NuAppFirewall: An Open-Source macOS Application Firewall for Corporate Security

Bianca Pachêco[1], Carolina Suyungütmüs[1], José Truta[1], Vinicius Delgado[1], Walber Araújo[1], João Brunet[1], Manoel Domingues Junior[2], Angela Melo[2]

[1]Universidade Federal de Campina Grande, Paraíba, Brasil

[2]Nubank, São Paulo, Brasil

{bianca.pacheco, carolina.freitas, jose.bomfim.truta.neto, vinicius.ataide.delgado, walber.wesley.felix.araujo.filho}@ccc.ufcg.edu.br, joao.arthur@computacao.ufcg.edu.br, {manoel.junior, angela.melo}@nubank.com

Abstract. *Managing network access is essential to ensure the security of both users and corporate ecosystems. On macOS, this control is implemented through Content Filters in firewall applications. However, the state of the practice consists mainly of proprietary consumer tools, while open-source alternatives lack the modularity needed for enterprise adoption. This paper introduces NuAppFirewall, an open-source application firewall developed and deployed in production at Nubank. It includes more than 250,000 automatically generated rules for macOS applications, derived from Nubank's validated accesses, providing a security foundation that other organizations can leverage for their own implementations.*

1. Introduction

The security and integrity of data stored on devices and remote workstations are of great importance in today's corporate landscape. Among the various techniques and solutions available to ensure security, firewall applications play a critical role. These tools enable the management of network access performed by applications on a device through rules that verify endpoints and destination ports, determining whether an application is authorized to establish a given connection. On the macOS operating system, such content filters are implemented using the Network Extension framework [Apple 2025a] developed by Apple, which facilitates the creation of network extensions to monitor, filter, and modify data traffic on Mac OS devices.

Although firewall solutions based on Network Extension already exist, such as Little Snitch [Objective Development 2025] and LuLu [Objective-See 2025], most are designed for personal use, where users themselves define access rules and configure the tool. In an enterprise setting, however, centralized management of permissions and access restrictions is crucial to reducing dependence on Virtual Private Network (VPN) connections and ensuring the security of remote workstations. Ideally, this control should be handled by the organization's security team, similar to how a traditional firewall operates on a router, preventing the delegation of critical decisions to users and minimizing risks of non-compliance with corporate policies.

Moreover, adapting existing solutions to this scenario has proven itself challenging. Most of them are proprietary, while open-source alternatives feature tightly coupled code with low modularity and strong interdependencies, making their adaptation to enterprise environments costly and, in many cases, unfeasible. Additionally, the standard rule sets that come with these applications typically cover only the most basic system accesses, resulting in a long process of creating a comprehensive and robust ruleset to enhance device security.

Given this scenario, this paper introduces NuAppFirewall, an open-source tool developed and deployed at Nubank [Nubank 2024] to meet the need for a distributed firewall in corporate settings. The solution addresses three main challenges: rule specification, efficiency and coverage in decision-making, and centralized distribution and management of access policies.

To specify rules, NuAppFirewall includes a catalog with more than 250,000 access rules, automatically generated from over 21 million network access logs collected over 33 days. These logs, which have been validated and provided by Nubank, were gathered from real-world usage by multiple employees. They underwent a rigorous validation and refinement process to ensure that the rules accurately reflect legitimate access patterns.

These rules have been designed to cover different decision-making scenarios, ranging from granular blocks (specific combinations of application, endpoint, and port) to broader restrictions, blocking access to all endpoints and ports for a given application. Additionally, tests have demonstrated that 99% of CPU usage measurements remained below 8.7%, confirming the efficiency of the solution.

The application and its set of rules are remotely distributed to Apple devices by the information security team through the Mobile Device Management (MDM) service adopted by Nubank. This ensures that all access policies are centrally defined, applied, and audited, eliminating the need for manual configurations by users.

This paper is structured as follows: the Background section presents essential concepts for developing firewall applications in the macOS environment and the state of the practice before the development of this solution. Then, in Section 3, we detail the functionalities and components of NuAppFirewall, including rule generation, access auditing, and deployment on Nubank devices. The Evaluation section discusses validation through testing, performance assessment, and production experience at Nubank. Finally, the Conclusion provides final considerations and future directions.

2. Background

This section explores key concepts and definitions, along with an overview of the firewall solutions currently available for macOS.

