

Article

Automated Conversion of CVE Records into an Expert System, Dedicated to Information Security Risk Analysis, Knowledge-Base Rules

Dovydas Benetis, Donatas Vitkus , Justinas Janulevičius, Antanas Čenys  and Nikolaj Goranin * 

Faculty of Fundamental Sciences, Department of Information Systems, Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius, Lithuania; d.vitkus@vilniustech.lt (D.V.); justinas.janulevicius@vilniustech.lt (J.J.); antanas.cenys@vilniustech.lt (A.Č.)

* Correspondence: nikolaj.goranin@vilniustech.lt

Abstract: Expert systems (ESs) can be seen as a perspective method for risk analysis process automation, especially in the case of small- and medium-sized enterprises that lack internal security resources. Expert system practical applicability is limited by the fact that the creation of an expert system knowledge base requires a lot of manual work. External knowledge sources, such as attack trees, web pages, and ontologies, are already proven to be valuable sources for the automated creation of knowledge base rules, thus leading to more effective creation of specialized expert systems. This research proposes a new method of automated conversion of CVE data from the National Vulnerability Database (version CVSS 2) into the knowledge base of an expert system and flags CVE records that have higher risk due to already existing exploit tools. This manuscript also contains a description of the method for implementing software and a practical evaluation of conversion results. The uniqueness of the proposed method is incorporation of the records included in the Cybersecurity and Infrastructure Security Agency (CISA) Known Exploited Vulnerabilities Catalog.

Keywords: information security; risk analysis; expert systems; knowledge base; automation; CVE data



Citation: Benetis, D.; Vitkus, D.; Janulevičius, J.; Čenys, A.; Goranin, N. Automated Conversion of CVE Records into an Expert System, Dedicated to Information Security Risk Analysis, Knowledge-Base Rules. *Electronics* **2024**, *13*, 2642. <https://doi.org/10.3390/electronics13132642>

Academic Editors: Vasilis Katos, Sotiris Ioannidis, George Hatzivasilis and Vasileios Mavroeidis

Received: 29 May 2024

Revised: 19 June 2024

Accepted: 1 July 2024

Published: 5 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the rapidly evolving cyber security threat and vulnerability landscape, there is a need for advanced tools and methodologies to effectively analyze and mitigate these risks just to keep up with the pace of such threats. This threat and vulnerability landscape, however, is represented by continuously updated and managed registers that include the name, nature, behavior, and other important features of such threats. Such registers provide datasets that can be used to achieve the required security-related goals.

The Common Vulnerabilities and Exposures (CVE) registry, along with the scores provided by the Common Vulnerability Scoring System (CVSS) managed by the MITRE Corporation and sponsored by the U.S. Department of Homeland Security (DHS) [1] and Cybersecurity and Infrastructure Security Agency (CISA), provides a knowledge base for IT vulnerabilities with input from representatives of a broad range of industry sectors, ranging from the finance sector to academia. It is the U.S. government's repository of standards-based vulnerability management data [2].

This study proposes an automated method for knowledge base development for expert systems (ESs) dedicated to information security risk analysis by transforming the CVE records into rules that support the knowledge base of a specifically designed expert system. It is achieved through a process for constructing a domain-specific knowledge base, thereby enhancing the precision and efficiency of information security risk analysis.

Moreover, the development of automated methods for knowledge base development in expert systems provides small- and medium-sized businesses with an analysis of potential vulnerabilities, also ensuring that their risk assessment processes are based on the

most up-to-date and relevant information available [3]. This approach to risk analysis is essential in proactively identifying and mitigating potential security threats before they escalate into full-fledged cyber incidents, thereby safeguarding critical assets and data from malicious actors.

2. Background and Related Works

Risk analysis requires assessing the concepts and the interrelationships between threats, vulnerabilities, asset values, security controls, and overall risk. Manually collecting and maintaining such a knowledge base of all actual parameters tends to be overwhelming, especially for smaller organizations, as it requires a lot of resources [4]. To manage and interact with such knowledge bases, expert systems provide a convenient way to facilitate large amounts of information by providing an interactive decision-making engine, as the sets of rules compiled in this way can form blocks of expert information, much larger than that of a human expert [5]. The most suitable type of expert system for information security risk analysis is a classical, rule-based system because the acceptance of the system's conclusions must be justified and requires a cognitive explanation of the conclusions accepted by the system. Also, classic expert systems can be described by a finite number of rules, leaving the possibility to expand, remove, or modify them if necessary. This determines their universality and wide application [6,7]. Forming knowledge can be achieved by using classic, semi-automated, and automated methods [8]. This study focuses on the latter one. This also helps with one of the biggest issues—updating the knowledge base. It is one of the biggest problems when using expert systems, so to solve this problem an automated document processing method, using reliable, recognized, and constantly updated sources, should be used to ensure the reliability and novelty of the knowledge base being maintained [9].

