

AN ONTOLOGICAL APPROACH TO MODELING SOFTWARE WEAKNESSES BASED ON CWE ARCHITECTURAL CONCEPTS

SARTABANOVA ZH.E.^{1*}, DIMITROV V.T.², URDABAYEVA G.ZH.³,
ARULMURUGAN R.⁴

***Sartabanova Zhanar Elibayevna**¹ – PhD, senior lecturer, Heriot Watt university, Aktobe Campus, Aktobe, Kazakhstan.

E-mail: Z.Sartabanova@hw.ac.uk, <https://orcid.org/0000-0003-4942-5117>

Dimitrov Vladimir Tododrov² - PhD, professor, faculty of mathematics and informatics, Sofia university St. Kliment Ohridski, Sofia, Bulgaria

E-mail: cht@fmi.uni-sofia.bg, <https://orcid.org/0000-0002-7441-253X>

Urdabayeva Gulmira Zhadgereevna³ – Master, lecturer, K.Zhubanov Aktobe regional university, Aktobe, Kazakhstan.

E-mail: g.urdabaeva@zhubanov.edu.kz, <https://orcid.org/0000-0002-2723-7783>

Arulmurugan Ramu⁴ – PhD, associate professor, Department of information technology university, Mattu University, Mattu, Ethiopia

E-mail: A.Ramu@hw.ac.uk, <https://orcid.org/0000-0003-1030-9565>

Abstract. The article discusses the task of formalizing knowledge about software weaknesses based on the international Common Weakness Enumeration (CWE) list with a focus on the representation of Architectural Concepts. The relevance of the research is due to the growing number of cyber attacks and the need to take into account security threats not only at the operational stage, but also at the stages of designing and developing software systems. The purpose of the work is to study the system of software weaknesses based on CWE and to develop an ontological knowledge base focused on the needs of software architects. The research uses methods of analysis and synthesis, classification, logical and system analysis, as well as ontological modeling based on OWL, RDF and Protégé environment. As a result, an ontology has been developed in which the central element is the CWEEntries class, associated with the CVEEntries and CAPECEntries classes, which allows us to formalize the relationship between weaknesses, vulnerabilities and attack patterns. The conducted testing using SPARQL queries showed that the developed knowledge base supports semantic search, analysis of relationships between entities and solving applied problems related to the early detection of architectural risks. The scientific novelty of the work lies in the presentation of the architectural representation of CWE as a machine-readable ontological model of knowledge. The practical significance lies in the possibility of using the proposed knowledge base by architects, developers, researchers and teachers in secure-by-design tasks.

Key words: ontology, CWE, CVE, CAPEC, software weaknesses, vulnerabilities, attacks.

Introduction

In the context of the accelerating digitalization of government, commercial and social processes, cybersecurity has become one of the key factors for the sustainable functioning of information systems. The growing dependence of organizations on software platforms, network services, and distributed computing environments is accompanied by an increase in the number of cyber attacks aimed at violating the confidentiality, integrity, and availability of data. In these circumstances, software security becomes particularly important, since it is the errors and weaknesses made at the design and development stages that often become the root cause of subsequent vulnerabilities and incidents.

One of the most widespread international sources of systematic information about software weaknesses is the Common Weakness Enumeration (CWE), maintained by MITRE Corporation. The CWE is a generally accepted list of typical software and hardware weaknesses used by developers, researchers, and cybersecurity professionals as a reference and analytical framework. Of particular interest in this study is the CWE View: Architectural Concepts, which focuses on software architects and organizes weaknesses in architectural security tactics. This approach is important for the implementation of the secure-by-design principle, in which security risks should be taken into account not after the completion of development, but at the early stages of system design. [1]

Despite the widespread use of CWE, the use of this list in practice is associated with a number of difficulties. The organization of software weaknesses remains ambiguous: in different contexts, the same weakness can be interpreted in different ways, classifications are being revised, and

taxonomic relationships between weaknesses, vulnerabilities, and attacks are not always presented in a form convenient for automated analysis. As a result, it is often difficult for specialists responsible for designing and protecting cyber resources to use CWE as a tool for searching, comparing and interpreting knowledge at the architectural level. This problem is especially acute in cases where it is necessary not only to identify a particular weakness, but to trace its connection with potential vulnerabilities, attack scenarios and design solutions.

Modern cybersecurity research demonstrates a steady interest in ontologies, knowledge graphs, and semantic domain modeling. Such approaches make it possible to formalize concepts, their properties and relationships, providing a deeper interpretation of data compared to linear lists and tabular catalogs. At the same time, a significant part of the existing solutions is focused either on the operational level of threat analysis, or on the integration of CVE, CWE and CAPEC repositories without a pronounced emphasis on architectural interpretation of weaknesses. Consequently, there remains a research gap between general knowledge models in cybersecurity and the tasks of a software architect, who needs a tool to analyze weaknesses precisely at the level of design solutions.

