A Catalog of Security Patterns

ANDRÉ CORDEIRO, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
ANDRÉ VASCONCELOS, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal
MIGUEL CORREIA, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Security patterns offer a time-tested solution for common security issues. Patterns can help organizations develop secure software and achieve compliance with ever-increasing regulations. However, an analysis of the existing literature shows that there is a lack of uniformity when describing security patterns and a lack of an up-to-date catalog, as patterns are spread across multiple papers and publication venues, leading organizations to discard their use. Our work solves these gaps by developing a concise pattern catalog by analyzing over 295 security patterns and filtering them based on quality metrics, as well as the means to expand it further as new security patterns emerge. We also use our catalog to improve the Portuguese national cybersecurity reference board in order to help organizations achieve compliance.

CCS Concepts: • **Security and privacy → Software security engineering**; Security services; Systems security; • **Software and its engineering →** *Designing software.*

Additional Key Words and Phrases: security patterns, security requirements, software architecture, cybersecurity

ACM Reference Format:
André Cordeiro, André Vasconcelos, and Miguel Correia. 2022. A Catalog of Security Patterns. 22 (October 2022), 14 pages.

1 INTRODUCTION

The concept of *pattern* was first introduced in the field of Architecture in 1977 by Alexander [2]. He defined a pattern as the core solution for a problem that appears in a certain context. Later, in 1994, Gamma et al. expanded this concept to software design [12]. They explored patterns as a way to improve the design of reusable object-oriented software, arguing that what makes designers experts in the field is the knowledge of what worked before. Patterns allow a designer who understands them to apply them immediately on design problems without having to rediscover the solution. The concept of pattern started to be applied in the area of cybersecurity in 1997 [32].

Since 1997, the *security pattern* research landscape has evolved and matured, with more than 400 publications in the area. The main topics of research have been new security patterns, pattern classification, pattern templates, pattern validation, pattern catalogs, pattern evaluation, and pattern applications [15]. Today, security patterns can help reduce the gap between management and information technology (IT) positions, helping them make the right decisions about cyber security. Patterns have also been shown to help organizations secure their architectures without compromising their security needs [9]. However, there is currently a large discrepancy between the use of security patterns in academia and their use in industry, with only about 16% of research conducted in an organizational environment, which means that organizations have yet to fully adopt security patterns as a means of achieving secure software [15].

Authors' addresses: André Cordeiro, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rual Alves Redol 9, Lisboa, 1000-029, Portugal, andre.santos.cordeiro@tecnico.ulisboa.pt; André Vasconcelos, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rual Alves Redol 9, Lisboa, 1000-029, Portugal, andre.vasconcelos@tecnico.ulisboa.pt; Miguel Correia, INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Rual Alves Redol 9, Lisboa, 1000-029, Portugal, miguel.p.correia@tecnico.ulisboa.pt.

One of the main reasons for the lack of adoption in the industry is the heterogeneity of the sections in *templates* used to describe security patterns [4]. More than 50 different sections have been used to describe security patterns since 1997. Most of these sections are a specialization of already existing sections, for example section "Trade-Offs" that describes the trade-offs when applying a pattern is a specialization of the section "Consequences", meaning that they do not improve the security pattern descriptions, and instead increase the complexity of descriptions, deterring organizations from their usage, as it would take developers, designers, etc. too much time to go through and understand each pattern, as they would need to learn multiple description templates.

Another reason for the lack of adoption of security patterns in the industry is the lack of a comprehensive and up-to-date security pattern *catalog* that brings together existing, well-defined, security patterns in one place and relates the patterns between themselves to help guide developers through a step-by-step process for pattern instantiation [29]. The lack of an up-to-date catalog implies that current organizations have to either build a pattern catalog of their own before they can actually use the patterns in their architectures or choose a preexisting catalog that is either outdated and does not support new emerging technologies like cloud computing or the Internet of Things (IoT), hard to navigate, or lacks quality in their pattern descriptions. This high fragmentation of patterns over a large number of publications also impacts researchers, as most of the time they have to first build a pattern catalog in order to be used as a basis for conducting their research. Due to the time required to build a concise and diverse catalog that covers a wide range of security requirements, most researchers build their catalogs with limited scopes, so as not to sacrifice their actual research focus, leading most studies to make uncertain conclusions due to the limitations of the catalog used [34].

As mentioned above, security pattern catalogs are not a new research topic, as several catalogs have been established over the years. The most impactful catalog was presented by Schumacher et al. in 2006 and contains 46 security patterns, using the same template for all pattern descriptions [24]. In 2006, a technical report collected 35 security patterns from existing research that "applied a reduction process to both simplify the pattern landscape (...) and remove heterogeneity" [33]. However, compared to the work of Schumacher et al. the template used to describe the patterns misses a lot of relevant information necessary to instantiate the pattern, with most descriptions lacking clarity. More recently, Fernandez-Buglioni published a follow-up book that expanded the work of Schumacher et al. improving some of the patterns and adding new patterns, to a total of 68 security patterns [10]. The most recent catalog we could find was presented in 2021 by Papoutsakis et al. and focused solely on IoT [21]. Because it focuses only on that specific area, it has a limited scope.

Some of these catalogs are well described but are outdated, missing newer patterns for emerging technologies (e.g., cloud computing), a possible approach for obtaining an up-to-date catalog would be to update one of these catalogs with newer patterns. However, since none of these catalogs provides a description of the process used to build them, like the extraction process used, the filtration used if any, etc., it is impossible for researchers or organizations to update them or even verify their completeness.