2.1. The Design of the Expert System

Using an Expert System Shell provides a well-maintained platform to improve and update the created ES. An ESS can be any software that provides an ES with a certain knowledge base [10]. In the context of this study, the following ESSs have been analyzed:

- ES-Builder [11]—a robust solution offering some benefits as it is created specifically for educational purposes. The limitations include the inability to be installed and used locally. The shell is built using AJAX framework, and knowledge base facts and rules are stored in MySQL database.
- Pyke v. 1.1 (Python Knowledge Engine) [12]—it uses logic programming that is inspired by the Prolog programming language, but the shell itself is written in Python. Python functions, Pyke rules, and Pyke template variables and graph plans are key features of the Pyke knowledge base. Pyke has an inference engine that applies rules to facts to determine additional facts, forward chaining, and backward chaining. The limitations are that it is rather heavy on resources.
- Drools v. 8.44.0 [13]—is an open-source tool that has a large support community. This shell is based on the Java Rules Engine API (Java Specification Request 94) standard and uses an improved version of the RETE algorithm called ReteOO. Drools is a widely applicable and modern tool for ES development, which makes this shell one of the most popular tools for ES development. The limitations are the heaviness of the resources as well as the slightly outdated technology stack.
- CLIPS v. 6.4.1 (C Language Integrated Production System) [14]—developed in 1985 at NASA's Johnson Space Center and was originally called NAIL (NASA's Artificial Intelligence Language). The syntax of CLIPS is similar to the Lisp programming language. This shell is written in the C programming language, so program extensions can be written in C, and CLIPS itself can be called from other C programs. This led to the popularity of this tool, and since 2005 CLIPS has been one of the most widely used ES development tools. A pretty lightweight solution, though the limitations come when dealing with the easiness to use and the slightly outdated technology stack.

- Jess v. 7.1 (Java Expert System Shell) [15]—widely used to create rule-based ESs. This shell was created in 1995 based on CLIPS, and Jess has all the features that CLIPS has, but unlike CLIPS, Jess is not open source, but it is free for educational purposes. Jess has an ES development environment integrated with the Eclipse platform, which is one of the strongest points for this ESS. It also provides cross-platform functionality. The shortcoming is its heaviness on resources.

For this research, it was decided to use the Jess ES shell. This decision was based on the platform popularity for scientific research, code availability (open-source), cross-platform support, and previously proven compatibility with automatically formed knowledge bases.

2.2. Automated Methods of Forming the Knowledge Base of Expert Systems

Ontologies can be used in the formation of ES knowledge bases for information security risk analysis. For example, the ontology in [16] relates to the ROPE methodology, which is carried out to assess a company's information security, focusing on business processes and risk management. This ontology includes concepts known in the field of information security, such as assets, vulnerabilities, threats, and risk management.

Another ontology suitable for information security risk analysis is presented by [17], which can be used as a tool to identify the level of system vulnerability based on internal user accounts and system configuration. The developed tool is based on a taxonomy that defines the settings of system users and includes different behavioral motives, such as conscious and unconscious activities of the system user.

Some ontologies are related to external information security standards, such as the ontology based on ISO 27002 [18], which can be applied to perform information security analysis. The yet-to-be-created ontology is related to several external information security standards [19], which are comprehensive and have better branching and depth properties in terms of visualization, increasing the coverage of security standards compared to other existing information security ontologies [20]. The ontology of Ramanauskaitė et al. was created to unify security standards and was linked to the following standards: ISO 27001 [21], PCI DSS [22], ISSA 5173 [23], and NISTIR 7621 [24], which are suitable for performing information security risk analysis.

There are several methods for converting ontologies into an ES knowledge base, including the following:

- DAMLJessKB software [25] transforms DAML (DARPA Agent Markup Language) ontologies into Jess expert system rules. This method only uses specific DAML ontologies for conversion, which are converted to Jess expert system rules, being the major drawback of this method.
- DLEJena software [26] is able to convert a pD semantics-compatible OWL 2 RL profile into a Jena expert system knowledge base. Although the developed program does not use the full OWL 2 RL profile, it is able to successfully convert most of the OWL 2 RL ontologies. One of the shortcomings of this method is that this method is limited to the specific Jena expert system.
- The method proposed by [27]. The main idea of this method is to use the existing information security ontologies, converting them into a set of rules of expert systems through the universal RIF (Rule Interchange Format) format, from which it is easy to convert to the format of the set of rules supported by the selected expert system. This method can be widely applied because it solves the shortcomings of the previously described ontology conversion methods, where the rules are converted from a specific ontology language to a specific ES knowledge base.

These methods contribute to information security risk analysis; however, there is yet another issue to be solved—the newness and actuality of the knowledge base. To ensure this feature, it is necessary to constantly monitor the updates of the appropriate information security ontologies. The information security ontologies themselves are not created every day, and there are relatively few of them. Likewise, when a new ontology appears, its suitability for information security risk analysis should be assessed by an

information security expert. The use of ontology conversion methods in the formation of the ES knowledge base for information security risk analysis is a one-time use.

Attack trees have double representations: graphical and textual. Hence, they can be applied to a wide variety of information systems. The use of attack trees in the formation of an expert system knowledge base for the analysis of information security risks enables analyzing not only attack vectors but also the probability, cost, and selection of security measures of possible attacks, which would help analysts make decisions that would reduce or eliminate risks. Therefore, ref. [20] developed a method that automatically converts attack trees into an ES knowledge base. The method is based on converting data from attack trees collected from various sources into ES knowledge base facts. The nodes of the attack trees and their information are converted to facts, and the root of the attack tree is used to create an attack description. The converted attack tree is saved in the form of CLIPS rules. Each generated CLP file is formed as an ES knowledge base, which can later be imported into an expert system or combined with an existing ES knowledge base.

