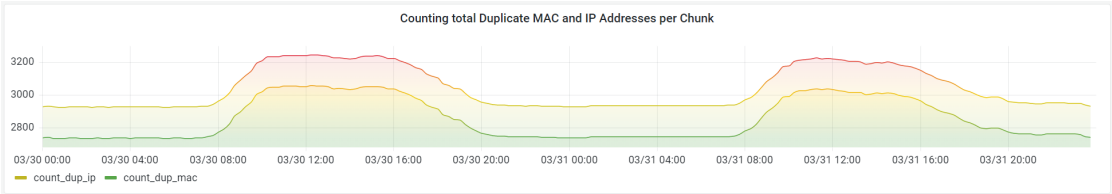

Figure 4.13: Total Duplicated MAC and IP Addresses per Chunk

Based on Figure 4.13, there are slight and repeated fluctuations in IP and MAC duplication. These values remain almost the same during the night, but start to increase at the beginning of the working day and decline towards the end. The figure also reveals that some of these duplicates are associated with always-connected devices like routers, servers, and other infrastructure devices. This could be due

to reasons such as using virtual networks on the same physical adapters or device misconfiguration, which needs to be examined in the network. The other part of the duplication that occurs during the day is likely related to devices connected by users, and it's more important to investigate since it affects users' connection. This chart provides valuable insights into the duplication of MAC and IP addresses on the system, allowing for effective monitoring.

Distinct Duplicated MACs and IPs is the last chart that represents the same concept as the previous dashboard, but it considers only the cases where an IP address is seen with different MAC addresses. If the MAC addresses for the same IP are the same (equal lines), this chart does not count those lines. Similarly, if a MAC address has multiple IP addresses, only distinct IP addresses are counted.
Figure 4.14 shows that the values for these duplications are much lower than in the previous chart, which presented total duplications. This means that most of the duplications are related to equal lines, and only a few are related to the case when the IP has been seen with multiple distinct MACs or, in the case of MAC duplication, when the MAC address has been seen with different IPs for several times. The chart shows that the variation is almost stable, with the system detecting 4 to 6 duplicates in the case of IP and 50 to 55 in the case of MAC duplication during these periods.
As explained before, this chart and the chart of equal lines in the first Statistics Dashboard, are parts of the total duplicated IP and MAC addresses that were described earlier. This chart provides valuable insights into the duplication and administrators can monitor this chart in real time and track the trend over a longer period of time. Additionally, a query dashboard is available to provide the administrator with a list of these duplications, which is explained in the section 4.2.1.

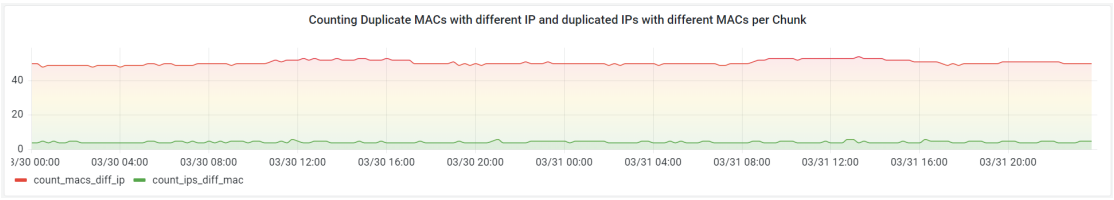

Figure 4.14: Distinct Duplicated MACs and IPs

4.2 Query Dashboards

The second type of dashboard is the query-based dashboard. They are created for statistics that cannot be derived during the processing and analyzing phase, as these queries require comparison with data from other time periods. These queries compare values over time for different purposes and provide results based on specific input values.
There are 10 query-based dashboards that provide various statistics. These dashboards are designed to allow the operator to interact with the system by providing input values, which the queries used to retrieve relevant results.

4.2.1 Dashboard 1 and 2

The first dashboard provides information about the IPs that have been seen with multiple MAC addresses in the same chunk of data. This dashboard receives the date and time as input and provides information about that specific chunk. In Figure 4.15, the given date and time were '2023-04-04 13:15:01'. The dashboard runs three different queries based on the input data. The first query shows the number of IPs that have been seen with multiple MAC addresses for the given chunk. As we can see, on the given date and time, five IPs were detected. The next query provides the list of those five IPs, so further investigation can be performed on them. The third query retrieves all the lines that have been counted from the 'router_log' which provides other information such as ID of the row in the table, date and time, name of the router, MAC address, IP address, operating system type, type of device, and so on.
The second dashboard receives the date and time as an input, similar to the first dashboard. However, this dashboard provides information about the MAC addresses that have been seen with different IPs. Similar to the previous dashboard, it provides the number of MAC addresses that have been identified, the list of them, and the complete rows of those MAC addresses in that specific date and time from the database.