In this regard, it seems reasonable to use knowledge engineering methods and artificial intelligence tools to formalize the architectural representation of CWE in the form of an ontological knowledge base. This approach can expand the search, systematization, and analysis of software weaknesses, as well as reduce the impact of taxonomic ambiguity by explicitly defining classes, properties, and semantic relationships.

The object of the research is the CWE software weaknesses system, and the subject is the architectural concept of the CWE weaknesses system. The purpose of the research is to study the system of software weaknesses based on CWE and to develop an ontological knowledge base of this system for software architects. To achieve this goal, the following tasks were set: to analyze the CWE structure; to explore the CWE View: Architectural Concepts; to evaluate the applicability of the OWL and RDF ontological modeling languages, as well as the Protégé environment.; to design a knowledge base based on the CWE architectural concept; to propose use cases for the developed model for architects, software developers and university professors. [2]

The scientific novelty of the study lies in the fact that the architectural representation of CWE is considered not as a static reference list, but as a basis for building a machine-readable ontological knowledge model that supports semantic links between weaknesses, vulnerabilities and attack patterns. The practical significance of the work is determined by the possibility of using the developed knowledge base as a decision support tool in software design, as a means of systematizing knowledge for researchers and as an educational and methodological model for teaching disciplines in the field of software engineering and information security.

The research is limited to the lists of weaknesses maintained by MITRE, as they are the generally accepted public standard in the field. At the same time, the work focuses on the architectural concept of CWE, although the proposed methodology can be adapted to other conceptual representations of weaknesses.

Materials and methods of research

The methodological basis of the research consists of general scientific methods of analysis and synthesis, generalization, abstraction, grouping and classification of data, as well as logical, comparative and systematic analysis. The analysis of taxonomies and taxonomic elements, ontological modeling, synthesis of the knowledge base shell and knowledge search in a semantically structured model were used as special methods. [3-7]

At the first stage of the study, an analysis of the subject area was carried out. The source material was information about software weaknesses from the CWE system, as well as related information about vulnerabilities and attack patterns. The analysis was focused on identifying the key entities of the subject area, determining their semantic relationships, and choosing the architectural logic of knowledge organization. As a result, three basic entities of the future model were identified: CWEEntries as a class of software weaknesses, CVEEntries as a class of vulnerabilities, and CAPECEntries as a class of attacks.

At the second stage, the ontology was designed in Protégé 5.5.0. The OWL language was used

for the formal representation of knowledge, and the RDF and RDFS tools were used to describe the structure and relationships. Ontology was designed as a system of interrelated classes, subclasses, properties, and individuals. Standard OWL constructs were used in the simulation, including owl:Class, rdfs:subClassOf, owl:equivalentClass, and owl:disjointWith. The basic superclass was owl:Thing, which corresponds to the general principles of constructing OWL ontologies. [8; 9]

At the third stage, a hierarchy of classes was formed. The central class of the model was defined by CWEEntries, reflecting the weaknesses of the software in terms of the architectural concept. Within this class, the categories corresponding to the architectural representation of the CWE were identified. Additionally, the CVEEntries and CAPECEntries classes were introduced to link weaknesses to specific vulnerabilities and attack patterns (figure 1). The hierarchy was formed according to the «class — subclass» principle, which ensured the logical structuring of the knowledge base and the possibility of its further expansion.

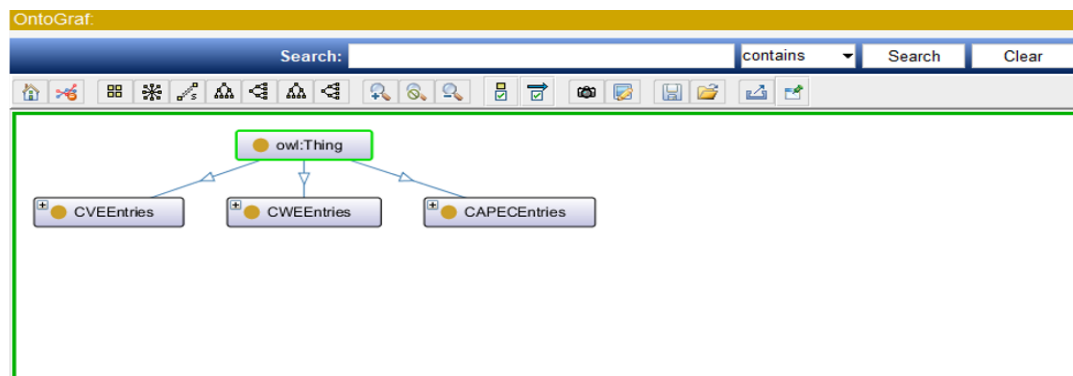


Figure 1. The ontological graph

At the fourth stage, the ontology was filled with instances. Identifiers, descriptions, and semantic characteristics were recorded for each individual (figure 2). Instances of the CWEEntries, CVEEntries, and CAPECEntries classes were filled in, after which relationships were set between them reflecting belonging to the category, the relationship of weakness to vulnerability, and the relationship of weakness to the attack pattern.