Our work introduces a uniform and up-to-date catalog of security patterns. This catalog was built using a systematic process that ensures its quality and allows researchers and organizations to expand it as new patterns emerge. Our work also expands the navigability of the catalog by classifying the patterns against security properties like confidentiality, integrity, authentication, etc. that can be used by organizations to choose the patterns depending on the security property they are aiming to achieve. This catalog can be used as both a reference for organizations to build secure systems and for researchers to accelerate their work without the need to build a catalog that distracts them from their research focus.

In order to further reduce the gap between the use of patterns in academia and industry, we explored a use case for security patterns. We matched the security patterns in the catalog with the categories described in the Portuguese cybersecurity reference board [6], expanding the information that organizations have access to, to achieve compliance with the regulation.

Table 1. "Search space exclusion criteria." from [15]

#	Description
1	If the journal is not indexed in JCR AND SJR is below 0.4
2	If the conference/workshop has a Qualis below B3 OR ERA below B OR H5- Index below 5
3	If the journal/conference/workshop metrics are not available

The remainder of the document is structured as follows. Section 2 describes the process used to build our pattern catalog and the results of applying that process. Section 3 describes the results of using our catalog to improve the Portuguese national cybersecurity framework. Section 4 describes how we evaluated our catalog using attack patterns. Section 5 describes the shortcomings of our work and finally, Section 6 concludes this document.

2 BUILDING THE CATALOG

This section explains the four phases of the process used to build the pattern catalog, why they were necessary, and the results that we achieved applying each phase when developing our catalog.

The first phase, extraction, corresponds to the discovery and selection of previously published patterns to be used for the rest of the building process. The second phase, filtration, is when we refine the previously extracted patterns using qualitative metrics to ensure both the quality of the catalog and its size, making it easier to navigate. The third phase, conversion, is where we take all the accepted patterns from the previous phase and translate their description into a standard template that is used in all the patterns in the catalog, ensuring the uniformity of the description. The final phase, classification, is where we categorize the patterns using a classification scheme that enriches the navigability of the catalog for organizations or researchers who wish to use the catalog. Finally, subsection 2.5 provides the results of applying each phase when building our catalog.

2.1 Extraction

The extraction phase focuses on the discovery of security patterns from previously published research. For this phase, we decided to start with the papers identified in the study by Jafari and Rasoolzadegan [15], which includes 274 papers on security pattern research. Using their work as a starting point, we were able to accelerate the pattern extraction process.

This study does not analyze security pattern research later than 2017, however, since our aim was to build an updated catalog, we expanded the searching strategy to papers published up to 2021. We used the same search strategy for articles beyond 2017 as described in the study, consisting of a manual search, a database search, and backward snowballing [30].

For the search string used for the manual and database searches, we made a slight change from the study. Although in the study they used keywords "security patterns" and "misuse/threat pattern", we decided to use only "security patterns" as our catalog focuses only on security patterns. Furthermore, we applied the search string to the same Title, Abstract, and Keywords sections of the papers. The databases used for our search were IEEE Xplore, ACM Digital Library, Springer Link, ScienceDirect, and Scopus.

For the paper exclusion/inclusion criteria, we mainly used the same inclusion/exclusion criteria as Jafari and Rasoolzadegan (see Tables 1 and 2). However, we decided to change the paper exclusion criteria of Jafari and Rasoolzadegan so as not to exclude books due to their influence in the field, with the two most complete pattern catalogs to date being two of those books [24][10]. A complete description of the inclusion/exclusion criteria and the reasons behind them can be found in Sections 2.1.5 to 2.1.7 of their study [15].

Table 2. "Paper exclusion criteria." modified from [15]

#	Description
1	The paper belongs to an excluded search space
2	The paper is published in a conference/workshop with a period of below 5
3	Paper/Book cannot be accessed (e.g., not indexed)

2.2 Filtration

This phase focuses on improving the catalog by only including well-constructed security patterns. To do this, we filtered the patterns identified in the previous phase. For us, this filtration phase was necessary in order to ensure the catalog quality and readability by removing lackluster patterns, as well as repeated patterns.

In other to filter out these kinds of pattern, we established simple criteria that the patterns needed to follow using as a basis the metrics established by Laverdiere et al. [17]. We decided to be as strict as possible when filtering the patterns in order to condense the catalog as much as possible, making it easier to navigate, so any pattern that failed to meet one criterion was discarded from the catalog. The quality criteria used to filter the patterns were as follows:

- Reusability: The pattern can be used in different contexts.
- Compositionality: The pattern can be used with other patterns without losing the security properties that it or the other patterns provide.
- Validability: The implementation and usage of the pattern can be validated.
- Platform independence: The pattern can be used independently of programming languages, operating systems, frameworks, technologies, hardware, standards, protocols, etc.
- Pattern independence: The pattern does not depend on patterns excluded by our filtration phase. This does not include patterns unrelated to security, like design patterns.
- Description completeness: The description of the patterns needs to provide, at the very least, a name for the pattern, the problem it aims to solve, and the solution for that problem.
- Security properties relationship: The pattern must be related to security properties (e.g., confidentiality, integrity, availability, etc.).
- Noncomposite: The pattern is not just a collection of other patterns that can be used to fix a specific problem.