4.2.2 Dashboard 3 and 4

The third and fourth dashboards work on duplicated IPs and MACs in the same chunk of data, respectively. These dashboards receive the date and time of the chunk as input and provide information in this regard. Figure 4.16 shows an example for duplicated MACs in the same chunk, where the dashboard receives the date and time as input and then runs two queries.

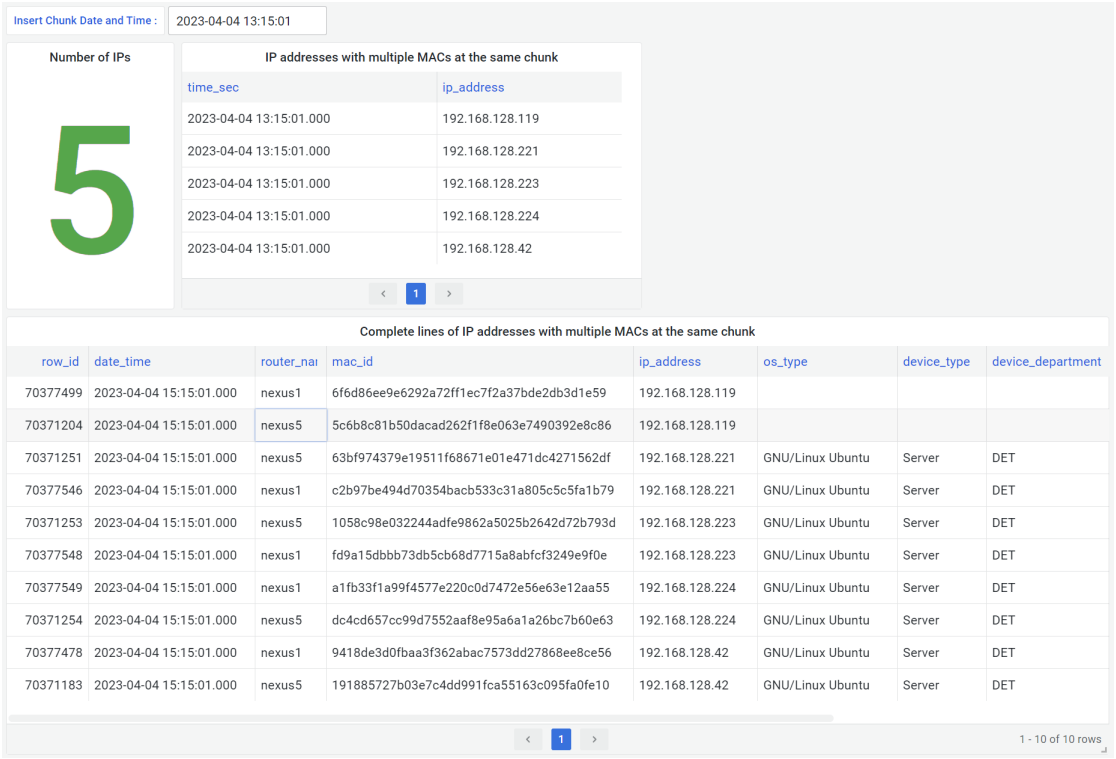

Figure 4.15: Query Dashboard - IPs with multiple MACs at the same chunk

Firstly, it retrieves the total number of duplicated MACs in the given date and time. Secondly, it provides the list of those MACs that have been detected, along with the number of occurrences and the list of routers that those MACs have been received from as input to the ARP table. In figure 4.16, the dashboard has received '2023-04-04 13:15:01' as input and presents the total number of duplicated MACs in that time, which is 3,111. We can also see the list of MACs in the second section along with the number of occurrences. As mentioned before, due to respect privacy, the MAC addresses are hashed in this project.