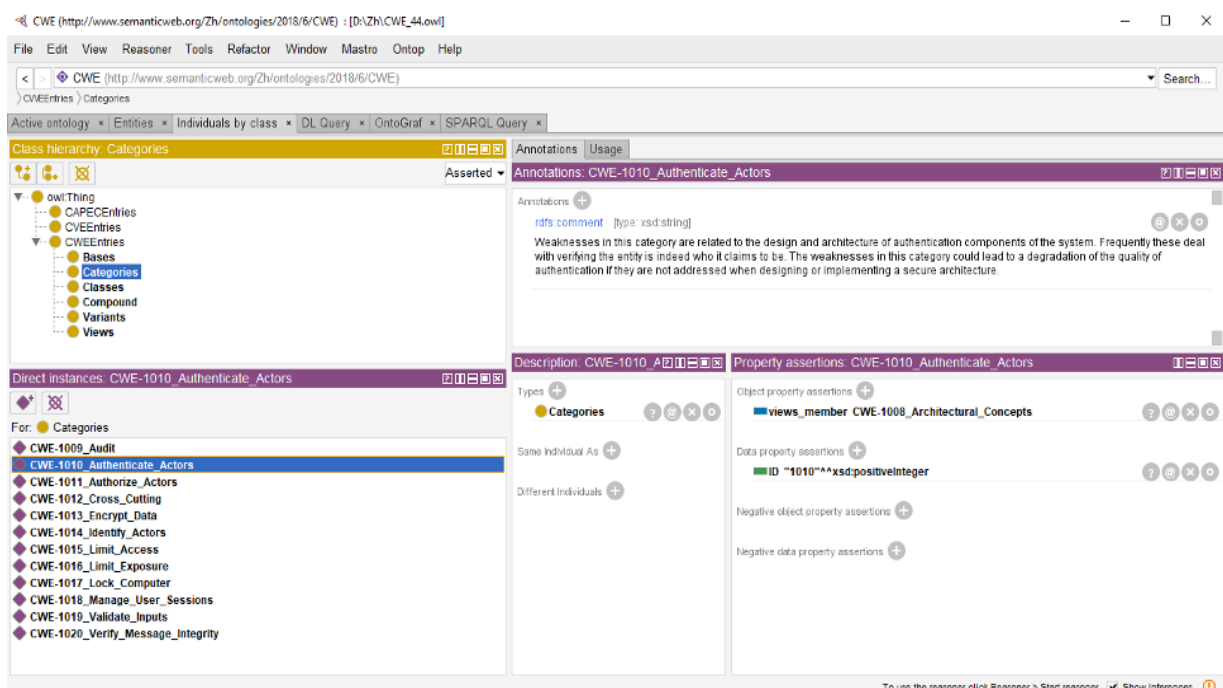


Figure 2. Category structure

At the fifth stage, the object properties and data type properties were defined. The object properties were used to define the relationships between the entities of the ontology. In particular, the references property linked weaknesses to vulnerabilities, related_attack_patterns — weaknesses to attacks, categories_member and views_member — elements with corresponding categories and views (figure 3).

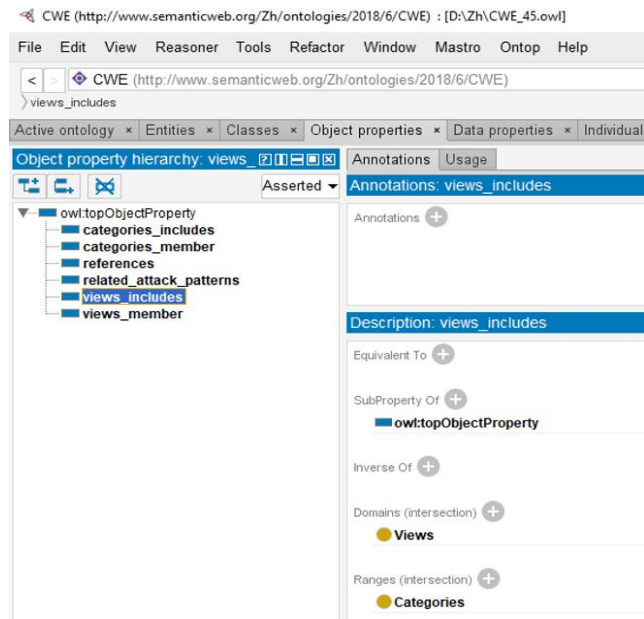


Figure 3. The object properties

A domain and range were set for each property, which ensured the correctness of semantic relationships and limited the scope of their application. The properties of the data type were used to describe the attributes of weaknesses, including the identifier, methods of introduction, possible consequences, likelihood of exploitation, demonstration examples, applicable platforms, order of weakness, and detection methods (Figure 4).