The analysis of the existing automated methods of forming the knowledge base of expert systems has revealed that there are no methods that would describe the conversion of data provided by CVE (Common Vulnerabilities and Exposures) into the knowledge base of expert systems, and this cannot be carried out with the existing methods. This supports the idea that the inclusion of the CVE data source in the knowledge base of expert systems for information security risk assessment would prove to be useful, as CVE records include known software vulnerabilities with the name and version of the vulnerable software, and with the inclusion of the Common Vulnerability Scoring System (CVSS), vulnerability score calculation information security risk analysis of system data could assess the risks posed by the used software to the confidentiality, integrity, and availability of information. This was also confirmed by a study conducted by the Ponemon Institute, in which almost 3000 information security professionals from various countries were surveyed in 2018–2019, and it was found that 60% of information security breaches are related to the software used, known vulnerabilities (CVEs), and that 62% of organizations were unaware that their information was vulnerable to vulnerabilities in the software they were using. Since existing automated methods cannot convert CVE data into a formable ES knowledge base for information security risk analysis, the development of a new method is required.

2.3. Existing Data Transformation Models for CVE Data

Researchers have previously tried to align and facilitate the CVE data using the MITRE ATT&CK Framework. This framework describes malicious behaviors and provides mitigation strategies for each reported attack pattern. The authors introduced a dataset of 1813 CVEs mapped with MITRE ATT&CK techniques and proposed models to automatically link a CVE to one or more techniques based on the text description from the CVE metadata. They achieved it through machine learning and pre-trained BERT-based language models while counteracting the highly imbalanced training set with data augmentation strategies based on the *TextAttack* framework. This model aims to find kill chain scenarios inside complex infrastructures and enable the prioritization of CVE patching by the threat level [28].

Another source aims to predict the severity of the CVE from the vulnerability description using deep learning. It proposes a novel approach for predicting the severity of vulnerabilities based on their CVE description using natural language models. The model was validated with a test dataset of 7765 CVEs, yielding an accuracy of 84.2% [29].

3. The CVE Data

CVE records are administered by a US non-profit organization, MITRE. They publicly provide access to the CVE database, which can be downloaded freely or directly searched on the website itself. MITRE administers CVE records, assigning them unique numbers and providing the basic information of the registered vulnerability: CVE number, status, description, information source, phase, score, and comments. This way, the reported

vulnerability can be linked to other security tools and services. This resource does not provide information such as potential risk, impact, or more detailed technical vulnerability information that would be used to assess the risk or impact of a vulnerability and perform an information security risk analysis.

The National Vulnerability Database (NVD) is another source from which CVE data can be freely downloaded in JSON format or via an API. This source provides CVE data with additional information such as vulnerable software or hardware versions, CVSS Vulnerability Scoring System information, and other useful attributes that can be used to assess the security risks of the software in use, making it useful information for information security purposes and risk analysis. The CVE information provided by NVD is expanded and more detailed; therefore, the data provided by it are selected as a source for the automated knowledge base formation of an expert system for information security risk analysis.

3.1. CVE Basic Data

The CVE data provided by the NVD are presented in the JSON schema. After analyzing the CVE data provided by NVD, the main data that will be used for the automated conversion to the ES knowledge base were selected. Data that are useful for information security risk analysis were selected (see Table 1).

Table 1. Selected basic CVE data.

Selected Data	Data Type	Description of the Option	Obligation
id	String	This field specifies the unique identification number of the CVE.	Yes
publishedDate	String	The date of publication of the vulnerability is indicated, and this information is relevant to assess the date from which the risk posed by the vulnerability is relevant. It also helps to assess whether it is a new or long-known vulnerability that could have already caused one or another damage.	No
lastModifiedDate	String	This field indicates the date the vulnerability information was updated. This is relevant when monitoring the change in vulnerability information; based on the changed date, it is possible to initiate the update of vulnerability information in the ES knowledge base.	No
vulnerable	Boolean	This field indicates whether the specified PU version is vulnerable. Possible values: false or true. This field can be used to include only those versions of the PI that are vulnerable to the ES knowledge base.	Yes
cpe23Uri	String	This field indicates the firmware version using the CPE scheme. CPE is a structured naming scheme for information technology systems, software, and packages. According to CPE, the expert system will be able to determine whether the used PI has known vulnerabilities.	Yes
description_data.value	String	This field describes vulnerability itself. This information is useful for obtaining more information about the vulnerability.	No

Key CVE data were analyzed and selected to be used in the automated formation of the ES knowledge base. Among the selected data in Table 1, there are no specified selected vulnerability scoring system (CVSS) data. NVD uses several versions of the CVSS assessment systems. To evaluate and choose which version and which data to use in the formation of the knowledge base, studies of the CVSS versions used by NVD and their data were conducted.

3.2. The CVSS Data

NVD uses three versions of the CVSS scoring system for CVE records: 2.0, 3.0, and 3.1. Comparing the main differences between the versions of CVSS estimation calculation systems found that CVSS versions 3.0 and 3.1 use more criteria to calculate the vulnerability score. The CVSS 3 scoring system is much more comprehensive and accurate than the CVSS

version 2 scoring system, making CVSS version 3 more useful for information security risk analysis [30].