4.2.3 Dashboard 5

Dashboard five provides information related to identical lines in the same chunk of data. When one IP and its corresponding MAC address have been seen more than once, that line is considered identical. This dashboard receives the date and time as input and shows the total number of identical lines detected in all the routers for that chunk of data. Additionally, the dashboard performs a query and retrieves those identical lines from the database. In Figure 4.17, we can see that

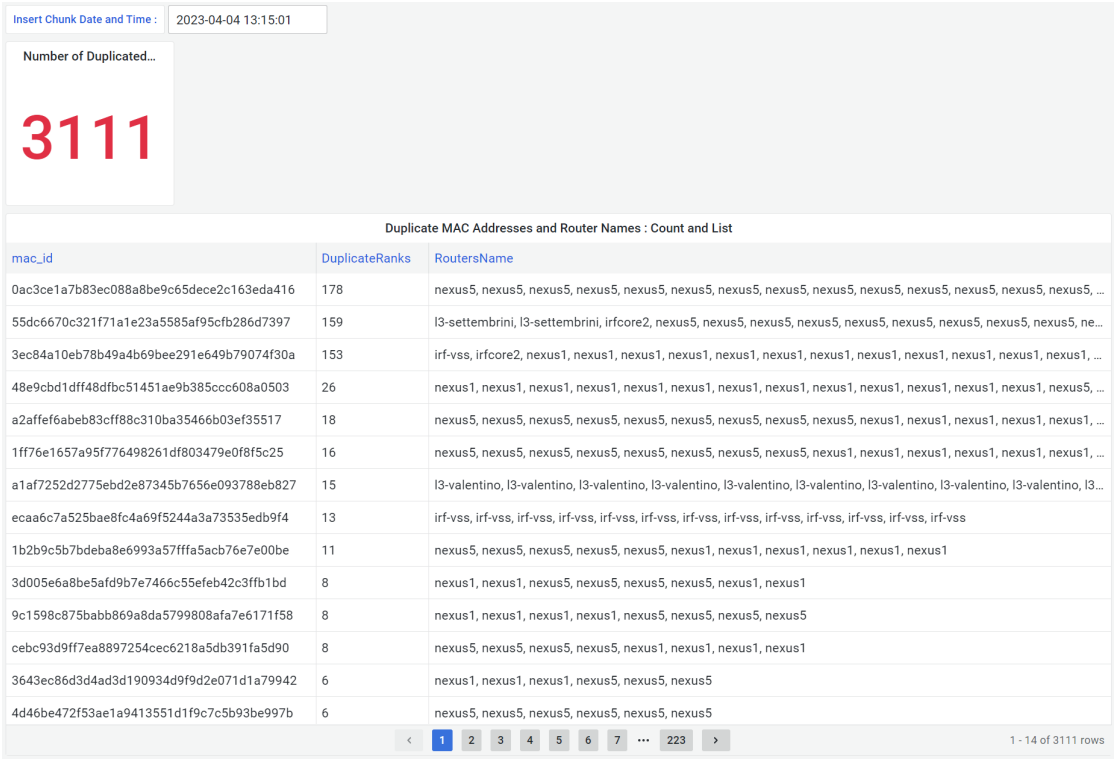

Figure 4.16: Query Dashboard - Duplicated MACs at the same chunk

the dashboard shows a total of 3,295 identical lines for the given date and time. In the next section, we can see the IPs and corresponding MAC addresses, the number of occurrences of this combination, and the names of the routers that these identical lines come from. This dashboard helps to identify these lines for further investigation.

4.2.4 Dashboard 6

Dashboard six provides information about the variation of connected devices in each department over time. This is an interactive dashboard that requires the user to input the name of a department. When the dashboard is opened, it retrieves the list of departments from the database and provides this list in a drop-down menu. As shown in Figure 4.18, the department IT has been selected.
The chart in the dashboard shows the variation of connected devices in the chosen department over time. This helps to monitor the number of devices connected to the network in a specific department and identify any trends or anomalies.

