

HARDWARE AND SOFTWARE INVENTORY BEST PRACTICES APPLIED TO GOVERNMENT COMPUTER NETWORK AND SYSTEMS

Welber Santos de Oliveira¹, Felipe Barreto de Oliveira¹, Fábio Lúcio Lopes de Mendonça¹, Luiz Augusto dos Santos Pires¹, Renato José da Silva Camões², Robson de Oliveira Alburquerque¹ and Georges Daniel Amvame Nze¹

¹Electrical Engineering Department, University of Brasilia - Brasilia, Federal District 70910-900, Brazil

²LATITUDE/UNB decision-making technology laboratory - University of Brasilia - Brasilia, Federal District 70910-900, Brazil

ABSTRACT

Security of computer environments is one of the main topics in the last decade. Adopting best practices is also important because it provides parameters to evaluate systems and networks in terms of use and security related to hardware and software inventory. Considering such aspects, this paper presents management best practices for information systems and computer networks related to hardware and software inventory using open-source solutions, pointing at security and asset monitoring applied to the environment of the Brazilian General Attorney of the National Treasury (PGFN). The adopted methodology presents strategies for good results in evaluating asset monitoring and management processes related to the inventory of hardware and software used in the PGFN computer network.

KEYWORDS

Hardware Inventory, Software Inventory, Best Practices, Network Security

1. INTRODUCTION

This paper considers the adoption of best practices allied to open-source software to provide solutions for hardware and software inventory of computer networks used in the environment of the Brazilian General Attorney of the National Treasury (PGFN) to help computer management and computer security controls to reduce security risks.

Companies of different sizes are aware of the risks and give significant focus to the efforts of their information technology (IT) and security teams to prevent attacks on their computer systems by malicious actors. To accomplish this, companies establish robust safeguards that include individuals, processes, and technology to defend against cyber-security threats. Even though these defenses, at some point, are good, providing some security controls against threats, it might be surprising to acknowledge that they are sometimes being built on quicksand (Smyth, 2015). Also, the first two controls from the SANS Critical Security Controls (SANS, 2021) emphasize the need to inventory, track, and rectify hardware devices to allow access only for those authorized. This is important because attackers seek to exploit unprotected systems attached to the network, such as outdated software versions.

According to (Hughes, 2022), over 75% of organizations have increased their use of open-source tools, recognizing their growing dependence, while the 2020 crisis led 68% of companies to adopt open-source solutions to cut costs (Germain, 2021). (Ruffin & Ebert, 2004) highlights the legal aspects of open-source software, allowing use, modification, and distribution under certain conditions. Proprietary products contrast in terms of flexibility and costs. Embracing open source implies adjusting support and implementing best practices, even without being the owner.

This paper presents the methodology supported by management best practices for information systems and computer networks related to hardware and software inventory using open-source software. The main idea is to reduce risks while providing good security to hardware and software assets' visibility and monitoring in the

PGFN. The adopted methodology shows that adopting best practices allied to open-source software brings satisfactory results in evaluating asset monitoring and management processes related to the inventory of hardware and software used in the computer network of PGFN.

This paper is organized as follows. After this introduction, section 2 provides some related work considering the adoption of best practices, followed by section 3, where the methodology adopted in this work is presented and explained. Chapter 4 provides a case study and the corresponding results. Section 5 concludes this paper and provides some ideas for future works.

2. RELATED WORKS

Several studies have proposed the implementation of best practices in software inventory usage. Some of these will be mentioned in this section.

In work by Muyumba & Phiri (2017) was presented a model for efficient management of spare parts inventory for the Zambian Air Force (ZAF) equipment was proposed. The aim was to develop an architecture capable of preventing human errors (such as inventory duplication) that could arise from manual controls. To achieve this, a cloud-based architecture was developed, leveraging barcode technology. The outcome was an automated inventory control system that significantly improved speed and reduced susceptibility to errors.