After analyzing all CVE records submitted to NVD until 16 June 2021 17:36, it was found that not all CVE records provided by NVD have CVSS estimates. This study analyzed 164,921 records, of which 155,007 records had CVSS estimates. Also, estimates used by CVSS version 3 were found to have the fewest CVE records, and all records with CVSS 3 estimates also have CVSS version 2 estimates.

Although CVSS version 3 uses more criteria to evaluate vulnerabilities and can therefore be more accurate in calculating vulnerability estimates, CVSS version 2 covers almost twice as many records as CVSS version 3. For the ES to make unified decisions for information security risk analysis, it is necessary to choose unified data, based on which the ES will make relevant decisions. Therefore, CVSS version 2 and the data provided by it when forming the ES knowledge base were chosen to be used.

3.3. CISA Known Exploited Vulnerabilities Catalog

CISA has found that vulnerabilities with a low CVSS score can cause just as much damage to information security because a chain of vulnerabilities can be exploited during an attack. Also, a vulnerability may be rated with the highest CVSS score, but its exploitation may be very difficult and unexploitable. Therefore, for these reasons, CISA started developing a catalog of exploits of known vulnerabilities from 3 November 2021.

CISA updates this directory to include additional exploitable vulnerabilities as they become known and when they meet the following conditions:

- The vulnerability has a CVE ID assigned to it.
- There is solid evidence that the vulnerability has been actively exploited in the public domain.
- There is an obvious remedy for the vulnerability, such as a software update from the manufacturer.

Incorporating information from CISA's Known Exploited Vulnerabilities Catalog into the evolving knowledge base of expert systems for information security risk analysis is beneficial. Such vulnerabilities that are included in this directory have a higher probability of being exploited and harming information security [31].

From the data provided by the CISA Known Exploited Vulnerabilities Catalog, the CVE ID and the date of the vulnerability were chosen to be used. The CVE ID will be used to mark entries that are included in this directory, with the date as additional information about the newness of the inclusion.

4. The Proposed Method

A new method, converting the CVE data from NVD with CVSS version 2.0 to the knowledge base of the expert system and marking those CVE entries that are included in the CISA Known Exploited Vulnerabilities Catalog, is proposed. A diagram of the method is presented in Figure 1.

The method steps, as presented in Figure 1, are detailed as follows:

- Metadata are imported from the NVD. Since NVD provides data metadata (SHA256 hash sums), these data are downloaded and saved to compare with already downloaded data and to avoid re-sending the same data files.
- Is such file already downloaded? Before starting the CVE data download, the metadata from the previously downloaded CVE data are checked against the newly downloaded metadata from the NVD. If the metadata matches—the CVE data file is not sent, and if the metadata does not match—the process of downloading CVE data is initiated. In this way, data download time is saved.
- Data file is downloaded and saved. There are two ways to download vulnerability data from NVD: by downloading archives (in GZ or ZIP formats) containing CVE data in JSON format or by using the application programming interface (API). Both methods have advantages and disadvantages. It was decided to use a standard data

download from NVD, which sends archives containing files in JSON format. This way, all CVE data are downloaded faster, and local requests are not tracked.

- Data are extracted and read. In this process, the downloaded archives are extracted, and the CVE data are obtained.
- Does the CVE entry have a CVSS V2.0 score? This process picks only those CVE records that contain available CVSS version 2.0 scores. If the record does not have it, it is not included in the forming ES knowledge base.
- Does the CVE specify the software? Only those CVE records that contain software specifications are included.
- Data from CISA are downloaded and imported. To mark in the conversion process those records that are included in the CISA Known Exploited Vulnerabilities Catalog, this directory is downloaded from the CISA website in CSV format, and the data in it are loaded.
- CVE data are converted into Jess rule database. This process converts CVE data with CVSS version 2 estimate data into Jess rules. This process also marks those CVE records that are included in the CISA Known Exploited Vulnerabilities Catalog and additionally extracts vulnerable software information from the CPE data contained in CVEs: manufacturer, name, and version.