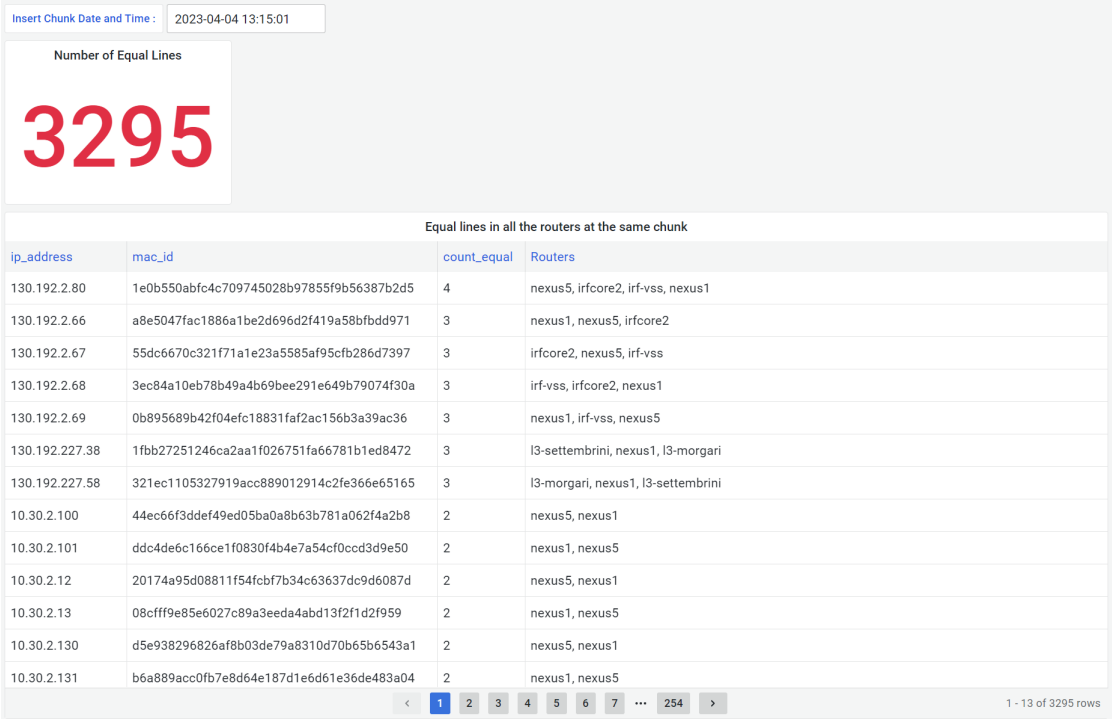

Figure 4.17: Query Dashboard - Equal Lines at the same chunk

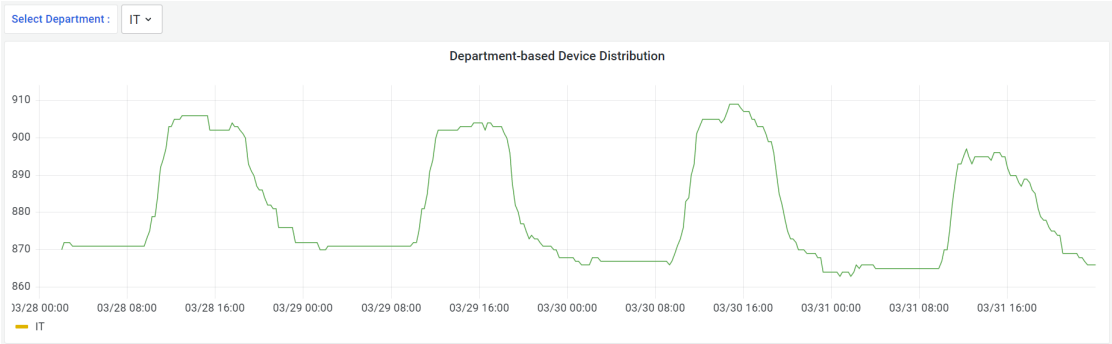

Figure 4.18: Query Dashboard - Department-based Device distribution

4.2.5 Dashboard 7 and 8

Dashboard 7 and 8 aim to control the metadata table for any anomalies. Since the metadata table is updated manually by operators, there may be some outdated information. For instance, an IP address in the metadata table may not have its most recent MAC address. These dashboards report such anomalies. In Figure 4.19, we can see Dashboard 7 which checks each IP address from the ARP table

of the last imported chunk in the metadata table. If the MAC address obtained from the ARP tables of the routers does not match the stored MAC address for that IP in the metadata table, the dashboard reports this anomaly. It displays the checked IP address, the MAC address obtained from the ARP table of the router, the MAC address stored in the metadata table that needs to be updated, and the operator's information who has registered and updated that device in the metadata table. It should be noted that, for privacy reasons, both MAC addresses and user information are hashed using a specific algorithm in this table.