2.1. Network Extension

Network Extension is a type of System Extension [Apple 2025b], a framework that enables extending macOS functionalities by operating in user space rather than at the kernel level. Specifically, Network Extensions can intercept, modify, or redirect network traffic on the device, allowing for the customization and enhancement of network functionalities on macOS. This framework facilitates the development of network-related applications

and services, such as VPNs, firewalls, and other network management solutions. In this project, we used the Network Extension framework to intercept traffic securely and stably, enabling a firewall compliant with corporate requirements like Nubank's.

2.2. Network Flow and Rules

In the context of the Network Extension framework, a network flow refers to an abstraction that represents a network connection that will be monitored, analyzed, and potentially filtered by extensions using the framework's API. This abstraction encapsulates information about the network traffic passing through the device, such as URL, host, port, and flow direction. This information enables applications to intercept, modify, or block such connections based on predefined rules.

To allow or block network flows, administrators specify access rules, which serve as instructions guiding the firewall's behavior when monitoring and filtering traffic between networks. In NuAppFirewall, the rules determine the action to be taken on a specific network flow based on the endpoints, ports, or application specified in the rule's body. Each rule has the following attributes, as defined in Table 1.

Table 1. Access Rule's Attributes and Definitions

Attribute	Definition
ruleID	Unique identifier for the rule
action	Action to be applied (allow or block)
application	Bundle ID, full path, or subpath of the application requesting the connection
endpoint	URL, host, or IP address to which the rule applies
port	Port associated with the endpoint to which the rule applies
destination	Connection destination, consisting of the "endpoint:port" pair

2.3. Mobile Device Management (MDM)

Mobile Device Management (MDM) [IBM 2025] enables organizations to remotely manage and secure devices. It provides standardized configuration, large-scale application deployment, policy enforcement, and remote monitoring capabilities.

Apple's MDM framework, introduced in macOS 10.7, uses the Apple Push Notification service (APNs) to communicate between MDM servers and devices. The framework offers two primary configuration tools: Configuration Profiles (.mobileconfig XML files) for device and user-level settings, and Policies for automating complex tasks like software distribution. In this work, we leverage MDM to centrally manage application deployment and firewall configurations across Nubank's device fleet through Configuration Profiles and automated policies.

2.4. Related tools

Among the existing firewall applications for macOS, we highlight Little Snitch [Objective Development 2025] and LuLu [Objective-See 2025], both of which enable network traffic control and are designed for personal use. Little Snitch, a proprietary software, allows the creation of multiple rule sets that can be associated with different networks. However, its commercial nature imposes restrictions on adaptations for corporate environments. Alternatively, LuLu, a free and open-source tool, monitors outbound connections in real time. However, it exhibits some structural deficiencies, such as excessively long functions, highly coupled and complex code, and a lack of unit tests, making its evolution challenging.

Furthermore, neither Little Snitch nor LuLu provides a predefined set of rules based on real-world usage scenarios, an essential feature for ensuring greater reliability and security in network traffic control. Given these limitations, it becomes evident that the development of an alternative solution is necessary, one that is more closely aligned with the security and performance requirements essential in corporate contexts.

3. Our Solution: NuAppFirewall

Designed to meet corporate security demands, NuAppFirewall is an open-source firewall built to enhance network security in macOS environments. Leveraging a robust catalog of over 250,000 automatically generated rules, the application intelligently filters and validates network traffic. These rules are derived from log analysis performed by the Endpoint Detection and Response (EDR) tool adopted by Nubank. With NuAppFirewall, macOS users gain an extra layer of security, proactively identifying and mitigating potential threats.

Figure 1 represents the architecture of NuAppFirewall. In Section 1 of the figure, the access rule development pipeline is presented. The process begins with the analysis and processing of the dataset from the EDR tool, along with a CSV text file containing the applications to be blocked and their respective bundle IDs. This data is handled by the RulesParser module, which generates the rule file in a structured format (JSON or PLIST).

The distribution of access rules to corporate devices is carried out via MDM solution, as shown in Section 2 of Figure 1. This can be done in two different ways, depending on the size of the rule set: (i) using an installation policy, where the rule file is downloaded to a specified directory for larger rule sets; and (ii) via a configuration profile, embedding the rules as a value within the file for smaller sets. The Utils module imports and formats the rules, storing them in data structures managed by the RulesManager module, as detailed in Section 3 of Figure 1.