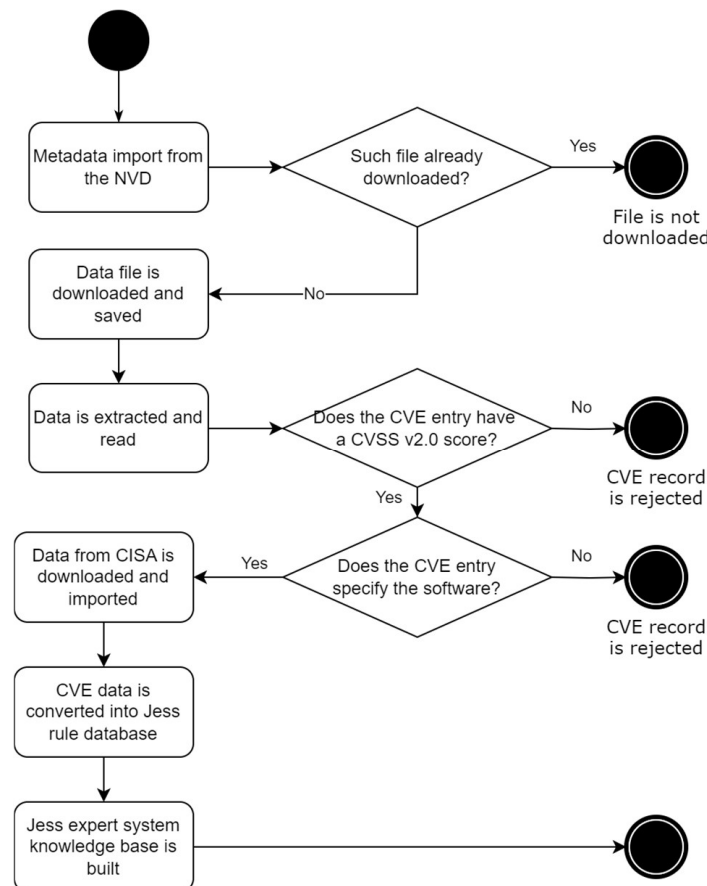


Figure 1. A method for converting CVE data into the ES knowledge base.

During the process, only selected CVE data, which are relevant for information security risk analysis, are converted. An example of the selected CVE data conversion is presented in Figure 2, where the data used to build the knowledge base are highlighted in red.

Jess ES's knowledge base is made up of a list of facts, known as working memory. In this conversion process, the relevant CVE record data are restructured into the Jess ES fact structure; thus, the fact list is converted into the Jess ES knowledge base. Facts in Jess can be of three types:

- Unsorted facts—they are like rows in a relational database table, where table columns correspond to named data fields, which are called slots. When writing an unsorted fact, slots can be specified in any order. Unsorted facts are the most used type of facts and a good choice in most situations.
- Sorted facts—they do not have the structure of named fields, they are just a short, flat list. Such facts are convenient for simple pieces of information that do not require structure.
- Shadow facts—they are unsorted facts that are associated with Java objects in the real world—they provide the ability to reason about events that are occurring outside Jess ES.

```
{ "resultsPerPage":1,"startIndex":0,"totalResults":1,"result":{"CVE_data_type":"CVE","CVE_data_format":"MITRE","CVE_data_version":"4.0","CVE_data_timestamp":"2021-06-17T15:45Z","CVE_items":[{"cve":{"data_type":"CVE","data_format":"MITRE","data_version":"4.0","CVE_data_meta":{"ID":"'CVE-2021-26416'"},"ASSIGNER':"secure@microsoft.com"},"problemtype":{"problemtype_data":{"description":["lang":"en","value":"'NVD-CWE-Vulnerability'"]}}},"configurations":{"CVE_data_version":"4.0","nodes":[{"operator":"OR","children":[{"cpe_match":{"vulnerable":true,"cpe23Uri":"'cpe:2.3:o:microsoft:windows_10:20h2:*:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_10:1607:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_10:1809:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_10:1909:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_10:2004:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_server_2016:-:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_server_2016:20h2:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_server_2016:1909:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_server_2016:2004:*:*:*:*','cpe_name':[]},{'vulnerable':true,'cpe23Uri':'cpe:2.3:o:microsoft:windows_server_2019:-:*:*:*:*','cpe_name':[ ]}]}}},"impact":{"baseMetricV3":{"cvssV3":{"version":"3.1","vectorString":"'CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:N/I:N/A:H','attackVector':'NETWORK','attackComplexity':'LOW','privilegesRequired':'LOW','userInteraction':'NONE','scope':'CHANGED','confidentialityImpact':'NONE','integrityImpact':'NONE','availabilityImpact':'HIGH','baseScore':7.7,'baseSeverity':'HIGH'},'exploitabilityScore':3.1,'impactScore':4.0},'baseMetricV2':{'version':'2.0','vectorString":"'AV:N/AC:L/AU:N/C/N:I/N:A/C",'accessVector':'NETWORK','accessComplexity':'LOW','authentication':'NONE','confidentialityImpact':'NONE','integrityImpact':'NONE','availabilityImpact':'COMPLETE','baseScore':7.8},'severity':'HIGH','exploitabilityScore':10.0,'impactscore':6.9,'acInsufInfo':false,'obtainAllPrivilege':false,'obtainUserPrivilege':false,'obtainOtherPrivilege':false,'userInteractionRequired':false}},'publishedDate':'2021-04-13T20:15Z','lastModifiedDate':'2021-04-16T20:09Z'}}]}
```

Figure 2. Example of a CVE record.

Unsorted facts are general purpose and widely applicable, sorted facts are useful for working with small pieces of information, and shadow facts are used to allow the ES to respond to things happening outside the ES. For converting CVE data to Jess ES facts, the most appropriate fact type is unsorted facts. So, the output of the conversion process looks like the example shown in Figure 3.

- Jess expert system knowledge base is being built. This process creates and saves the end result, a file containing the CVE data facts that make up the Jess ES knowledge base.