Note that the filtration process was made chronologically from oldest to newest to ensure that the patterns on which the current pattern depends are already evaluated and filtered as per the "Pattern independence" criteria.

Finally, after filtering all patterns using the criteria, we manually analyze each pattern in order to remove repeated or similar patterns.

2.3 Conversion

The conversion phase is where we uniformized the descriptions of all patterns into a single template.

For the description of all patterns, we decided to use the template described in the book "Pattern-oriented software architecture" (POSA) [5], which is supported by research on the heterogeneity of pattern templates published in 2014 [4]. In this paper, the authors analyzed different templates used to describe security patterns and identified that security patterns do not require specialized sections for their descriptions, as such, the template used to describe security patterns can be the same as the one used for other types of patterns, such as design patterns.

Table 3. Template sections used to describe the patterns.

Section	Description
Name	The name of the pattern.
Intent	A short description that summarizes the key points of the pattern.
Example	"A real-world example demonstrating the existence of the problem and the need for the pattern." [5]
Context	"The situations in which the pattern may apply." [5]
Problem	"The problem the pattern addresses, including a discussion of its associated forces." [5]
Solution	"The fundamental solution principle underlying the pattern." [5]
Structure	"A detailed specification of the structural aspects of the pattern." [5]
Dynamics	"Typical scenarios describing the run-time behavior of the pattern." [5]
Implementation	"Guidelines for implementing the pattern." [5]
Example Resolved	"Discussion of any important aspects for resolving the example that are not yet covered in the Solution, Structure, Dynamics and Implementation sections." [5]
Variants	"A brief description of variants or specializations of the pattern." [5]
Known Uses	"Examples of the use of the pattern, taken from existing systems." [5]
Consequences	"The benefits the pattern provides, and any potential liabilities." [5]
See Also	"References to patterns that solve similar problems, and to patterns that" complement the pattern in case. [5]
References	Documents, studies, papers, books, etc. that are referenced in the other sections of the pattern.
Sources	The sources such as research papers, books, etc. where the pattern was proposed.

Although the study advocates not to add new sections to the design pattern template, we decided to go against this and added two additional sections related to the metadata of the patterns: "References" and "Sources". The References section was added in order to isolate the patterns from the catalog itself, since that, if the references related to each pattern were at the end of the catalog, then organizations would have a harder time extracting the necessary patterns that apply to them from the catalog. The Sources section specifies where the pattern was taken from giving credit to the authors and also providing the users of the catalog with an avenue to learn more about the pattern, such as the reasons behind it.

The complete list of sections and their description is shown in Table 3. All patterns in the catalog that used different sections in their descriptions were manually converted to our template.

2.4 Classification

It is not easy to navigate a catalog of patterns due to their size. To improve the navigability of the catalog, we classified all patterns in the catalog into a number of sets, each with their respective properties.

We chose to classify our catalog using the classification scheme by Yskout et al. [33] that classifies the patterns based on the development phases they apply and the security objectives they help achieve. Table 4 provides each classification category and its description. Note that a security pattern can be classified with more than one security property.

We chose this classification using the research of Alvi and Zulkernine that compared security pattern classification schemes based on their objectives, descriptions, and dimensions [3]. They compared a total of 29 classification schemes on their navigability, completeness, acceptability, comprehensibility, determinism, mutual exclusivity, repeatability, and usefulness. They identified the work of Yskout et al. [33] as the most complete classification scheme of the ones analyzed and as such we decided to use it.

Table 4. Description of each category by Yskout et al. [33] of the classification scheme used.

Category	Description
Application Architecture	A pattern is an application architectural pattern if its introduction has system-wide implications.
Application Design	A pattern is an application design pattern if its introduction only has local implications.
System	A pattern is a system pattern if its introduction has its main effects on the environment in which the application will be deployed, ideally independent of the application.
Confidentiality	Sensitive data can only be read by the intended audience.
Integrity	Resources have not been tampered with.
Availability	The system is available and responsive for its users.
Accountability	The tracing of important (or all) actions performed on the system back to a particular user.
Non-repudiation	The undeniability of having performed an action.
Anonymity	The state of being unidentifiable within a group of entities.
Privacy	The right to be informed and control the flow of information about oneself.
Secure data transmission	Protection of data in transit.
Controlled Access	Restriction of access to operations on resources to authorized entities only.
Identification	The claims an entity makes about its identity. The verification of these claims is called the authentication.

2.5 Results

This section provides the results of building our catalog using the process previously described.

As shown in Figure 1, using our extraction process we identified 309 studies, articles, and books from 1997 to 2021 on security pattern research (274 from 1997 to 2017, 31 from 2018 to 2021, and 4 books). After identifying this research, we individually analyzed each publication to check for the description of security patterns. In the end, of those 309 publications, 104 described one or more security patterns, for a total of 295 security patterns identified.

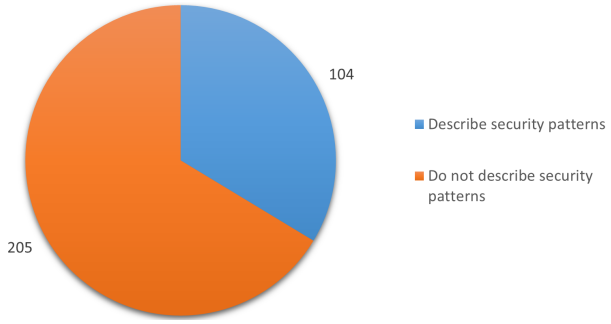

Fig. 1. Percentage of research that describes at least 1 security pattern.