The study conducted by Silva & Pinto (2019) aims to propose a review of asset management in a microenterprise in Bragança Paulista. To achieve this, defined steps were implemented, encompassing the generation of business value from the solution. They installed OCS Inventory NG and performed a Network Inventory, deploying software agents on each asset in the park. As a practical outcome, the authors obtained reports with an environmental analysis that facilitates decision-making.

In the work presented by Labanda-Jaramillo et al. (2019), research was conducted to enhance the IT management processes at the National University of Loja. For this purpose, the study was divided into several stages, one focusing on administering network asset configurations. As a result, using the OCS Inventory proved efficient in identifying the characteristics of the university's asset configurations, thereby assisting in identifying obsolete systems or hardware.

In the study proposed by Shackleton (2017), one of the objectives was to assess the applicability of the OCS Inventory in the NIST 800-53 framework (Joint Task Force Transformation Initiative, 2013). To achieve this, a Proof of Concept (PoC) was developed, configuring Windows 10 servers, a collection of servers from 2008 R2 to 2012 R2, and 5 Ubuntu 16.04 LTS servers. As a result, the author concluded that the following controls are implemented in the OCS Inventory: ID CM-8 (1), ID CM-8 (2), and ID CM-8 (7). The controls ID CM-8 and ID CM-8 (4) were partially covered. However, the remaining controls (ID CM-8 (3), ID CM-8 (3)(a), ID CM-8 (6), and ID CM-8 (8)) were not implemented.

3. METHODOLOGY AND PROCESSES

The methodology elaborated for the PGFN includes the description of the methods and processes developed during the research, establishing techniques that were used to elaborate the solution for knowledge management and security controls.

3.1 Methods Employed

To execute the proposed objectives, 10 phases were elaborated for implementing the software and hardware inventory management system in the PGFN. Figure 1 indicates the phases considered, and table 1 resumes each phase. These practices are related to the key points of best practices expressed in the work of Olson & Weins (2009).



Figure 1. Methodology phases applied to hardware and software inventory for the PGFN

Table 1. Phases resume

Phase	Description
Initial Meetings	To meet the needs of the PGFN, meetings were conducted with managers and technicians to understand the organization's demands. Technical issues were identified in PGFN's environment to enhance management and motivate the specific needs of the management team. Reference to key point "Executive Sponsorship" presented by Olson & Weins (2009).
Addressing Security Controls	Exploring the issues identified in the initial meetings, as well as the concerns raised by the PGFN team itself, the need to prioritize efforts toward implementing hardware and software inventory management was emphasized. As it discusses the priorities and how the organization should handle inventory software, this item is related to the key point of "Open-Source Policy" in the work of Olson & Weins (2009).
Current Environment Analysis	To define the tool that best fits the scenario found, it was also necessary to have a greater notion not only of the existing problems but also of the plans and objectives with inventory management, in addition to understanding the size of the computational park and its organization in the different units of the PGFN.
Documentation review	Practice is necessary for understanding the environment, customizing the tool, and adapting to the established strategy and previous steps. Important to identify sources of research, communities, support, and websites. It can be referred to as the key point of "Provisioning" in the work of Olson & Weins (2009).
Choosing an Inventory Management Software	An analysis and comparison of various open-source software, including GLPi, OCS Inventory, Zabbix Inventory, and CITSmart, were conducted. The tools were presented to the management team, highlighting their strengths and weaknesses. Additionally, the computational requirements were assessed. Based on this analysis, the adoption and implementation of OCS Inventory was proposed as the suitable solution to meet the identified demands. As this item is analyzing the conditions of the software and choosing the best solution based on the software's capabilities that attends best to the organization's policy, it is related to the key point "Open-Source Policy" in the work of Olson & Weins (2009).
Tool Adaptation	Regardless of the extent of tool coverage, each topology and technological park present variations between companies, which justifies the need for customization in using these tools. For this, a customized script was created, ensuring the full operation of the tool. This practice referred to the key point, "Open-Source Policy," in the work of Olson & Weins (2009).
Documentation Guide	An OCS Inventory installation script was established, including the script created. Reference can be made to the key point "Requests and Approvals" in the work of Olson & Weins (2009)
Simulating Tool Performance	An implementation and simulation within a controlled environment were executed, employing a Virtual Machine to execute the installation and the tool's efficacy was examined. This action can be related to the key point "Operationalizing Open-Source Policies," as highlighted in the work of Olson & Weins (2009).
Tests and demonstrations	Practice responsible for the tests, demonstrations, and audit analysis of the proposed environment, verifying its adequate functioning. This step is like the key point, "Auditing," presented in the work of Olson & Weins (2009).
Reports	Practice is presented to generate reports on the steps performed as well as the results obtained during the process, this practice being like the key point "Reporting" described by Olson & Weins (2009).