The ExtensionManager intercepts each flow and forwards it to the NuAppFacade, which then redirects the request to the FlowManager. The FlowManager retrieves access rules from the catalog, stored in data structures managed by the RulesManager, and applies these rules to determine whether the flow will be allowed or blocked.

3.1. Rule Generation

The rule catalog consists of both permission and blocking rules (Allow and Block), each with different inputs. For Allow rules, as mentioned previously, a comprehensive analysis

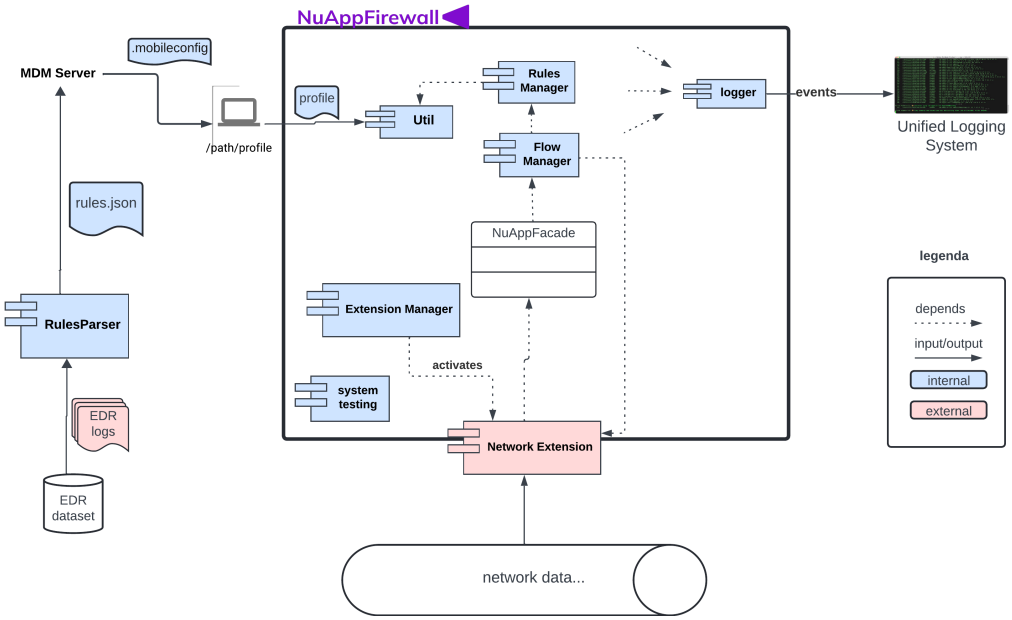

Figure 1. NuAppFirewall's general architecture

is conducted on logs derived from the EDR tool adopted by Nubank. These logs record multiple requests, containing details about which application performed the access, the accessed endpoint, and the corresponding port used, with each line representing a logged event. Based on this information, multiple specific Allow rules are created, reflecting the captured access patterns from the EDR solution.

In the context of Block rules, their nature is slightly different, as there are no rules as granular as the Allow rules; in this case, only rules that block an entire application exist. To achieve this, a consultation was conducted with the Product Owner to identify the applications that should be blocked by NuAppFirewall. Additionally, it is important to highlight that, due to the nature of this type of rule, endpoint and port information become irrelevant, which is why it is not included in the input for generating such rules.

From these inputs, a single file is generated to be used by NuAppFirewall: the catalog itself. This catalog can only be produced in JSON or PLIST formats and contains more than 250,000 rules. The file includes unique keys that represent applications. From a given application, it is possible to retrieve other relevant information, such as the nature of the associated rules (Allow or Block), the Identifier associated with the application, and a list of "Destinations" — a term that refers to a tuple consisting of an endpoint and a port, representing an access instance according to the nature of the rule in question. The scripts used for the catalog generation are available on NuAppFirewall-Catalog's[1] repository.

3.2. Access validation workflow

The access validation process in NuAppFirewall follows an approach that prioritizes specificity in application identification and security through the predominance of blocks in case of conflicts. This workflow is divided into three main stages: data extraction, search for applicable rules, and their application.

[1]https://github.com/nufuturo-ufcg/NuAppFirewall-catalog

In the first stage, the system captures information from the network flow, such as the application's unique identifier (bundle ID) and its full path in the file system, the destination (URL, host, or IP), and the port used. This data is used to correlate the connection with the appropriate rule to be applied.