Using our filtration phase, we were able to remove 216 low-quality patterns and 113 repeated patterns from the 407 security patterns identified, we were left with a total 106 security patterns (see Figure 2). Some examples of low-quality patterns included the:

- "Whitelist" pattern from [23] that protects IoT devices against unknown attackers by only allowing communication with identified partners and blocking all other communication. Although the pattern itself

is useful, there is no reason for it to be tied to the IoT and instead should have been described in a way that makes it possible to apply it in other contexts.
- "BlockBD: Blockchain Security Pattern for Big Data Ecosystems" pattern from [20] that improves the security of Big Data ecosystems; however, it can only be applied when using a specific security reference architecture developed by the same authors, making the pattern very situational and useless in most cases.
- "Secure Software as a Service" pattern from [19] that describes how to achieve data confidentiality when using outsourced solutions such as Google Apps for business or Amazon Cloud services. However, the solution of this pattern requires the pattern "Hidden Storage" that, as far as we know, has yet to be published. Without this "Hidden Storage" pattern, this pattern cannot be implemented.
- "Location-aware" pattern from [25] that advocates security modules to be at a certain distance from a specific location. For example, firewalls should be close to the source to eliminate traffic as soon as possible, reducing traffic. However, the description is very limited and does not describe what kind of security modules this pattern applies to, and what is the actual problem it aims to fix.
- "Privacy Policy Icons" pattern from [27] that describes how to make it easier for users to be aware of our privacy policies, by using specific icons that can be glanced over to get the general idea of an organization policies. While this pattern makes sense, it is not a security pattern but a usability or privacy pattern, as it does not actually increase the security of a system.
- "Secure Software Container" pattern from [28] that describes how to secure a software container using security patterns. This pattern does not describe a new solution and instead uses other security patterns to solve the problem. Since this pattern does not provide a new solution, it only increases the size and complexity of the catalog.
- "Increase monitoring within a time window of a negative workplace event" pattern from [11] that advocates organizations to increase monitoring of their systems right after a negative event, such as employee layoffs. This is not actually a pattern, but a set of best practices for organizations to follow and, as such, should not be in a catalog of patterns.

The number of repeated patterns was high mainly due to the inclusion of older pattern catalogs in our extraction phase, as some of them already cataloged some of the research that was analyzed. An example of such a repeated pattern is the "Authorization" pattern that appears in 4 different publications [8], [22], [24], and [10]. All of the "Authorization" patterns describe the same pattern and as such we chose to include only the best described pattern and removed the others.

For our conversion phase, most authors tended to use the same template that we have described. However, for patterns described using different templates, such as the pattern "Administrator Object"[16], with sections such as "Resultant context", we converted these patterns into our template using the mapping developed by M. Bunke [4]. So, for example for the "Administrator Object" pattern, the section "Resultant Context" became the "Example Resolved". By making this conversion, it becomes easier for entities to use the catalog as they no longer need to familiarize themselves with multiple types of descriptions.

We were able to convert all the patterns analysed into our pattern template however, some of the sections in our template were still incomplete, as the authors of some patterns did not provide enough information in their description to fill all sections of our template. About 36% of the patterns did not have enough information to create a complete description using our catalog (not counting sections "Structure", "Dynamics", "Variants", and "See Also", which are not required for some patterns). The least described sections were the "Dynamics", "Structure", "Example Resolved", "Implementation", and "Example" with 68.6%, 75.4%, 80.5%, 82.2%, and 91.5% completion, respectively. The sections most described were the "Problem", the "Solution", and the "Context" which are considered the bare minimum for a pattern to be understood.

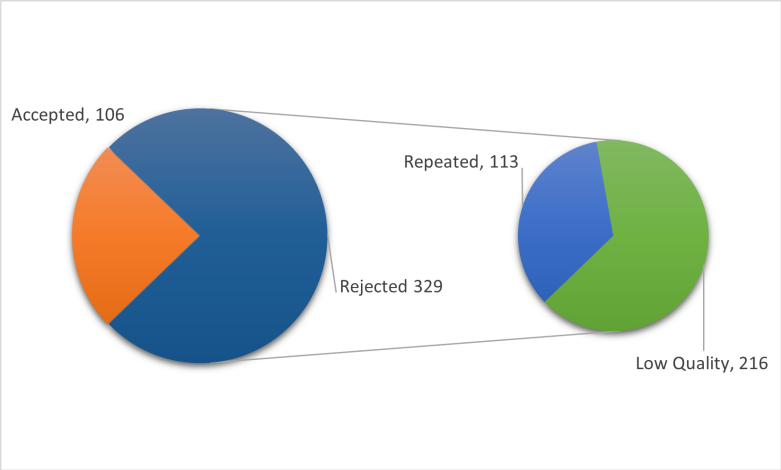

Fig. 2. Number of accepted, rejected and duplicate patterns

When classifying the security patterns, it became evident that the classification scheme we choose fails to classify the security patterns related to the analysis phase of system development such as "I&A Requirements" and "Audit Requirements", as these types of patterns cannot be classified using the existing categories. In order to classify these patterns, we decided to extend the development phases categories with a new one named "analysis" which include patterns that are applied before building a system, such as requirements gathering patterns and assessment patterns.