3.2 Processes Used

To assist the implementation of the new tool, a process model was designed according to the Business Process Management Notation (BPMN) (Figure 2), which expresses the actions performed by each of the actors present in the process, thus presenting an equal and general view of the best practices applied.

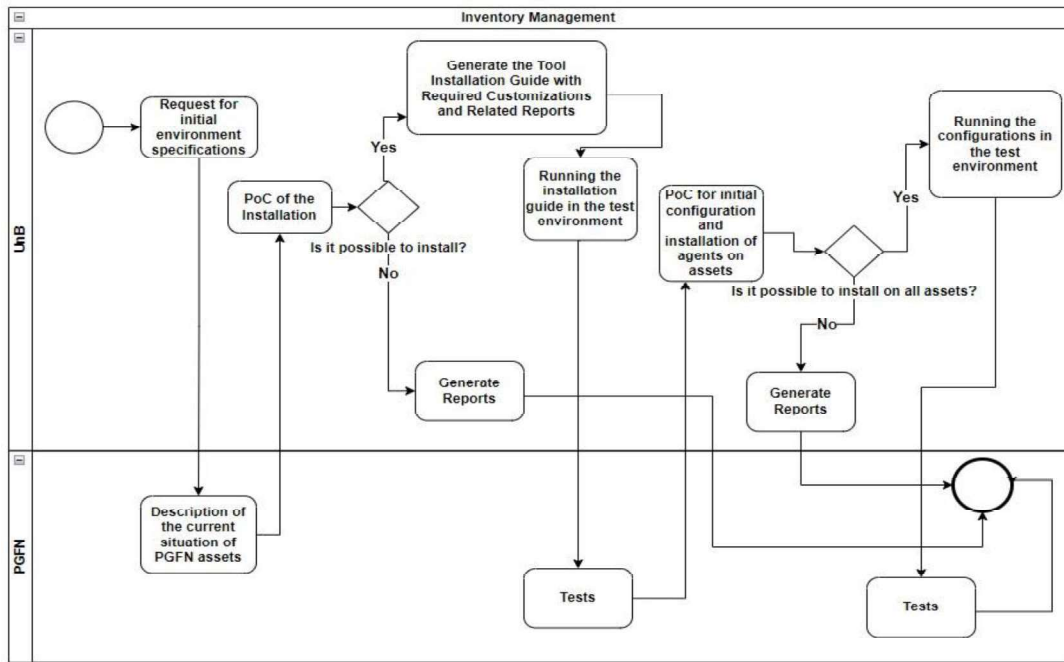


Figure 2. BPMN Model

To execute this process, a case study was created, which initially required a detailed description of the initial scenario of the structure. After the survey, it was necessary to formulate and create test scenarios, where several forms of updating were tested to build and update the installation guide. Finally, the improvements achieved were presented.