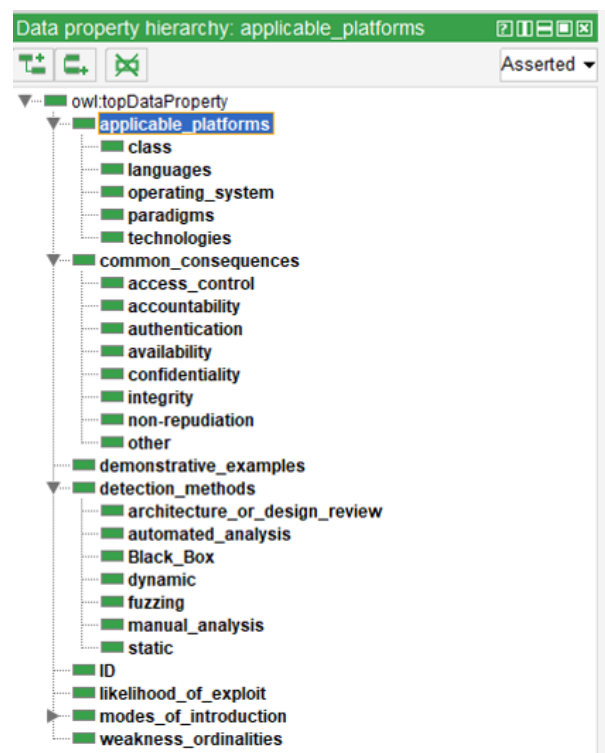


Figure 4. The data properties

At the sixth stage, the model was tested and validated. The correctness of the ontology was tested in Protégé at the level of class structure, hierarchy of properties and relationships between individuals. Additionally, a functional knowledge base check was performed using SPARQL queries to the RDF representation of the model. Several types of queries were used: standard samples, queries based on literal values, queries based on class instances, and queries based on object properties. This allowed us to assess how well the developed model is suitable for real-world knowledge search and analytical processing.

Finally, the applied applicability of the knowledge base was assessed based on a scenario approach. Typical situations from the practice of cybersecurity and architectural analysis were considered, in which it was necessary to identify relevant vulnerabilities, weaknesses, and attacks based on technological or contextual characteristics. This approach allowed us to verify not only the formal correctness of the model, but also its usefulness for the practical work of a software architect.

Results and its discussion

As a result of the research, an ontological knowledge base of software weaknesses was developed, focused on the architectural concept of CWE. The created model combined information about weaknesses, vulnerabilities and attacks in a single semantic space and provided the possibility of their structured presentation, linking and subsequent analysis.

The key result was the formation of an ontological structure in which the CWEEntries class occupies a central place. A hierarchy of categories was implemented within it, reflecting architecturally significant groups of weaknesses. This made it possible to systematize weaknesses not as isolated records, but as elements of a broader architectural model. Such an organization provides a more convenient transition from the general architectural view to specific weaknesses and related objects.

An additional result was the inclusion of the CVEEntries and CAPECEntries classes in the model. This ensured traceability between a software weakness, a corresponding vulnerability, and a possible attack pattern. As a result, the knowledge base began to perform not only a descriptive function, but also an analytical support function, allowing to trace the chain of «architectural weakness — vulnerability — attack».

The essential result of the research was the definition of a set of object and attribute properties. The object properties provided logical connections between the elements of the model, and the properties of the data type made it possible to formalize the descriptive characteristics of weaknesses. Due to this, each entry in the knowledge base has received both a structural position in the ontology and a meaningful description, including the identifier, consequences, likelihood of exploitation, applicable platforms and detection methods. This increased the completeness of the semantic representation of the subject area.

Visualization of the ontology in the form of a graph has shown that the model has connectivity and interpretability. The established relationships between classes and instances have confirmed that the developed structure is suitable for presenting declarative knowledge in the field of software weaknesses and can be used as an information model of a subject area. [10; 11]

The practical effectiveness of the knowledge base was confirmed by the execution of SPARQL queries. The results showed that the model supports data extraction for various analytical scenarios.: search for weaknesses within a given category, obtain the values of individual attributes, identify vulnerabilities related to a particular technology, and determine the relationship between attack patterns, weaknesses, and CVE records. Queries for multiple conditions simultaneously allowed for a more accurate and context-sensitive search, which is especially important for solving the tasks of a software architect.