In the second stage, the applicable rules are searched following a hierarchy of specificity: first, the rules associated with the bundle ID; then, those based on the application's full path; and finally, the search for subpaths is used. Figure 2 illustrates the hierarchical structure of rule specificity in NuAppFirewall, where rules are applied with increasing granularity levels. The combination of different attribute values allows the application to function in distinct operational modes and enforce varying degrees of network flow restrictions.

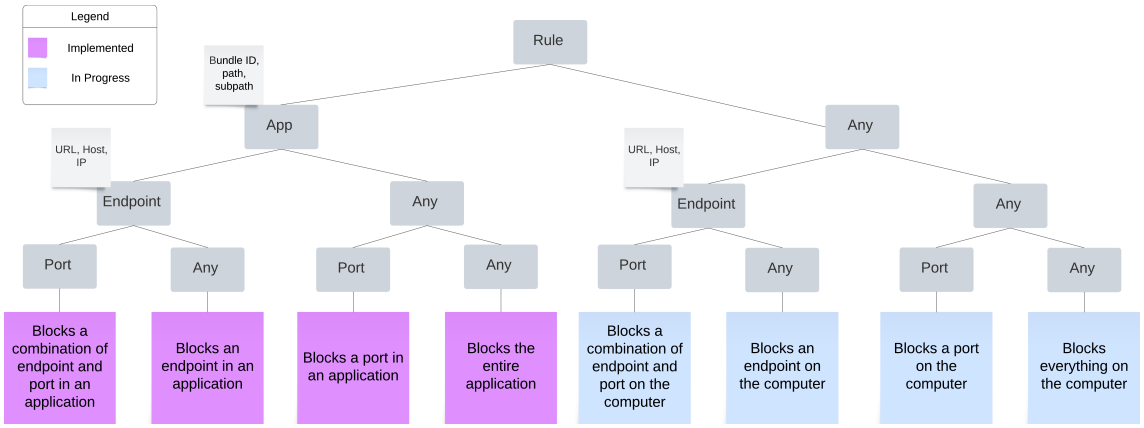

Figure 2. Rule validation tree

To optimize the search, a set is used as a "flag" to indicate the existence of rules associated with an application. If such rules exist, all parameter combinations (URL, host, IP, and generic) are retrieved at once, in constant time (O(1)), through dictionaries of dictionaries. This eliminates separate searches for different destination combinations or rule types (block and allow), allowing all rules for the flow to be filtered and prioritized at once.

In the third stage, the selection of the appropriate rule occurs within the set retrieved in the previous stage. If no rule is found, the system returns to the search stage to evaluate the next level: rules based on the full path and, if necessary, subpaths. If more than one rule is found, prioritization starts with general rules and progresses to more specific combinations. At any level, blocks always take precedence over permissions, ensuring that the default-deny principle is upheld.

Currently, the application operates in passive-allow mode, meaning that if no rule is found at any level, the connection will be allowed. However, we are working on the creation of new modes, such as passive-block, which will reverse this behavior by denying the connection if no matching rules are found.

3.3. MDM Integration

We deployed our application through the MDM solution adopted by Nubank. The configuration profile deployed to the devices grants permissions for the extensions utilized by

the application, access to protected files, assignment of the filter type and filter strength (in this case, the firewall plug-in), and designates the extension as a packet filter provider. This approach eliminates the need to request authorization for these actions from the machine administrator.

The installation policy is executed in two stages: first, the rule package is downloaded to the designated directory, followed by the download of the release version of the application package. These mechanisms ensure that NuAppFirewall and its rule catalog are distributed and executed on the selected machines. Starting with version 2.0.0, the rule set can also be distributed via configuration profile, thus enhancing control and monitoring through MDM and facilitating the streamlined updating of the rules.

Within the MDM environment, distribution can be customized based on groups and users, thereby allowing for more tailored content filtering and enabling the use of distinct rule sets that cater to the specific needs of each team.

4. Evaluation

4.1. Evaluation and Validation

The validation and evaluation of NuAppFirewall involves system testing, unit testing, and performance analysis to ensure its effectiveness and reliability. In system testing, test rules are deployed, and the firewall is activated. Connections are then made to the specified URLs, and macOS syslog logs are analyzed. The results are classified into three categories: absence of a corresponding log, discrepancy in rule application, or full compliance.