4. CASE STUDY AND RESULTS

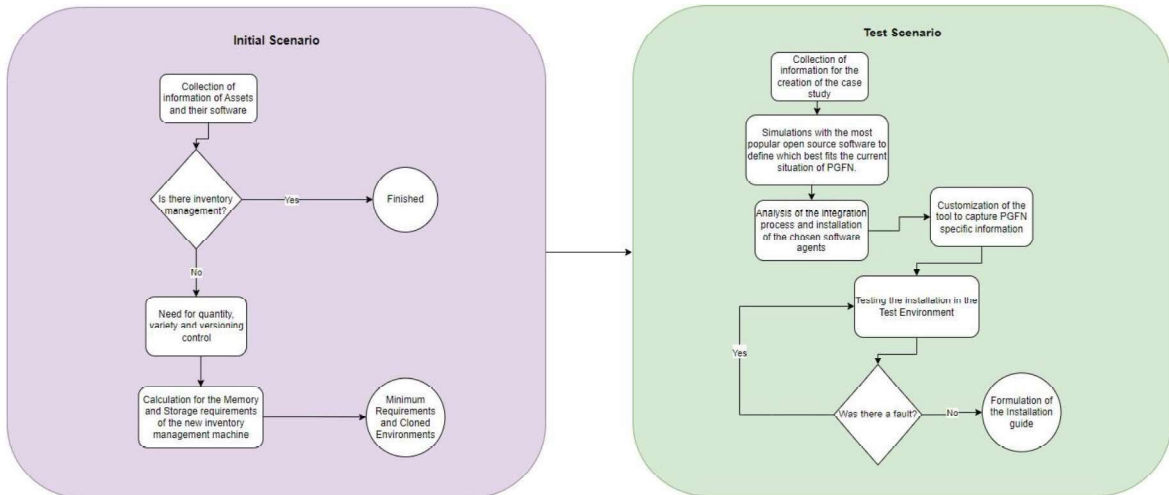


Figure 3. Flows used to structure and apply defined best practices

As a guide for the case study, 2 flowcharts were described (Figure 3), showing the steps followed in each scenario where the first scenario describes how to verify the actual situation in the company, using special strategies to know the specifications. The second scenario explains how to adapt the open-source solutions for the different proposed specifications. Each scenario will be presented with more details in the following.

4.1 Initial Scenarios and Test Considerations

The PGFN has extensive hardware and software assets and needs a tool to manage this inventory as optimally as possible. First, it was necessary to understand what the inventory being requested covered, to align the expectations of the parties, and to investigate the history of inventory management in the agency. To this end, meetings were held where real cases were presented and the main problems to be solved were detailed, to determine the mandatory requirements of the software to be implemented.

The decision was reached that open-source management software would be the most viable option for solving this demand, as it would be free of financial costs. Based on the research, the research team determined the reasons for keeping track of inventory and defining software and hardware inventory. Having a discovery and inventory system that can track assets by hardware, operating system, applications, and versions is important when it comes to keeping track of computers and which users are using them, thus discovering unknown and unauthorized devices on the network and determining the heterogeneity of the computer park and thus making decisions about when to update or not update any network assets. The hardware inventory includes everything connected to the network, with people, or that generates information. This class includes devices such as laptops, cell phones, tablets, desktops, printers, servers, routers, etc. Software inventory is a specialist program that detects operating systems, licenses, versions, manufacturers, developers, drivers, complementary devices, libraries, registry keys, and APIs. This inventory involves collecting information on all the software used in the company, cross-referencing the information, and then making decisions.

After comparing open-source software inventory management solutions and presenting the pros and cons to the PGFN team, one specific issue became relevant. In addition to the comparative criteria established, a list of requirements was given that the inventory solution must manage to satisfy the needs of PGFN's inventory management model, these being: Machine name, IP and MAC address; Vendor; Model; CPF (Brazilian Person ID Number); Status; Software (Name and version); Last capture (Last login register); Group (Region); Location.

Several options that would meet the needs mapped at PGFN were analyzed to determine the best choice among those studied: OCS Inventory, Zabbix Inventory, CITSmart, and GLPi. The first three were presented to the PGFN; a description is attached to this report. After further synchronous virtual meetings, expectations were aligned, and the steps to be taken by the team were determined.

Therefore, in addition to presenting the chosen inventory software in a comparative manner, it was necessary to demonstrate each by collecting all the data presented in the previous section.