By using this improved classification scheme, we were able to classify all patterns in our catalog, and create a number of sets, each with different applications and properties. For example, an organization that wants to control access to their systems no longer needs to go through the complete catalog to find the patterns that apply to their situation. Instead, they check which patterns provide access control mechanisms and which of those affect the system and are then left with a small set of 19 patterns that are specifically related to what they are looking for.

Of the classified patterns, 47 are related to application design, 30 are related to the system in which the software runs, 15 are related to the analysis of a system, and 13 are related to the application architecture. For the security properties, the security property with by far the most patterns is access control with 60 patterns, followed by identification, accountability, integrity, confidentiality, and secure data transmission, each with 19 to 25 patterns. Anonymity, privacy, availability, and non-repudiation have the least number of patterns each with 11 to 13 security patterns; this is due to the fact that our catalog is focused on security patterns and does not analyze privacy or reliability patterns that would better relate to these properties [18].

Table 5 presents the names of all the patterns in our catalog. The full description of the patterns as well as all the information regarding each phase can be found at https://github.com/xextreme22/Security-Patterns.

3 USING THE CATALOG

Regulations are complex, sometimes redundant and inconsistent [31]. Yimam and Fernandez advocated the use of patterns as a way to handle compliance complexities, uncertainties, and overlaps and that this approach "could improve compliance, security, privacy, and overall software quality". They concluded their work by stating that "an architecture built from patterns can be used as a common language among architects, developers, business

Table 5. Complete names of all patterns in the catalog.

Abstract IDS	Abstract Virtual Private Network	Access Control List
Access Control Requirements	Access Control to Physical Structures	Actor and Role Lifecycle
Actor-based Access Rights	Administrator Hierarchy	Administrator Object
Alarm Monitoring	Anonymity Set	Asset Valuation
Asymmetric Encryption	Audit Requirements	Audit Trails and Logging Requirements
Authenticator	Authorization	Automated I&A Design Alternatives
Behavior-Based IDS	Biometrics Design Alternatives	Capability
Check Point	Circle of Trust	Content-Dependent Authorizer
Context-Enhanced Authorizer	Controlled Access Session	Controlled Object Factory
Controlled Object Monitor	Controlled Process Creator	Controlled Virtual Address Space
Credential	Demilitarized Zone	Digital Signature with Hashing
Discretionary Access Control	Dynamic Integrity Protection	Enterprise Partner Communication
Enterprise Security Approaches	Enterprise Security Services	Execution Domain
Fault Container	Fine-Grain Access Control	Front Door
Hidden Metadata	History-Based Authentication	I&A Requirements
Identity Federation	Identity Provider	Information Obscurity
Input Guard	Integrity Attestation (Remote Attestation)	Integrity Protection
Intrusion Detection Requirements	Known Partners	Layered Encryption
Layered Operating System	Mandatory Access Control	Microkernel Operating System
Modular Operating System	Morphed Representation	Multilevel Secure Partitions
Multilevel Security	Mutual Authentication	Neuralyzer
Non-Repudiation Requirements	Output Guard	Packet Filter Firewall
Password based key exchange	Policy Enforcement	Policy-Based Access Control
Privilege-Limited Role	Protected Entry Points	Protection Reverse Proxy
Protection Rings	Proxy-Based Firewall	Reference Monitor
Relays	Risk Determination	Risk-Based Authenticator
Role Rights Definition	Role-Based Access Control	Role-Hierarchies
Secure Boot (Authenticated Boot)	Secure Channels	Secure Embedded Logging
Secure GUI System	Secure Pre-Forking	Secure Storage
Security Accounting Requirements	Security Logger and Auditor	Whitelisting Firewall
Security Session	Signature-Based IDS	Single Access Point
Software Container	Stateful Firewall	Static Integrity Protection
Symmetric Encryption	Third Party-Based Authentication	Threat Assessment
Transport Layer Security	Trust Trap Mitigation	Virtual Address Space Structure Selection
Virtual Machine Operating System	Vulnerability Assessment	web of trust
Security Needs Identification for Enterprise Assets		

owners, managers, service providers, and auditors. It can also be used as a reference to design and implement automated systems that can be used for compliance verification.

In 2015, Alaküla and Matulevičius [1] analyzed how security patterns could help an insurance company achieve business process compliance with the ISO/IEC 27001:2013 [14] standard by analyzing their business processes before and after the security risk-oriented patterns are applied and security constraints are introduced. They observed that patterns "contribute to the business process compliance".

Following the positive results of Alaküla and Matulevičius, we have used our catalog to expand the Portuguese cybersecurity guideline "Quadro Nacional de Referência para a Cibersegurança" (National Cyber Security Reference Board), QNRCS in short, by manually matching the patterns in the developed catalog with the subcategories described in the guideline [6]. In this way, we improved the description of each category by providing patterns that can help satisfy it, helping organizations achieve compliance with the guideline. This association also aims

Table 6. Categories of the QNRCS (Portuguese cybersecurity framework).

Objective	Description
Identify	Helps identify the resources that support important functions and the respective associated risks.
Protect	Focus on the development and implementation of necessary safeguards to guarantee the provision of services or goods, supporting and reinforcing the organization's ability to limit or contain the impact of possible occurrence of a cybersecurity incident.
Detect	Establishes adequate and timely practices to detect the occurrence of cybersecurity events, through the continuous monitoring of information networks and systems and the implementation of detection processes.
Respond	Develops and implements practices that carry out response actions to a cybersecurity incident that has been detected.
Recover	Establishes practices and maintains resilience plans to restore any capacity and/or service that has been compromised following a cybersecurity event.

to help the Portuguese National Cybersecurity Center (Centro Nacional de Cibersegurança – CNCS) develop future certification conformity schemes for its guidelines.