Unit tests are conducted using functions that generate all possible permutations of data accepted by the rules and flows. The validation process includes the initialization, addition, and removal of rules, the precedence of block rules over allow rules, and the rejection of invalid rules. Flow tests ensure the correct application of rules in simulated connections and the adoption of the default behavior (allow) when no rule applies. Coverage analysis revealed 100% coverage of critical functions. 17 tests were performed, and due to the parameter combinations, in some cases, each test executes 48 rule variations with 48 flow variations, totaling 2,304 executions in a single test.

For the performance analysis of NuAppFirewall version 2.0.0, the system was tested on five MacBook Air M1 (8GB RAM) devices, with a focus on resource consumption and stability. A total of 308,709 data samples were collected during the usage period, with automatic data collection using the psutil 6.1.0 library, which recorded an average CPU consumption of 0.72% and 10.97 MB of RAM. The Cumulative Distribution Function (CDF) was employed for a detailed analysis of CPU consumption. The results indicate that 99.9% of the measurements recorded a maximum consumption of 22%, while 99% remained below 8.7%. Additionally, 90% of the samples showed consumption of up to 2.2%, and in 50% of the cases, CPU usage was nearly zero. During the initial seconds following the extension's activation, expected spikes occurred, reflecting the intensive processing required at that moment. These results demonstrate that CPU consumption remains low for the majority of the time, with spikes confined to the initial moments of extension activation.

4.2. Deployment in Production

After developing the tool and stabilizing its first version in a testing environment, deployment within Nubank's infrastructure began incrementally on December 10, 2024, starting with 16 machines from the information security team. Remote workstations from selected teams were gradually added, allowing close monitoring of system behavior and user feedback as the application ran silently in the background. Although no malfunctions have been reported so far, the need to grant user-specific exceptions soon emerged to support testing and operational requirements, such as allowing access to browsers blocked by default, a functionality now recognized as essential for future implementation.

Building on this initial success, we are currently executing an expansion strategy to broaden the tool's adoption across the organization. Our phased rollout plan encompasses not only adjacent technical teams but aims to establish the solution as the standard firewall management system throughout Nubank's corporate infrastructure. This enterprise-wide implementation strategy includes developing comprehensive documentation, establishing support processes, and creating scalable deployment mechanisms to facilitate adoption across diverse departments and use cases within the organization. The expansion initiative is designed to maintain the same level of security and reliability demonstrated in the initial deployment while accommodating the varied requirements of different business units.

5. Conclusion

In this article, we presented NuAppFirewall, a tool designed to provide a robust, efficient, and corporate-ready firewall solution that can benefit various companies with security demands similar to those of Nubank. By combining an access monitoring-based rule catalog with a modularized codebase adapted for distribution via MDM, we achieved competitive results.

Future system enhancements will focus on expanding access control capabilities. A new passive-block mode will be implemented to deny unauthorized access attempts, while server-based administration will enable centralized management of user and group access rules through configuration profiles. The system's validation scope will also be broadened beyond its current ability to restrict access at the application and endpoint level, allowing for port-specific controls within applications and system-wide rule enforcement.

Executable, documentation and demonstration: The source code, documentation, installation steps, video guide and executable are available on NuAppFirewall's repository[2]. The demonstration will be held with the usage of the authors' devices, in the following order: (i) distributing the configuration profile using an MDM service; (ii) distributing the application and catalog package through the same MDM service; (iii) showing the logs in the devices to demonstrate the application being started on the machine; (iv) accessing blocked and allowed applications with NuAppFirewall running.

Acknowledgements. This work was developed under the NuFuturo project, a collaboration between the Federal University of Campina Grande and Nubank.

[2]https://github.com/nufuturo-ufcg/NuAppFirewall

References

Apple (2025a). Network extension documentation. Available at https://developer.apple.com/documentation/NetworkExtension. Accessed: 24/01/2025.

Apple (2025b). System extensions documentation. Available at https://developer.apple.com/documentation/systemextensions. Accessed: 24/01/2025.

IBM (2025). Mobile device management. Available at https://www.ibm.com/br-pt/topics/mobile-device-management. Accessed: 27/01/2025.

Nubank (2024). Nubank. Accessed: 2024-02-13.

Objective Development (2025). Little snitch. Available at https://www.obdev.at/products/littlesnitch/index.html. Accessed: 29/01/2025.

Objective-See (2025). Lulu - the macos firewall. Available at https://objective-see.org/products/lulu.html. Accessed: 29/01/2025.