4.2 Comparison of Best Solutions

OCS Inventory was recommended following a comparative study carried out by the research infrastructure team, which was then presented to PGNF. The study evaluated five key criteria: documentation, updates, community, usability, functionality, and robustness, as shown in Table 2.

Table 2. Criteria Evaluation

Phase	Description
Documentation	both tools offer comprehensive resources, with GLPI exhibiting a finer granularity by categorizing its documentation into distinct levels: administrator, user, and developer.
Updates	GLPI's updates are more recent than those of OCS. Nevertheless, this disparity doesn't wield substantial significance in the comparison, as both tools adhere to an identical server update schedule. The notable divergence is primarily evident in the version intervals concerning agent updates.
Community	While GLPI boasts a larger community of contributors, OCS maintains a more vibrant presence on the GLPI subreddit. It's essential to underscore that both tools have thriving communities, and various deployment solutions are accessible for each.
Usability	When considering usability, it's important to mention that both tools support the Portuguese language. However, OCS distinguishes itself with a simpler and more intuitive interface, in contrast to GLPI's interface, which offers many configuration options that can occasionally lead to complexities in navigation.
Functionality	GLPI not only maintains an asset inventory but also offers features like ticket management, problem tracking, resource reservation, maintenance planning, and much more. It's used to facilitate efficient management of IT assets and improve process organization. An interface is utilized to manage the

Phase	Description
	<p>inventory of computers, peripherals, printers, and related components, employing inventory tools like Fusion Inventory or OCS Inventory; Managing warranties and financial data, including purchase orders, warranties, and extensions; Data Center Infrastructure Management (DCIM); Knowledge base and Frequently Asked Questions (FAQ); Managing the lifecycle of items and overseeing contracts, contacts, and documentation pertaining to inventory items; Incident, request, problem, and change management; Supports numerous plugins that provide additional features.</p> <p>OCS Inventory performs automatic device discovery on the network, gathers information, and maintains an up-to-date inventory. It is designed to provide accurate and real-time information about hardware and software assets within an organization. Relevant inventory information; Network discovery; Cutting-edge broadcast system for deploying software, executing scripts, and issuing commands on computers, all while avoiding network overload; SOAP-accessible Web service; Intuitive web interface for ease of use; Plugin support through APIs; Support for multiple operating systems, including Microsoft Windows, Linux, BSD, Sun Solaris, IBM AIX, HP-UX, MacOS X, Android; N-tier architecture utilizing common standards, HTTP/HTTPS protocols, and XML data formatting</p>
Robustness	<p>The device attributes that OCS can capture are the operating system version, installed software and their versions, equipment serial number, hardware component model and features, and more. The agent's IP Discover functionality enables the discovery of all computers and devices on the network, streamlining inventory item registration.</p> <p>The sole requirement to receive information from a computer is that the OCS Inventory client program, the agent, is installed. Once installed, the agent itself transmits the data to the server. For inventorying data from a device with an embedded system, where agent installation isn't feasible, collection must occur via SNMP protocol. Moreover, even smartphones with Android can be inventoried through the OCS Inventory app.</p>

4.3 Tools Adaptations

In general, it is common for inventory solutions to record practically all this information, but two of them became a temporary obstacle in the execution of the project. The Group and Locality fields do not exactly depend on information present on the machine, as they are related to external factors and, therefore, cannot be automatically cataloged by the inventory software. A machine's Locality is determined by its IP address, where each locality corresponds to a different IP range determined by PGFN. The Group (Region) obeys a domain restriction of 5 items, and its value for each device obeys a logic involving its Locality.

The primary solution presented, which allowed this information to be mapped, was to use the TAG field (a free field for entering any information), which can be filled in during the configuration of the inventory solution agent so that it would be possible to add the equipment's Locality so that it could be used as a query parameter during the generation of reports and inventory management.

However, the need for this manual work to fill in the Locality and Group fields made the proposed alternative unfeasible, so the challenge became how to register the Locality and Group in an automated way.