CNCS developed QNRCS as part of the EU push "for a high common level of security of network and information systems across the Union countries" [7]. The QNRCS is based on the ISO/IEC 27000 [14] series and the COBIT framework. It is structured in five key security objectives divided into categories and further divided into subcategories that support the continuous improvement of the cybersecurity of organizations. For some of the subcategory the document provides an example of a technological implementation, a procedural implementation and evidence, consisting of a generic description of how it could be applied; however, these descriptions are small and lack enough information for organizations that want to achieve the objective. The five key objectives and their definitions are shown in Table 6.

Using our association, organizations have access to more information on how to secure their systems and comply with this regulation. For example, if an organization wants to comply with the subcategory "DE.MC-1" which states that information networks and systems must be monitored to detect potential incidents, they must implement an intrusion detection system (IDS) or a firewall. However, the regulation does not provide any information on how to build an IDS or firewall. Using our matching, organizations get access to patterns such as Intrusion Detection Requirements, Abstract IDS, Behavior-Based IDS, Signature-Based IDS, Packet Filter Firewall, Protection Reverse Proxy, Proxy-Based Firewall, Stateful Firewall, and Whitelisting Firewall that not only provide information on how to build an IDS or firewall, but also the different types that exist and what kind of requirements might be needed.

We were able to associate security patterns from our catalog with 86.9% of the existing categories in the QNRCS and for 69.6% of the subcategories. Appendix A shows part of this association. The full association can be found in https://github.com/xextreme22/Security-Patterns. Of all the objectives in the guideline, "Protect" has by far the most unique patterns associated with a total of 94 unique patterns, followed by "Detect" with 32, "Identify" with 14, "Respond" with 8, and "Recover" 1. These numbers show us that security patterns focus mainly on protecting software against cybersecurity threats and neglect other important categories like "Recover" and "Respond" and as such future research should focus on the identification and development of recovery/response oriented security patterns.

4 EVALUATION

Security patterns are difficult to evaluate, since "security is a quality for which there are no numerical measures. It can only be defined relative to another system or by showing that a system satisfies some predefined security

Table 7. Comparison between catalogs.

Catalog	Ours	[24]	[33]	[10]	[21]
Number of Patterns	106	46	35	68	37
Publication year	2022	2006	2006	2013	2021
Uniform description	Yes	Yes	No	Yes	No
Details the process used to build the catalog	Yes	No	No	No	No
The catalog was evaluated	Yes	No	No	No	No

properties" [9]. However, we can evaluate a catalog of patterns using attack patterns, which are the counterparts of security patterns; instead of describing solutions to a potential security problem, they describe attacks [9]. By understanding that for each attack pattern there is at least one security pattern or a combination of patterns in our catalog that mitigates or detects the attack in cases where mitigation is impossible, we can conclude that our catalog is complete for the attacks considered.

Since we needed to ensure that we have a large enough number of attack patterns to account for the majority of attacks, we decided to use an attack library [26]. Attack libraries include most types of attack, meaning that we can have enough confidence in the completeness of our catalog by using them. We used the MITRE ATT&CK library for our evaluation due to its size and popularity.

Using the mitigation strategies present in the attack pattern description, for the 191 attack patterns in the repository, we were able to find at least 1 security pattern in our catalog to mitigate or detect 175 of the attacks. It is important to note that all of the patterns that our catalog does not currently provide a solution either happen outside the scope of the organizations and as such they are either difficult or even impossible to mitigate/detect, or they are better left unmitigated, as mitigation would only make the system less secure.

We do not evaluate the description of patterns in the catalog individually due to our filtration phase described in Section 2.2, which already filters the patterns using qualitative metrics, meaning that only well-described patterns are included in the catalog. We also do not evaluate security patterns individually and whether they actually help designers, engineers, and programmers secure their system because there is already research focusing specifically on this aspect and showing the positive impact of security patterns [9][13].

Table 7, shows a comparison between the existing catalogs we identified with our catalog. We can see that our catalog is the most up-to-date and has the most amount of patterns using a uniform description. It is also the only one that was evaluated.

5 THREATS TO VALIDITY

It is important to state the threats that exist that can invalidate our results.

As mentioned, we used the SMS study by Jafari and Rasoolzadegan as a basis for our study. As this study was published in 2018, considering the fast-changing nature of the field, it is possible that it is outdated. However, during our analysis of studies published between 2018 and 2021, we did not find evidence that this is the case.

Our extraction phase focuses on patterns published in academic venues and does not have a specific search strategy for patterns found in the industry, such as patterns found in software projects. However, as stated in [15], organizations have yet to adopt security patterns as a means of obtaining secure software, and, as such, we do not expect to find a substantial amount of security patterns described outside of academic research.

Since our work was done manually, bias is a potential threat to validity. We minimized bias by making our process as objective as possible with clear questions that do not leave room for interpretation. As our first publication in the field, we also have no bias toward previous studies published by us.

6 CONCLUSION AND FUTURE WORK

Security patterns can help organizations build secure enterprise architectures that comply with regulations without obstructing business needs. However, the literature on security patterns is highly disperse and lacks uniformity, leading organizations to overlook their usage.