Testing on applied examples has shown that the developed knowledge base can be used for risk analysis at the design stage. In particular, the model made it possible to perform a thematic vulnerability search by technology, identify attacks related to a given context, and compare them with specific weaknesses and vulnerabilities. This confirmed that the ontology is suitable for supporting the tasks of analyzing architectural solutions and early detection of threats.

Thus, the results of the study confirm that the ontological representation of CWE Architectural

Concepts makes it possible to create a structured, machine-readable and application-useful knowledge base capable of supporting the search, analysis and interpretation of information about software weaknesses at the architectural level.

The results show that the ontological approach is an adequate tool for formalizing knowledge about software weaknesses in the context of architectural design. Unlike the traditional CWE representation in the form of a list or catalog, the developed model defines the semantic structure of the subject area and makes it possible to move from the reference use of data to their analytical interpretation.

One of the most significant conclusions is that the architectural representation of weaknesses can be effectively translated into a machine-readable form without losing semantic connections between entities. The central role of the CWEEntries class and its relation to CVEEntries and CAPECEntries allowed us to reflect not only the classification of weaknesses, but also their practical manifestations. This is especially important for a software architect, since architectural decisions are made when it is necessary to take into account not individual defects, but systems of interrelated risks.

The developed knowledge base confirms the expediency of shifting the focus from after-the-fact vulnerability analysis to an earlier stage - the analysis of architectural weaknesses in the design of the system. This approach corresponds to the modern understanding of security as a property that should be embedded in the system initially. If a weakness is detected at the architectural level, the chances of preventing it are significantly higher, and the cost of fixing it is lower than at the later stages of the software lifecycle.

The results of SPARQL testing have shown that the proposed model provides sufficient flexibility for practical search and analysis. The ability to generate queries based on technological features, attack patterns, or links between weaknesses and vulnerabilities expands the application potential of the system. This means that the developed knowledge base can be considered not only as a research artifact, but also as a component of decision support systems in the field of secure design.

However, the study revealed a number of limitations. Firstly, the completeness of the model directly depends on the quality and completeness of the filling of the ontology. If some of the links between weaknesses, vulnerabilities, and attacks are missing or insufficiently detailed, the analytical capabilities of the knowledge base also decrease. Secondly, the current model is limited by the CWE architectural concept and does not fully cover other representations of weaknesses or additional contexts of their exploitation. Thirdly, the work confirmed the applicability of the model in typical scenarios, but for a final assessment of its effectiveness, broader validation is needed on real projects and in integration with secure SDLC processes.

The educational and research potential of the developed model deserves special attention. For teachers, it can serve as a visual tool to explain the relationship between design errors, vulnerabilities, and attacks. For researchers, the knowledge base is of interest as a basis for further development of cybersecurity ontologies, integration with other repositories, and building intelligent knowledge search systems. For developers and architects, it can serve as a structured reference and analytical tool when choosing safer design solutions.

The prospects for further research are related to the expansion of the ontology by including architectural patterns and antipatterns, protection measures, risk mitigation mechanisms, and additional data sources. An important area is also the integration of the knowledge base with automated software system analysis tools and the construction of intelligent services capable of using the ontological model to identify potential weaknesses in architectural documentation and design artifacts.

In general, the discussion of the results confirms that the proposed approach has both scientific novelty and applied value. It allows us to consider CWE Architectural Concepts as the basis for an intelligent knowledge model and thereby creates conditions for a deeper analysis of the architectural security of software.

Conclusion

The article considers the task of formalizing knowledge about software weaknesses based on the architectural representation of CWE. The relevance of the research is due to the fact that

weaknesses that arise at the design and development stages become the most important sources of subsequent vulnerabilities and attacks, and existing ways of organizing knowledge about them do not always provide convenience of search, interpretation and practical application at the architectural level.

In the course of the research, the set goal was achieved: a system of software weaknesses based on CWE was studied and an ontological knowledge base of this system for software architects was developed. For this purpose, the CWE structure and the representation of Architectural Concepts were analyzed, the choice of OWL, RDF and Protégé environment as tools for ontological modeling was justified, the knowledge base was designed and its applicability was verified using SPARQL queries and application scenarios.

The main result of the work was the development of an ontological model that combines software weaknesses, vulnerabilities, and attack patterns in a single semantic space. The central element of the model is the CWEEntries class, associated with the CVEEntries and CAPECEntries classes, which provides a systematic representation of the domain and supports analytical tracing between architectural solutions, weaknesses, vulnerabilities and attacks.