IPs with different MAC_ARP and MAC_DNS

IP_address	MAC_ARP_table	MAC_DNS_table	ACR_Utl	MATR_SA
130.192.160.23	25fe12d4bde30daa645daed08e78b5b0f8e5dcb9	340607584CB28E0E74B569BB13FB42F15144E7A4	DAD	C20742C111C0F4366F64895C6351B67F0B24CDE1
130.192.160.33	67788fcf32aeeea7d71ad3d98f811dddcc841c1e	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.34	5b841fa69d8b5bcb4b4fea8e6ad2fafa4b49ce80	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.35	4fafbce18f8340b9d5420f99ae2eed018626f59a	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.36	4aee6af36ce531ba3181f202bfa58c678c1cb803	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.37	65d10e9c7915bf2db450aa982a1fa9f2e0fe1b77	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.38	fc6f4f6b62bbb962476f7a082338f27f27f510b7	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.39	252e6af6563a6e4119c77278007a753965a2086b	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.40	2da49be6355a86bd0d3d73123c7e4cf6c5a3b317	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80
130.192.160.41	9d61d8c68c60d344b9985c58c2e17c3713c77196	5FF830C6315A6ED9C51B43502CA5AA16D15C63DC	CELM	1063593203B77CDECB860D1A0F35432C7EFB8E80

‹ 1 2 3 4 5 6 7 ··· 508 ›

1 - 10 of 5079 rows

Figure 4.19: Query Dashboard - ARP and Metadata Table data Anomalies detected based on IP

Dashboard 8 provides similar functionality as Dashboard 7, but instead of checking MAC addresses for an IP, it checks the IP address obtained from the routers against the metadata table. The dashboard reports an anomaly if an IP address obtained from the routers in the last imported chunk does not match the IP address stored in the metadata table for a given MAC address.

4.2.6 Dashboard 9 and 10

Dashboard 9 and 10 help the operator to retrieve information related to an IP or MAC address. Dashboard 9 receives an IP address as input and first checks that IP in the metadata table named 'replicated_polito_dns'. If the query finds a row that has that IP address, it reports all the information of that row in both tables. As we can see in the first section of Figure 4.20 for the IP address of 130.192.53.31, it retrieves all the columns in that table for that IP address. Then it checks that IP in the table of ARP table information named 'router_log'. Therefore, we can see all the IP and corresponding MAC addresses that have been imported from the

router ARP table. The last two columns show the beginning and ending of the period that the IP address is checked in that table. Actually, it tries to check that IP in all the existing rows of data.