As such, we have analyzed 295 scattered security patterns and consolidated them into a concise catalog of 106 patterns using a systematic methodology that includes most patterns developed throughout the years, facilitating their adoption by organizations in this way. The patterns analyzed were filtered using quality metrics that guarantee the quality of each pattern description. Furthermore, the descriptions of all security patterns were converted to a standard template to improve the readability and uniformity of the catalog. The catalog also includes a classification of all security patterns for improved selection and navigation. Finally, by using a clearly defined approach to building the catalog, organizations and researchers can easily change or expand the catalog, depending on their needs.

We have successfully used the catalog to help the Portuguese CNCS improve their reference framework for cybersecurity (QNRCS) by matching all security patterns with the subcategories of the framework, expanding the amount of information organizations have access to, in order to comply with the framework, and helping the center develop certification schemes for their framework.

The catalog was evaluated using the patterns in the MITRE ATT&CK repository. By using this repository, we were able to evaluate the completeness and diversity of patterns in regard to existing attacks by using known attack patterns. Our catalog showed that for all attack patterns in the repository that should be mitigated or detected, it includes at least one security pattern that helps mitigate or detect that attack, proving its completeness and diversity.

For future work for our catalog, we have identified three possible paths. First, a revision of all security pattern descriptions should be made in order to complete missing information, and update information that may have become outdated throughout the years like the examples given or references. Second, the identification and development of new security patterns to further expand the catalog. Finally, the catalog should be updated as new security patterns are published using the same methodology.

In the future an evaluation of our mapping between the catalog and the QNRCS should be made to evaluate its effectiveness on helping organizations achieve compliance with the QNRCS, by analysing previously recorded information of organizations that achieved or tried to achieve compliance with the guideline without the use of security patterns against organizations that used this mapping.

ACKNOWLEDGMENT

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID).

REFERENCES

[1] Mari-Liis Alaküla and Raimundas Matulevičius. 2015. An experience report of improving business process compliance using security risk-oriented patterns. In *IFIP Working Conference on The Practice of Enterprise Modeling*. Springer, 271–285.

[2] Christopher Alexander. 1977. *A pattern language: towns, buildings, construction.* Oxford university press.

[3] Aleem Khalid Alvi and Mohammad Zulkernine. 2012. A comparative study of software security pattern classifications. In *2012 Seventh International Conference on Availability, Reliability and Security*. IEEE, 582–589.

[4] Michaela Bunke. 2014. On the description of software security patterns. In *Proceedings of the 19th European Conference on Pattern Languages of Programs*. 1–10.

[5] Frank Buschmann, Kevlin Henney, and Douglas C Schmidt. 2007. *Pattern-oriented software architecture, on patterns and pattern languages.* John wiley & sons.

[6] Centro Nacional de Cibersegurança 2019. *Quadro Nacional de Referência para a Cibersegurança.* Centro Nacional de Cibersegurança.

[7] European Parliament and of the Council. 2016. Directive (EU) 2016/1148 of the European Parliament and of the Council. http://data.europa.eu/eli/dir/2016/1148/oj.

[8] Eduardo B Fernandez and Rouyi Pan. 2001. A pattern language for security models. In *In Proc. of PLoP*, Vol. 1. Citeseer.

[9] Eduardo B Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Michael VanHilst. 2010. Measuring the level of security introduced by security patterns. In *2010 International Conference on Availability, Reliability and Security*. IEEE, 565–568.

[10] Eduardo Fernandez-Buglioni. 2013. *Security patterns in practice: designing secure architectures using software patterns*. John Wiley & Sons.

[11] Lori Flynn, Jason Clark, Andrew P Moore, Matthew Collins, Eleni Tsamitis, David Mundie, and David McIntire. 2013. Four insider IT sabotage mitigation patterns and an initial effectiveness analysis. In *Proceedings of the 20th Conference on Pattern Languages of Programs*. 1–19.

[12] Erich Gamma, Richard Helm, Ralph Johnson, Ralph E Johnson, John Vlissides, et al. 1995. *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH.

[13] Spyros T Halkidis, Nikolaos Tsantalis, Alexander Chatzigeorgiou, and George Stephanides. 2008. Architectural risk analysis of software systems based on security patterns. *IEEE Transactions on Dependable and Secure Computing* 5, 3 (2008), 129–142.

[14] ISO/IEC 27001:2013 2013. *Information technology — Security techniques — Information security management systems — Requirements*. Standard. International Organization for Standardization, Geneva, CH.

[15] Abbas Javan Jafari and Abbas Rasoolzadegan. 2020. Security patterns: A systematic mapping study. *Journal of Computer Languages* 56 (2020), 100938.

[16] Saluka R Kodituwakku, Peter Bertok, and Liping Zhao. 2001. APLRAC: A Pattern Language for Designing and Implementing Role-Based Access Control.. In *EuroPLoP*, Vol. 1. 2001.

[17] MA Laverdiere, Azzam Mourad, Aiman Hanna, and Mourad Debbabi. 2006. Security design patterns: Survey and evaluation. In *2006 Canadian Conference on Electrical and Computer Engineering*. IEEE, 1605–1608.

[18] Jörg Lenhard, Lothar Fritsch, and Sebastian Herold. 2017. A literature study on privacy patterns research. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 194–201.

[19] Santiago Moral-García, Santiago Moral-Rubio, Eduardo B Fernández, and Eduardo Fernández-Medina. 2014. Enterprise security pattern: A model-driven architecture instance. *Computer Standards & Interfaces* 36, 4 (2014), 748–758.