All facts in Jess ES are created using the *deftemplate* template, which defines the fields that an input fact can have. For rule-based systems, a *deftemplate* is like a database schema that defines the way the system views the data it uses. Therefore, before entering converted CVE facts into Jess ES for the first time, it is necessary to map the fields used. The fields used by the converted CVE facts and their description are given in Figure 4.


```
(MAIN::vulnerability (id CVE-2021-26416) (publisheddate 2021-04-13T20:15Z)
(lastmodifieddate 2021-04-16T20:09Z) (description "Windows Hyper-V Denial of Service
Vulnerability") (cpe23uri cpe:2.3:o:microsoft:windows_10:1607:*:*:*:*:*:*
cpe:2.3:o:microsoft:windows_server_2016-*:*:*:*:*:*
cpe:2.3:o:microsoft:windows_server_2019-*:*:*:*:*:*
cpe:2.3:o:microsoft:windows_10:1809:*:*:*:*:*
cpe:2.3:o:microsoft:windows_server_2016:1909:*:*:*:*:*
cpe:2.3:o:microsoft:windows_10:1909:*:*:*:*:*
cpe:2.3:o:microsoft:windows_10:2004:*:*:*:*:*
cpe:2.3:o:microsoft:windows_server_2016:2004:*:*:*:*:*
cpe:2.3:o:microsoft:windows_10:20h2:*:*:*:*:*
cpe:2.3:o:microsoft:windows_server_2016:20h2:*:*:*:*:*)) (vectorstring
AV:N/AC:L/Au:N/C:N/I:N/A:C) (accessvector NETWORK) (accesscomplexity LOW)
(authentication NONE) (confidentialityimpact NONE) (integrityimpact NONE)
(availabilityimpact COMPLETE) (basescore 7.8) (severity HIGH) (exploitabilityscore
10.0) (impactscore 6.9) (known_exploited NO) (date_added_to_known_exploited NO)
(affected_products "Microsoft Windows 10 1607" "Microsoft Windows server 2016 -"
"Microsoft Windows server 2019 -" "Microsoft Windows 10 1809" "Microsoft Windows
server 2016 1909" "Microsoft Windows 10 1909" "Microsoft Windows 10 2004" "Microsoft
Windows server 2016 2004" "Microsoft Windows 10 20h2" "Microsoft Windows server 2016
20h2"))
```

Figure 3. Example of a converted fact.

```
(deftemplate vulnerability
  (slot id
    (type string))
  (slot publisheddate
    (type string))
  (slot lastmodifieddate
    (type string))
  (slot description
    (type string))
  (multislot cpe23uri)
  (slot vectorstring
    (type string))
  (slot accessvector
    (type string))
  (slot accesscomplexity
    (type string))
  (slot authentication
    (type string))
  (slot confidentialityimpact
    (type string))
  (slot integrityimpact
    (type string))
  (slot availabilityimpact
    (type string))
  (slot basescore
    (type float))
  (slot severity
    (type string))
  (slot exploitabilityscore
    (type float))
  (slot impactscore
    (type float))
  (slot known_exploited
    (type string))
  (slot date_added_to_known_exploited
    (type string))
  (multislot affected_products))
```

Figure 4. Description of converted CVE data Jess ES.

4.1. Program Prototype of the Method Converting CVE Data into the ES Knowledge Base

Based on the proposed method, a program prototype that implements the idea of the developed method was created using the Python programming language. It was chosen due to its wide range of use, compatibility with various operating systems, and the fact that it is free.

When the prototype of the created program is launched, all actions of the created method are performed automatically—no user intervention is required. The program prototype initially performs a check of the metadata of locally existing CVE files and

the metadata of new CVE files downloaded from NVD. The verification is performed by downloading the metadata from the NVD and comparing it with the already locally existing metadata stored in the `nvd_cache.json` file created by the application. This file stores metadata file names and SHA256 hash sums of CVE data files, and these data are compared with the newly downloaded metadata to determine whether the CVE data file has already been downloaded or not. If it is determined that such a CVE data file has already been downloaded, the file is not sent again, and if it was not, the file is downloaded. This saves data download time if the CVE data file has not been updated in NVD. The prototype of the application displays the progress of this process to the user during the metadata verification and file upload function. Downloaded CVE data files are in ZIP format archives, which are placed in the created “nvd” directory.

After downloading the CVE data files from NVD, the extracted function is initiated. During this activity, the CVE data in JSON format are extracted from the downloaded archive. The archived JSON files are placed in the created “data” directory.

After the file extraction function is completed, the function of reading JSON files and converting the CVE data contained in them to the Jess ES knowledge base is initiated. During this phase, only those CVE records that have CVSS V2 estimates with the specified vulnerable software version (CPE value) are sampled when building the Jess ES knowledge base. That is, if the scanned CVE record does not have a CVSS V2 estimate, it is rejected, and if a CVE record has a CVSS V2 estimate but does not have specified CPE fields, the record is also rejected and not included in the forming ES knowledge base. During data conversion, data from CISA’s Known Exploited Vulnerabilities Catalog are also downloaded and loaded, with which it is checked whether the converted CVE record is included in this catalog; if it is—the CVE record is marked accordingly. Even during the conversion of CVE data, vulnerable software information is extracted from CPE data contained in CVE: manufacturer, name, and version.

After the program prototype completes the conversion function, information about the total number of CVE records read, the total number of CVE records converted to the ES knowledge base, and the total number of CVE records rejected due to the specified conditions are displayed to the user. Also, for the convenience of the user, a Jess ES data template is provided, which the user can copy and use for data description before importing the automatically generated CVE data facts into the ES knowledge base.