Studying the criteria that establish the Locality field, we saw that the IP range that characterizes each locality could be used as a search parameter in the inventory database, allowing us to locate all the equipment that belongs to a specific IP range. However, with this query, it would still not be possible to separate the elements if the range is not defined by an entire octet. To solve this problem, the possibility of using Regular Expressions in the query was explored, as this feature will allow for a much more specific query.

After a careful analysis, PGFN asked for some clarifications on the organization of some data, including two pieces of information that depended on factors external to the information stored in the asset: the group, which indicates the region in the country where an item is located, and the location, which is determined by a predetermined IP mask that represents the PGFN unit where the item is located.

4.4 Comparison and Discussions

Table 3 compares the results of the studies related to the proposed work. The following sections describe the importance of the results obtained.

Table 3. Comparison between related work and the proposal presented

Heading level	Definition of Best Practices	Process Model and Flowcharts	Generalization	Improving control and monitoring
(Ruffin & Ebert, 2004)	-	-	X	-
(Muyumba & Phiri, 2017)	-	-	-	-
(Olson & Weins, 2009)	-	-	X	-
(Arsan et al., 2013)	X	-	X	X
This research	X	X	X	-

4.4.1 Discussions

Regarding improving software optimization and updates, the license updates for OCS involve upgrading the system. They can be associated with improvements in functionality and increased security, among other aspects. Keeping the OCS (Open Computers and Software Inventory) software updated is crucial for various reasons, but two stand out. A) Security: Frequent updates include security patches that help protect the system against known vulnerabilities. By keeping the OCS updated, PGFN reduces the risk of cyber-attacks and potential invasions. B) Compatibility: As new technologies and operating systems are released, OCS updates can ensure compatibility with these new platforms, ensuring the software functions seamlessly in updated environments. Updating OCS is essential to ensure security, stability, performance, compatibility, access to new features, and adequate technical support.

Regarding vulnerability prevention, security tests, and assessments should be conducted to identify potential weaknesses in the system and address them before attackers exploit them. Within this scope, it's crucial to implement robust authentication methods such as strong passwords, two-factor authentication, or integration with secure authentication systems to prevent unauthorized access. Another critical factor is ensuring that data transmitted and stored by the OCS is encrypted, preventing unauthorized third-party access to confidential information. Lastly, stringent access control to the system's functionalities is important to prevent vulnerabilities, allowing only authorized users access to specific areas and resources.

Regarding asset control and monitoring, as per the OCS implementation documentation mentioned in (Olson & Weins, 2009), it's possible to enhance control by focusing on the following aspects. A) Data Centralization: Enhancing OCS's capability to gather and store precise asset information, ensuring all relevant data is centralized and easily accessible. B) Process Automation: Implementing automation for asset data collection and updates, reducing manual errors, and ensuring real-time updated records. C) Management Policies: Establishing clear asset management policies to standardize procedures, assign responsibilities, and ensure compliance with regulations. D) Proactive Monitoring: Integrating continuous monitoring systems to swiftly identify and respond to asset changes, such as software updates, hardware modifications, or device movements within the network. E) Reporting and Analysis: Enhancing reporting and analysis capabilities to provide valuable insights into the asset status, aiding in strategic decision-making. F) Integration with Other Tools: Integrating with other management tools, such as help desk systems or security solutions, creates a comprehensive view of assets and enables more efficient actions.

Regarding aggregating the institution's management process, the OCS evidently contributes to PGNF's hardware and software asset management and control, maintaining an updated and detailed inventory. Additionally, it's instrumental in providing information about installed software versions and status, streamlining the planning of future updates and maintenance within PGNF. In most organizations, the OCS not only aids in compliance and audits but also significantly streamlines technical support by enabling swift identification of system issues and providing crucial maintenance-related information.

5. CONCLUSION

PGFN increased its visibility to assets with the proposed methods, processes, and best practices. Although there is no significant advantage, one must consider the limitations of capabilities and personnel that affect government institutions.

Considering all the important steps it may take, it is important to mention the absence of a correct management process regarding hardware and inventory. Without hardware and software inventory, the organization is blind to the number of risks with outdated hardware and software in their computer network. Only this is enough security risk for the management team to review its practices and solutions.