[20] Julio Moreno, Eduardo B Fernandez, Eduardo Fernandez-Medina, and Manuel A Serrano. 2019. BlockBD: a security pattern to incorporate blockchain in big data ecosystems. In *Proceedings of the 24th European Conference on Pattern Languages of Programs*. 1–8.

[21] Manos Papoutsakis, Konstantinos Fysarakis, George Spanoudakis, Sotiris Ioannidis, and Konstantina Koloutsou. 2021. Towards a collection of security and privacy patterns. *Applied Sciences* 11, 4 (2021), 1396.

[22] Torsten Priebe, Eduardo B Fernández, Jens I Mehlau, and Günther Pernul. 2004. A pattern system for access control. In *Research Directions in Data and Applications Security XVIII*. Springer, 235–249.

[23] Lukas Reinfurt, Uwe Breitenbücher, Michael Falkenthal, Paul Fremantle, and Frank Leymann. 2017. Internet of Things security patterns. In *Proc. PLoP*. 20.

[24] Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann, and Peter Sommerlad. 2006. *Security Patterns: Integrating security and systems engineering*. John Wiley & Sons.

[25] Alireza Shameli-Sendi, Yosr Jarraya, Makan Pourzandi, and Mohamed Cheriet. 2016. Efficient provisioning of security service function chaining using network security defense patterns. *IEEE Transactions on Services Computing* 12, 4 (2016), 534–549.

[26] Adam Shostack. 2014. *Threat modeling: Designing for security*. John Wiley & Sons.

[27] Johanneke Siljee. 2015. Privacy transparency patterns. In *Proceedings of the 20th european conference on pattern languages of programs*. 1–11.

[28] Madiha H Syed, Eduardo B Fernandez, and Paulina Silva. 2017. The secure software container pattern. In *Proceedings of the 23rd Conference on Pattern Languages of Programs (PLoP 2017). Vancouver, Canada.*

[29] Hironori Washizaki, Tian Xia, Natsumi Kamata, Yoshiaki Fukazawa, Hideyuki Kanuka, Takehisa Kato, Masayuki Yoshino, Takao Okubo, Shinpei Ogata, Haruhiko Kaiya, et al. 2021. Systematic Literature Review of Security Pattern Research. *Information* 12, 1 (2021), 36.

[30] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 1–10.

[31] Dereje Yimam and Eduardo B Fernandez. 2016. A survey of compliance issues in cloud computing. *Journal of Internet Services and Applications* 7, 1 (2016), 1–12.

[32] Joseph Yoder and Jeffrey Barcalow. 1997. Architectural patterns for enabling application security. In *Proceedings of the 4th Conference on Patterns Language of Programming (PLoP'97)*, Vol. 2. Citeseer, 30.

[33] Koen Yskout, Thomas Heyman, Riccardo Scandariato, and Wouter Joosen. 2006. A system of security patterns. (2006).

[34] Koen Yskout, Riccardo Scandariato, and Wouter Joosen. 2015. Do security patterns really help designers?. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 292–302.

A QNRCS ASSOCIATION

Table 8 presents part of the association of the pattern in the catalog with some of the subcategories of the "Protect" objective. The full association is available at https://github.com/xextreme22/Security-Patterns.

Table 8. QNRCS association of patterns for the category "PR.GA – Identity Management, Authentication and Access Control".

Subcategory	Patterns
PR.GA-1 – Identity management lifecycle must be defined	Actor and Role Lifecycle, Security Session.
PR.GA-2 – There must be physical access controls to information networks and systems	Access Control to Physical Structures, Relays.
PR.GA-3 – The organization must manage its remote accesses	Third party based authentication, web of trust, Content-Dependent Authorizer, History-Based Authentication, Fine-Grain Access Control, Rights Based Fine Grain Access Control, Whitelisting Firewall, Discretionary Access Control, Mandatory Access Control, Session-Based Attribute-Based Authorization, Risk-Based Authenticator, Role Based Fine Grain Acces Control, Authenticator, Abstract Virtual Private Network, Controlled Access Session, Credential, Role-Based Access Control, Protected Entry Points, Policy-Based Access Control, Multilevel Security, Access Control List, Session-Based Role-Based Access Control, Front Door, Access Control Requirements, Authorization, Check Point, Packet Filter Firewall, Protection Reverse Proxy, Proxy-Based Firewall, Stateful Firewall, Single Access Point.
PR.GA-4 - The organization must apply the principles of least privilege and segregation of duties in access management	Administrator Object, Role-Hierarchies, Actor-based Access Rights, Privilege-Limited Role, Controlled Access Session, Administrator Hierarchy, Capability, Role Rights Definition.
PR.GA-5 – The organization must protect the integrity of its communications networks	Whitlisting firewall, Transport Layer Security, Demilitarized Zone, Packet Filter Firewall, Protection Reverse Proxy, Proxy-Based Firewall, Stateful Firewall.
PR.GA-6 – The organization must verify the identity of employees and link them to their credentials	Credential, Identity Federation, Identity Provider, Front Door, Automated I&A Design Alternatives, Biometrics Design Alternatives, I&A Requirements, Known Partners.
PR.GA-7 – Mechanisms for authentication of users, devices and other information system assets must be defined	Third party based authentication, web of trust, History-Based Authentication, Mutual Authentication, Authenticator, Circle of Trust, Front Door, I&A Requirements.