The file “cve_jess_kb.dat” created by the program prototype is an automatically formed ES knowledge base, which contains information about known software vulnerabilities (CVEs) and their known exploitation in public space (see Figure 5).

4.2. Prototype Performance Evaluation

Three tests were conducted with the developed prototype of the program, during which the accuracy and performance of the program were tested. To evaluate the accuracy of the program prototype, the amount of CVE records read by the prototype and the amount of CVE records converted to ES knowledge were checked, and these amounts were compared with the raw CVE data from NVD. Raw CVE data were analyzed by uploading them to Elastic Stack and the number of CVEs being checked against the conditions raised. To evaluate the performance of the prototype, CVE was measured in data conversion times, including data downloads.

During the tests, the actual number of CVE entries in the NVD was checked, which is compared with the CVE entries read by the developed program prototype. Next, the actual target number of converted CVE records according to the set conditions was checked against the actual number of converted data of the program prototype. During the tests, the conversion time of the prototype CVE records to the Jess ES knowledge base was measured, including downloading data from the CVE database provided by the NVD. The results of all tests performed on the program prototype are summarized in Table 2.


```

1 (MAIN::vulnerability (id CVE-1999-0001) (publisheddate 1999-12-30T05:00Z) (lastmodifieddate 2010-12-16T05:00Z) (description "ip_input.c in BSD-
derived TCP/IP implementations allows remote attackers to cause a denial of service crash or hang via crafted packets.") (cpe23uri cpe:
2.3:o:freebsd:freebsd:1.0:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:1.1.5.1:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.1.7:*:*:*:*:* cpe:
2.3:o:freebsd:freebsd:2.2:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.2.8:*:*:*:*:* cpe:2.3:o:openbsd:openbsd:2.3:*:*:*:*:* cpe:
2.3:o:bsd:bsd_os:3.1:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.2.3:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.2.4:*:*:*:*:* cpe:
2.3:o:freebsd:freebsd:2.2.5:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.2.6:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.0:*:*:*:*:* cpe:
2.3:o:freebsd:freebsd:2.0.5:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.1.5:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.1.6:*:*:*:*:* cpe:
2.3:o:freebsd:freebsd:2.2.2:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.0.1:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:1.1:*:*:*:*:* cpe:
2.3:o:freebsd:freebsd:1.2:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.1.6.1:*:*:*:*:* cpe:2.3:o:freebsd:freebsd:2.1.7.1:*:*:*:*:* cpe:
2.3:o:freebsd:freebsd:3.0:*:*:*:*:* cpe:2.3:o:openbsd:openbsd:2.4:*:*:*:**) (vectorstring AV:N/AC:L/Au:N/C:N/I:N/A:P) (accessvector
NETWORK) (accesscomplexity LOW) (authentication NONE) (confidentialityimpact NONE) (integrityimpact NONE) (availabilityimpact PARTIAL) (basescore
5.0) (severity MEDIUM) (exploitabilityscore 10.0) (impactscore 2.9) (known_exploited NO) (date_added_to_known_exploited NO) (affected_products
"Freebsd Freebsd 1.0" "Freebsd Freebsd 1.1.5.1" "Freebsd Freebsd 2.1.7" "Freebsd Freebsd 2.2" "Freebsd Freebsd 2.2.8" "Openbsd Openbsd 2.3" "Bsd
Bsd_os 3.1" "Freebsd Freebsd 2.2.3" "Freebsd Freebsd 2.2.4" "Freebsd Freebsd 2.2.5" "Freebsd Freebsd 2.2.6" "Freebsd Freebsd 2.0" "Freebsd
Freebsd 2.0.5" "Freebsd Freebsd 2.1.5" "Freebsd Freebsd 2.1.6" "Freebsd Freebsd 2.2.2" "Freebsd Freebsd 2.0.1" "Freebsd Freebsd 1.1" "Freebsd