The results confirmed that the ontological approach is an effective means of systematizing knowledge about software weaknesses and can be used to support secure-by-design. The developed knowledge base allows you to perform semantic search, perform analytical queries and use the results for early detection of architectural risks. The practical significance of the research lies in the fact that the proposed model can be used by software architects, developers, researchers and teachers as a reference, analytical and educational tool.

Thus, the study confirms the prospects of developing ontological knowledge bases based on CWE Architectural Concepts to improve the quality of design solutions and develop intelligent tools to support secure software development.

References

1. MITRE. About CWE [Electronic resource]. – 2024. – URL: <https://cwe.mitre.org/> (date of request: 26.03.2026).
2. MITRE. CWE-1008: Architectural Concepts. CWE View 1008 [Electronic resource]. – 2026. – URL: <https://cwe.mitre.org/data/definitions/1008.html> (date of request: 26.03.2026).
3. Kanakogi K., Washizaki H., Fukazawa Y., Ogata S., Okubo T., Kato T., Kanuka H., Hazeyama A., Yoshioka N. Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques // Information. – 2021. – Vol. 12, No. 8. – Art. 298. – DOI: [10.3390/info12080298](https://doi.org/10.3390/info12080298).
4. Özdemir Sönmez F., Hankin C., Malacaria P. Attack Dynamics: An Automatic Attack Graph Generation Framework Based on System Topology, CAPEC, CWE, and CVE Databases // Computers & Security. – 2022. – Vol. 123. – Art. 102938. – DOI: [10.1016/j.cose.2022.102938](https://doi.org/10.1016/j.cose.2022.102938).
5. Santos J. C. S., Tarrit K., Mirakhorli M. A Catalog of Security Architecture Weaknesses // 2017 IEEE International Conference on Software Architecture Workshops (ICSAW). – 2017. – P. 220–223. – DOI: [10.1109/ICSAW.2017.25](https://doi.org/10.1109/ICSAW.2017.25).
6. Santos J. C. S., Peruma A., Mirakhorli M., Galster M., Veloz Vidal J., Sejfia A. Understanding Software Vulnerabilities Related to Architectural Security Tactics: An Empirical Investigation of Chromium, PHP and Thunderbird // 2017 IEEE International Conference on Software Architecture (ICSA). – 2017. – P. 69–78. – DOI: [10.1109/ICSA.2017.39](https://doi.org/10.1109/ICSA.2017.39).
7. Musen M. A. The Protégé Project: A Look Back and a Look Forward // AI Matters. – 2015. – Vol. 1, No. 4. – P. 4–12. – DOI: [10.1145/2757001.2757003](https://doi.org/10.1145/2757001.2757003).
8. W3C. OWL 2 Web Ontology Language Document Overview [Electronic resource]: W3C Recommendation. – 2012. – URL: <https://www.w3.org/TR/owl2-overview/> (date of request: 26.03.2026).
9. W3C. RDF 1.1 Concepts and Abstract Syntax [Electronic resource]: W3C Recommendation. – 2014. – URL: <https://www.w3.org/TR/rdf11-concepts/> (date of request: 26.03.2026).
10. Sartabanova Zh., Dimitrov V. Modelling of CWEs on the CWE-287 Example // CEUR

Workshop Proceedings. – 2019. – Vol. 2464.

11. Sartabanova Zh., Dimitrov V. T., Sarsimbaeva S. M. Applying the Knowledge Base of CWE Weaknesses in Software Design // Journal of Mathematics, Mechanics and Computer Science. – Al-Farabi Kazakh National University. – 2020. Vol. 108, No. 4. – P. 72–80. – DOI: [10.26577/JMMCS.2020.v108.i4.06](https://doi.org/10.26577/JMMCS.2020.v108.i4.06)

Әдебиеттер тізімі

1. MITRE. About CWE [Electronic resource]. – 2024. – URL: <https://cwe.mitre.org/> (date of request: 26.03.2026).

2. MITRE. CWE-1008: Architectural Concepts. CWE View 1008 [Electronic resource]. – 2026. – URL: <https://cwe.mitre.org/data/definitions/1008.html> (date of request: 26.03.2026).

3. Kanakogi K., Washizaki H., Fukazawa Y., Ogata S., Okubo T., Kato T., Kanuka H., Hazeyama A., Yoshioka N. Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques // Information. – 2021. – Vol. 12, No. 8. – Art. 298. – DOI: [10.3390/info12080298](https://doi.org/10.3390/info12080298).