In this sense, this research proposed a corresponding methodology followed by processes and the support of best practices to increase environmental security and apply better security controls, thus reducing the attack surface and increasing visibility and monitoring of the computer network. It applied a 10-phase methodology with the main objective to increase the quality of the security controls. The results show how best practices allied to good open-source management solutions increase the security of the environment in terms of visibility, monitoring, and helping decision-making processes supported by security risk facts in the environment when considering hardware and software inventory solutions.

As future work, it is the intention of this research to evaluate new scenarios as the needs of PFGN evolve; besides creating a robust plan for decision-making regarding the security risks of dealing with outdated hardware and software in a government environment, indicating what main advantages, limitations, and possible applications it may be considered.

ACKNOWLEDGEMENTS

This work is supported by the Office of the Attorney General of the National Treasury (No. PGFN 23106.148934/2019-67). It is also partially funded by CNPq – National Council for Scientific and Technological Development (PQ-2 312180/2019-5 in Cybersecurity and 465741/2014-2), in part by the National Department of Audit of the Unified Health System (SUS) - DENASUS (23106.118410/2020-85), in part by the Ministry of Economy of Brazil (DIPLA 005/2016 and ENAP 083/2016), partly by the Administrative Council for Economic Defense (CADE 08700.000047/2019-14), partly by the Office of the Solicitor General (AGU 697.935/2019), and partly by the Foundation for Research Support of the Federal District – FAPD.

REFERENCES

- Arsan, T., Başkan, E., Ar, E. and Bozkuş, Z. (2013). A software architecture for inventory management system. *In Innovations and Advances in Computer, Information, Systems Sciences, and Engineering* (pp. 15-27). Springer New York.
- Da Silva, S. F. and Pinto, J. D. S. (2019). Análise da importância da gestão de ativos de TI no ambiente de micro e pequenas empresas. *Revista Científica e-Locução*, 1(15), pp. 18-18.
- Germain, J. M. (2021). The rise of open source: Pandemic, economy, efficiency, trust. Available at: <https://www.linuxinsider.com/story/the-rise-of-open-source-pandemic-economy-efficiency-trust-87057.html> (Accessed 28 December 2023).
- Hughes, O. (2022). Open source is more important than ever, say developers. here's what's driving adoption. Available at: <https://www.zdnet.com/article/open-source-is-more-important-than-ever-say-developers-heres-why/> (Accessed 28 December 2023).
- Joint Task Force Transformation Initiative. (2013). NIST SP 800-53 Rev. 4. Available at: <https://csrc.nist.gov/pubs/sp/800/53/r4/upd3/final> (Accessed 28 December 2023).
- Labanda-Jaramillo, M., Chamba-Eras, L., Coronel-Romero, E., Granda, J. L., Quezada-Sarmiento, P.A. and Roman-Sanchez, M. (2019). June. Proposal for the Reengineering of Processes in the Management of Information and Communication Technology. *In 2019 14th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-5). IEEE.
- Muyumba, T. and Phiri, J. (2017). A Web based Inventory Control System using Cloud Architecture and Barcode Technology for Zambia Air Force. *International Journal of Advanced Computer Science and Applications*, 8(11).
- Olson, G. and Weins, K. (2009). Ten key elements of open source governance in the enterprise. Available at: <https://pt.slideshare.net/RogueWaveSoftware/openlogic-ten-elements-of-open-source-governance> (Accessed 28 December 2023).
- Ruffin, C. and Ebert, C. (2004). Using open source software in product development: A primer. *IEEE software*, 21(1), pp. 82-86.
- SANS. (2021). CIS Critical Security Controls Version 8. Available at: <https://www.cisecurity.org/controls/v8> (Accessed 08 January 2024).
- Shackleton, B. M. (2017). Towards Collection of Cost-Effective Technologies in Support of the NIST Cybersecurity Framework.
- Smyth, V. (2015). Cyber-security fortresses built on quicksand. *Network Security*, 2015(8), pp. 5-8.