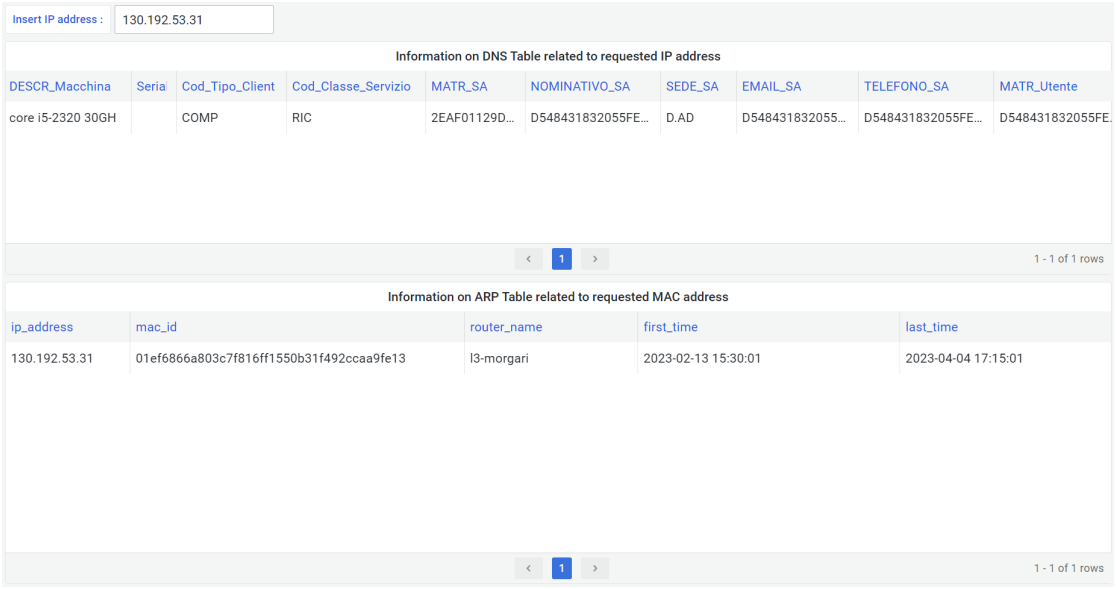

Figure 4.20: Query Dashboard - Search Information in the database based on IP

Dashboard 10 performs a similar search, but it receives a hashed MAC address as input and checks that MAC in both tables.

4.3 Alerts and Notification

The monitoring system implemented in this project is capable of providing a vast amount of information related to different aspects and parameters obtained from the ARP table and metadata information. However, due to the high volume of parameters, the monitoring system needs to have an automated system that can check and report anomalies to the operator in real time.
The alert and notification system has been designed for this monitoring to help the operator identify anomalies and critical issues as soon as possible. By defining different parameters and thresholds, the system can detect when a certain parameter has exceeded its normal range and automatically generate an alert or notification to the operator. This way, the operator can take immediate action and address the issue before it becomes more serious or causes further damage.
For instance, if the number of connected devices in a certain department exceeds a certain threshold, the system will automatically generate an alert and notify the operator via email or text message. Similarly, if there is a mismatch between the

MAC address obtained from the ARP table and the one stored in the metadata table, the system will generate a notification and alert the operator.

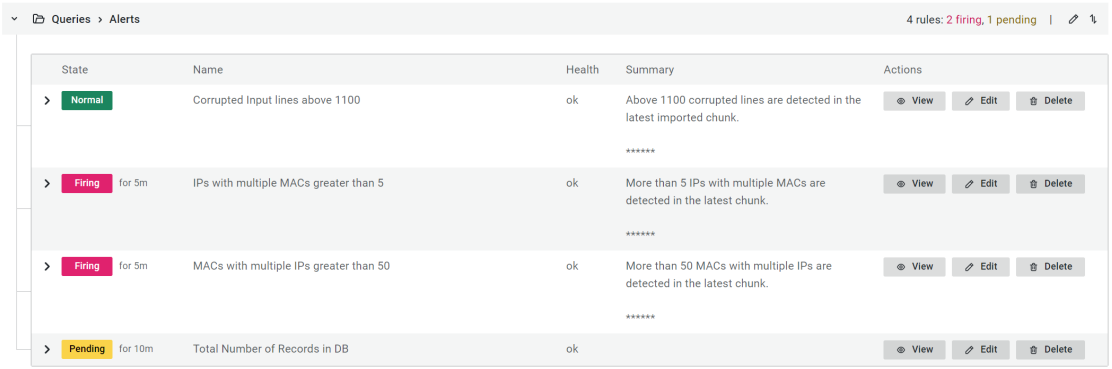

Figure 4.21: Alert and Notification Dashboard

Figure 4.21 displays the alert and notification dashboard created on Grafana for this project. The dashboard presents four examples of configured alerts. The first alert notifies the operator when the number of corrupted lines in the router exceeds 1100 lines. The second and third alerts also notify the operator once the number of IPs with multiple MACs exceeds 5 and when the number of MACs with multiple IPs exceeds 50, respectively, which have been triggered and communicated to the operator. the last one is a simple alert that reports to the operator the total number of rows in the table of the database, so the operator can track the health of the system and analyze the new input data. These examples are only a small sample of what can be configured. For every statistic and parameter monitored in the system, we can configure automatic checking and receive alerts via different media such as notifications, text messages, emails, Telegram, etc.
One of the media types that has been configured to receive alerts is Telegram, for which a dedicated Telegram bot has been created and connected to the monitoring platform. The bot is responsible for receiving alerts and presenting them to the user. It is a member of a group specifically created for the monitoring system, and all alerts are sent to that group. Any operator who is a member of this group can receive the alerts. Figure 4.22 shows this group and an example of an alert received by the Telegram bot. In this figure, we can see that a notification has been triggered for an alert regarding the number of corrupted lines that have exceeded a certain threshold.