4. Özdemir Sönmez F., Hankin C., Malacaria P. Attack Dynamics: An Automatic Attack Graph Generation Framework Based on System Topology, CAPEC, CWE, and CVE Databases // Computers & Security. – 2022. – Vol. 123. – Art. 102938. – DOI: [10.1016/j.cose.2022.102938](https://doi.org/10.1016/j.cose.2022.102938).

5. Santos J. C. S., Tarrit K., Mirakhorli M. A Catalog of Security Architecture Weaknesses // 2017 IEEE International Conference on Software Architecture Workshops (ICSAW). – 2017. – P. 220–223. – DOI: [10.1109/ICSAW.2017.25](https://doi.org/10.1109/ICSAW.2017.25).

6. Santos J. C. S., Peruma A., Mirakhorli M., Galster M., Veloz Vidal J., Sejfia A. Understanding Software Vulnerabilities Related to Architectural Security Tactics: An Empirical Investigation of Chromium, PHP and Thunderbird // 2017 IEEE International Conference on Software Architecture (ICSA). – 2017. – P. 69–78. – DOI: [10.1109/ICSA.2017.39](https://doi.org/10.1109/ICSA.2017.39).

7. Musen M. A. The Protégé Project: A Look Back and a Look Forward // AI Matters. – 2015. – Vol. 1, No. 4. – P. 4–12. – DOI: [10.1145/2757001.2757003](https://doi.org/10.1145/2757001.2757003).

8. W3C. OWL 2 Web Ontology Language Document Overview [Electronic resource]: W3C Recommendation. – 2012. – URL: <https://www.w3.org/TR/owl2-overview/> (date of request: 26.03.2026).

9. W3C. RDF 1.1 Concepts and Abstract Syntax [Electronic resource]: W3C Recommendation. – 2014. – URL: <https://www.w3.org/TR/rdf11-concepts/> (date of request: 26.03.2026).

10. Sartabanova Zh., Dimitrov V. Modelling of CWEs on the CWE-287 Example // CEUR Workshop Proceedings. – 2019. – Vol. 2464.

11. Sartabanova Zh., Dimitrov V. T., Sarsimbaeva S. M. Applying the Knowledge Base of CWE Weaknesses in Software Design // Journal of Mathematics, Mechanics and Computer Science. – Al-Farabi Kazakh National University. – 2020. Vol. 108, No. 4. – P. 72–80. – DOI: [10.26577/JMMCS.2020.v108.i4.06](https://doi.org/10.26577/JMMCS.2020.v108.i4.06)

CWE АРХИТЕКТУРАЛЫҚ ТҰЖЫРЫМДАМАЛАРЫНА НЕГІЗДЕЛГЕН БАҒДАРЛАМАЛЫҚ ЖАСАҚТАМАНЫҢ ӘЛСІЗ ЖАҚТАРЫН МОДЕЛЬДЕУГЕ ОНТОЛОГИЯЛЫҚ КӨЗҚАРАС

САРТАБАНОВА Ж.Е.^{1*}, ДИМИТРОВ В.Т.², УРДАБАЕВА Г.Ж.³,
АРУЛМУРУГАН Р.⁴

*Сартабанова Жанар Елибаевна¹ – PhD, аға оқытушы, Хериот-Уатт университеті, Ақтөбе кампусы, Ақтөбе қ., Қазақстан.

E-mail: Z.Sartabanova@hw.ac.uk, <https://orcid.org/0000-0003-4942-5117>

Димитров Владимир Тодоров² – PhD, профессор, Математика және информатика факультеті, Әулие Климент Охридский атындағы София университеті, София қ., Болгария.

E-mail: cht@fmi.uni-sofia.bg, <https://orcid.org/0000-0002-7441-253X>

Урдабаева Гулмира Жадгереевна³ – Магистр, оқытушы, Қ. Жұбанов атындағы Ақтөбе өңірлік университеті,

Ақтөбе қ., Қазақстан.

E-mail: g.urdabaeva@zhubanov.edu.kz, <https://orcid.org/0000-0002-2723-7783>

Арулмуруган Раму⁴ – PhD, қауымдастырылған профессор, Ақпараттық технологиялар кафедрасы, Матту университеті, Матту қ., Эфиопия.

E-mail: A.Ramu@hw.ac.uk, <https://orcid.org/0000-0003-1030-9565>

Андатпа. Мақалада архитектуралық тұжырымдамаларды ұсынуға баса назар аударып, халықаралық жалпы әлсіздіктер (CWE) тізіміне негізделген бағдарламалық жасақтаманың әлсіз жақтары туралы білімді ресімдеу міндеті талқыланады. Зерттеудің өзектілігі кибершабуылдардың көбеюіне және қауіпсіздік қатерлерін тек операциялық кезеңде ғана емес, сонымен қатар бағдарламалық жасақтама сәулетшілерінің қажеттіліктеріне бағытталған онтологиялық білім базасын құру. Зерттеуде талдау және синтездеу, жіктеу, логикалық және жүйелік талдау әдістері, сондай-ақ OWL, RDF және Protege ортасына негізделген онтологиялық модельдеу қолданылады. Нәтижесінде онтология дамыды, онда орталық элемент CWEEntries класы болып табылады, ол sveentries және CAPECEntries кластарымен байланысты, бұл әлсіздіктер, осалдықтар және шабуыл үлгілері арасындағы байланысты ресімдеуге мүмкіндік береді. SPARQL сұрауларын қолдана отырып жүргізілген тестілеу дамыған білім базасы семантикалық іздеуді, субъектілер арасындағы қатынастарды талдауды және архитектуралық тәуекелдерді ерте анықтауға байланысты қолданбалы мәселелерді шешуді қолдайтынын көрсетті. Жұмыстың ғылыми жаңалығы білімнің машинада оқылатын онтологиялық моделі ретінде CWE архитектуралық бейнесін ұсынуға жатыр. Практикалық маңыздылығы сәулетшілердің, әзірлеушілердің, зерттеушілердің және оқытушылардың ұсынылған білім қорын қауіпсіз жобалау тапсырмаларында пайдалану мүмкіндігінде жатыр.

Түйін сөздер: онтология, CWE, CVE, CAPEC, бағдарламалық жасақтаманың әлсіз жақтары, осалдықтары, шабуылдары.

ОНТОЛОГИЧЕСКИЙ ПОДХОД К МОДЕЛИРОВАНИЮ СЛАБЫХ МЕСТ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ОСНОВАННЫЙ НА АРХИТЕКТУРНЫХ КОНЦЕПЦИЯХ CWE

САРТАБАНОВА Ж.Е.^{1*}, ДИМИТРОВ В.Т.², УРДАБАЕВА Г.Ж.³,
АРУЛМУРУГАН Р.⁴

*Сартабанова Жанар Елибаевна¹ – PhD, старший преподаватель, университет Хериот-Уатт, Ақтөбе кампус, г. Ақтөбе, Казахстан

E-mail: Z.Sartabanova@hw.ac.uk, <https://orcid.org/0000-0003-4942-5117>

Димитров Владимир Тодоров² - PhD, профессор, факультет математики и информатики, Софийский университет имени Святого Климента Охридского, г. София, Болгария

E-mail: cht@fmi.uni-sofia.bg, <https://orcid.org/0000-0002-7441-253X>

Урдабаева Гулмира Жадгереевна³ – Магистр, преподаватель, Актюбинский региональный университет имени К.Жубанова, г. Ақтөбе, Казахстан

E-mail: g.urdabaeva@zhubanov.edu.kz, <https://orcid.org/0000-0002-2723-7783>

Арулмуруган Раму⁴ – PhD, ассоциированный профессор, кафедра информационных технологий, Университет Матту, г. Матту, Эфиопия.

E-mail: A.Ramu@hw.ac.uk, <https://orcid.org/0000-0003-1030-9565>

Аннотация. В статье рассматриваются возможности формализации знаний о слабостях программного обеспечения Common Weakness Enumeration (CWE) с точки зрения архитектурных концепций. Актуальность исследования обусловлена растущим числом кибератак и необходимостью учитывать угрозы безопасности не только на этапе эксплуатации, но и на этапах проектирования и разработки программных систем. Целью работы является изучение системы слабых мест программного обеспечения на основе CWE и разработка онтологической базы знаний, ориентированной на потребности разработчиков программного обеспечения. В исследовании используются методы анализа и синтеза, классификации, логического и системного анализа, а также онтологическое моделирование на основе сред OWL, RDF и Protégé. В результате была разработана онтология, в которой центральным элементом является класс CWEEntries, связанный с классами CVEEntries и CAPECEntries, что позволяет нам формализовать взаимосвязь между слабостями, уязвимостями и шаблонами атак. Проведенное тестирование с использованием запросов SPARQL показало, что разработанная база знаний поддерживает семантический поиск, анализ взаимосвязей между сущностями и решение прикладных задач, связанных с ранним выявлением архитектурных рисков. Научная новизна работы заключается в представлении архитектурного представления CWE в виде машиночитаемой онтологической модели знаний. Практическая

значимость заключается в возможности использования предложенной базы знаний архитекторами, разработчиками, исследователями и преподавателями в задачах обеспечения безопасности при проектировании.

Ключевые слова: онтология, CWE, CVE, CAPEC, слабости программного обеспечения, уязвимости, атаки.