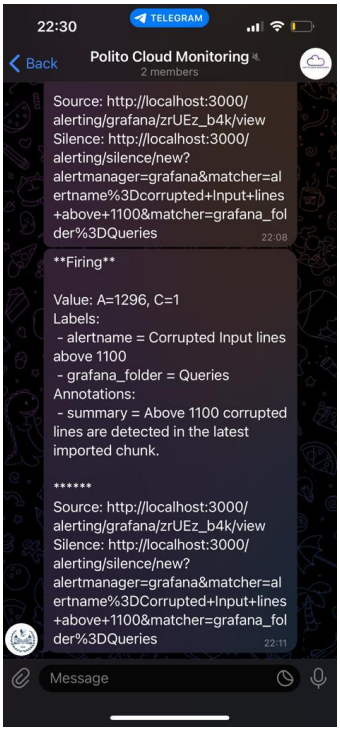

Figure 4.22: Alert and Notification - Telegram Media Type

Chapter 5

Conclusion

The rising usage of cloud computing and the growing dependence on networked infrastructure has made enhancing cloud network security a crucial priority. Monitoring plays a vital role in network security, and to tackle this issue, this thesis proposes a tailored monitoring system that concentrates on analyzing IP and MAC addresses using the ARP table data extracted from the network's routers. The monitoring system we have implemented in this project provides real-time information about the health of the network infrastructure. By collecting and analyzing data from routers, we can identify potential issues and take proactive measures to prevent downtime or other disruptions.

The system is constructed using a combination of open-source tools and custom scripts, enabling it to monitor a wide range of network-related metrics and parameters. All statistics are derived by analyzing each line of the ARP tables of the routers in the network, and then combining this information with metadata from a separate database that is regularly updated by network operators. This metadata table includes details such as the department where the device is used, the department where the user of the device works, the device type, the operating system, and more. Once all the required input data is stored in the database tables, the system can derive various statistics from the data.

The statistics generated by the system include information on network-connected devices, the real-time number of connected devices, the type and class of devices by number or percentage, the distribution of devices across different departments of the university, the number and list of identical lines or duplicated IP or MAC addresses in all the routers. All the statistics are presented in various dashboards in different graphs and charts using the Grafana platform which is one of the powerful visualization tools. By creating custom dashboards and visualizations, we can gain insights into the performance of the network and identify patterns or trends that might otherwise go unnoticed.

Network administrators can benefit greatly from the statistics on different dashboards by analyzing the trends and fluctuations in each statistic in any period, whether short or long-term. With the help of these dashboards, administrators can easily monitor the number of devices connected to the network, which is valuable information for assessing the network's capacity and preventing overload. Additionally, by keeping track of changes in the number of connected devices, administrators can quickly identify any significant increases or decreases that may indicate a security or disconnection issue. This platform allows administrators to check the real-time distribution of devices across different departments. This feature helps identify any issues with network traffic distribution and enables corrective actions to optimize the network. By ensuring efficient network traffic distribution, administrators can prevent potential bottlenecks and maintain optimal network performance and reliability.
This monitoring system can help network administrators detect and respond to security threats in real time. By using the system to check for duplicated IP and MAC addresses on the routers, administrators can quickly identify any malicious activity and threats on the network. Monitoring fluctuations in the number of duplications and incomplete ARP associations can also be crucial in identifying potential network attacks. In such cases, administrators can take immediate action to prevent further damage and ensure the security and integrity of the network. Another helpful feature is the ability to categorize the connected devices by operating system type. This feature enables network operators to better understand the distribution of operating systems and identify outdated or unsupported systems in use. In the case of specific vulnerabilities affecting a particular OS, operators can quickly determine how many devices in the network are using that OS and take steps to mitigate any flaws.
In addition to the technical aspects of the project, we have implemented a process for managing alerts and notifications. All alerts related to statistical changes or significant fluctuations in metrics are sent via various media, such as email or Telegram. This allows for real-time monitoring anytime, anywhere, providing a more proactive approach to network management.

One of the key benefits of this monitoring system is its flexibility. We can easily configure it to monitor different types and models of devices that have ARP tables inside. This makes it suitable for a wide range of environments, from small businesses to large enterprises. We can also customize the system to meet specific requirements or integrate it with other tools or systems.

Overall, we believe that this monitoring system has the potential to make a significant impact on the reliability and performance of network infrastructure. By providing real-time information and proactive alerts, we can help ensure that

networks are running smoothly and efficiently. We also believe that this project serves as a proof-of-concept for the use of open-source tools and custom scripts in monitoring and managing network infrastructure. We hope that this report will inspire others to explore this approach and contribute to the ongoing development of these tools and techniques.

Bibliography

[1] Giuseppe Aceto, Alessio Botta, Walter de Donato, and Antonio Pescapè. «Cloud monitoring: A survey». In: *Computer Networks* 57.9 (2013), pp. 2093–2115. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2013.04.001. URL: https://www.sciencedirect.com/science/article/pii/S1389128613001084 (cit. on pp. 1, 3).

[2] Dinkar Sitaram and Geetha Manjunath. «Chapter 8 - Managing the Cloud». In: *Moving To The Cloud*. Ed. by Dinkar Sitaram and Geetha Manjunath. Boston: Syngress, 2012, pp. 329–349. ISBN: 978-1-59749-725-1. DOI: https://doi.org/10.1016/B978-1-59749-725-1.00008-1. URL: https://www.sciencedirect.com/science/article/pii/B9781597497251000081 (cit. on p. 1).

[3] Dinkar Sitaram and Geetha Manjunath. «Chapter 2 - Infrastructure as a Service». In: *Moving To The Cloud*. Ed. by Dinkar Sitaram and Geetha Manjunath. Boston: Syngress, 2012, pp. 23–71. ISBN: 978-1-59749-725-1. DOI: https://doi.org/10.1016/B978-1-59749-725-1.00002-0. URL: https://www.sciencedirect.com/science/article/pii/B9781597497251000020 (cit. on p. 2).

[4] Dinkar Sitaram and Geetha Manjunath. «Chapter 3 - Platform as a Service». In: *Moving To The Cloud*. Ed. by Dinkar Sitaram and Geetha Manjunath. Boston: Syngress, 2012, pp. 73–152. ISBN: 978-1-59749-725-1. DOI: https://doi.org/10.1016/B978-1-59749-725-1.00003-2. URL: https://www.sciencedirect.com/science/article/pii/B9781597497251000032 (cit. on p. 2).

[5] Dinkar Sitaram and Geetha Manjunath. «Chapter 4 - Software as a Service». In: *Moving To The Cloud*. Ed. by Dinkar Sitaram and Geetha Manjunath. Boston: Syngress, 2012, pp. 153–204. ISBN: 978-1-59749-725-1. DOI: https://doi.org/10.1016/B978-1-59749-725-1.00004-4. URL: https://www.sciencedirect.com/science/article/pii/B9781597497251000044 (cit. on p. 2).

[6] Mahantesh N. Birje and Chetan M. Bulla. «Cloud Monitoring System : Basics , Phases and Challenges 4743». In: 2019 (cit. on p. 3).

[7] Ahmed Fahad, Abdulghani Ahmed, and Mohd Nizam Mohmad Kahar. «The importance of monitoring cloud computing: An intensive review». In: Nov. 2017, pp. 2858–2863. DOI: 10.1109/TENCON.2017.8228349 (cit. on p. 4).

[8] Tom Laszewski and Prakash Nauduri. «Chapter 1 - Migrating to the Cloud: Client/Server Migrations to the Oracle Cloud». In: *Migrating to the Cloud*. Ed. by Tom Laszewski and Prakash Nauduri. Boston: Syngress, 2012, pp. 1–19. ISBN: 978-1-59749-647-6. DOI: https://doi.org/10.1016/B978-1-59749-647-6.00001-6. URL: https://www.sciencedirect.com/science/article/pii/B9781597496476000016 (cit. on p. 4).

[9] Cem Gurkok. «Chapter 63 - Securing Cloud Computing Systems». In: *Computer and Information Security Handbook (Third Edition)*. Ed. by John R. Vacca. Third Edition. Boston: Morgan Kaufmann, 2017, pp. 897–922. ISBN: 978-0-12-803843-7. DOI: https://doi.org/10.1016/B978-0-12-803843-7.00063-6. URL: https://www.sciencedirect.com/science/article/pii/B9780128038437000636 (cit. on p. 4).

[10] Cyprain Onyia, Kelvin Nnamani, Ekene Alagbu, and Christopher Ezeagwu. «Comparative Analysis of OSI and TCP/IP Models in Network Communication». In: 7 (July 2021), pp. 08–14. DOI: 10.35629/9795-07060814 (cit. on p. 10).