Freebsd 1.2" "Freebsd Freebsd 2.1.6.1" "Freebsd Freebsd 2.1.7.1" "Freebsd Freebsd 3.0" "Openbsd Openbsd 2.4"))
2 (MAIN::vulnerability (id CVE-1999-0002) (publisheddate 1998-10-12T04:00Z) (lastmodifieddate 2009-01-26T05:00Z) (description "Buffer overflow in
NFS mountd gives root access to remote attackers, mostly in Linux systems.") (cpe23uri cpe:2.3:o:caldera:openlinux:1.2:*:*:*:*:* cpe:
2.3:o:redhat:linux:2.1:*:*:*:*:* cpe:2.3:o:bsd:bsd_os:1.1:*:*:*:*:* cpe:2.3:o:redhat:linux:4.0:*:*:*:*:* cpe:2.3:o:redhat:linux:
4.1:*:*:*:*:* cpe:2.3:o:redhat:linux:4.2:*:*:*:*:* cpe:2.3:o:redhat:linux:5.0:*:*:*:*:* cpe:2.3:o:redhat:linux:2.0:*:*:*:*:* cpe:
2.3:o:redhat:linux:3.0.3:*:*:*:*:* cpe:2.3:o:redhat:linux:5.1:*:*:*:**) (vectorstring AV:N/AC:L/Au:N/C:I:C/A:C) (accessvector NETWORK)
(accesscomplexity LOW) (authentication NONE) (confidentialityimpact COMPLETE) (integrityimpact COMPLETE) (availabilityimpact COMPLETE) (basescore
10.0) (severity HIGH) (exploitabilityscore 10.0) (impactscore 10.0) (known_exploited NO) (date_added_to_known_exploited NO) (affected_products
"Caldera Openlinux 1.2" "Redhat Linux 2.1" "Bsd Bsd_os 1.1" "Redhat Linux 4.0" "Redhat Linux 4.1" "Redhat Linux 4.2" "Redhat Linux 5.0" "Redhat
Linux 2.0" "Redhat Linux 3.0.3" "Redhat Linux 5.1"))
3 (MAIN::vulnerability (id CVE-1999-0003) (publisheddate 1998-04-01T05:00Z) (lastmodifieddate 2018-10-30T16:26Z) (description "Execute commands as
root via buffer overflow in Tooltalk database server rpc.ttdbserverd.") (cpe23uri cpe:2.3:o:sgi:irix:6.0:*:*:*:*:* cpe:2.3:o:sgi:irix:
6.1:*:*:*:*:* cpe:2.3:o:sgi:irix:6.2:*:*:*:*:* cpe:2.3:o:sgi:irix:6.3:*:*:*:*:* cpe:2.3:o:sgi:irix:5.2:*:*:*:*:* cpe:
2.3:a:tritreal:ted_cde:4.3:*:*:*:*:* cpe:2.3:o:sgi:irix:5.3:*:*:*:*:* cpe:2.3:o:sgi:irix:6.4:*:*:*:*:* cpe:2.3:o:hp:hp-ux:
10.03:*:*:*:*:* cpe:2.3:o:hp:hp-ux:11.00:*:*:*:*:* cpe:2.3:o:ibm:aix:4.1:*:*:*:*:* cpe:2.3:o:ibm:aix:4.1.1:*:*:*:*:* cpe:
2.3:o:sun:sunos:5.4:*:*:*:*:* cpe:2.3:o:sun:sunos:5.5:*:*:*:*:* cpe:2.3:o:sun:sunos:5.5.1:*:*:*:*:* cpe:2.3:o:sun:solaris:
2.6:*:*:*:*:* cpe:2.3:o:ibm:aix:4.2:*:*:*:*:* cpe:2.3:o:ibm:aix:4.2.1:*:*:*:*:* cpe:2.3:o:ibm:aix:4.3:*:*:*:*:* cpe:
2.3:o:sun:sunos:4.1.3:*:*:*:*:* cpe:2.3:o:sun:sunos:--:*:*:*:*:* cpe:2.3:o:hp:hp-ux:10.02:*:*:*:*:* cpe:2.3:o:ibm:aix:
4.1.2:*:*:*:*:* cpe:2.3:o:ibm:aix:4.1.4:*:*:*:*:* cpe:2.3:o:sun:sunos:5.0:*:*:*:*:* cpe:2.3:o:sun:sunos:5.2:*:*:*:*:* cpe:
2.3:o:hp:hp-ux:10.01:*:*:*:*:* cpe:2.3:o:ibm:aix:4.1.3:*:*:*:*:* cpe:2.3:o:ibm:aix:4.1.5:*:*:*:*:* cpe:2.3:o:sun:sunos:
5.1:*:*:*:*:* cpe:2.3:o:sun:sunos:5.3:*:*:*:**) (vectorstring AV:N/AC:L/Au:N/C:I:C/A:C) (accessvector NETWORK) (accesscomplexity LOW)
(authentication NONE) (confidentialityimpact COMPLETE) (integrityimpact COMPLETE) (availabilityimpact COMPLETE) (basescore 10.0) (severity HIGH)
(exploitabilityscore 10.0) (impactscore 10.0) (known_exploited NO) (date_added_to_known_exploited NO) (affected_products "Sgi Irix 6.0" "Sgi Irix
6.1" "Sgi Irix 6.2" "Sgi Irix 6.3" "Sgi Irix 5.2" "Tritreal Ted_cde 4.3" "Sgi Irix 5.3" "Sgi Irix 6.4" "Hp Hp-ux 10.03" "Hp Hp-ux 11.00" "Ibm Aix
4.1" "Ibm Aix 4.1.1" "Sun Sunos 5.4" "Sun Sunos 5.5" "Sun Sunos 5.5.1" "Sun Solaris 2.6" "Ibm Aix 4.2" "Ibm Aix 4.2.1" "Ibm Aix 4.3" "Sun Sunos
4.1.3" "Sun Sunos --" "Hp Hp-ux 10.02" "Ibm Aix 4.1.2" "Ibm Aix 4.1.4" "Sun Sunos 5.0" "Sun Sunos 5.2" "Hp Hp-ux 10.01" "Ibm Aix 4.1.3" "Ibm Aix
4.1.5" "Sun Sunos 5.1" "Sun Sunos 5.3"))
4 (MAIN::vulnerability (id CVE-1999-0004) (publisheddate 1997-12-16T05:00Z) (lastmodifieddate 2018-10-12T21:29Z) (description "MIME buffer overflow

```

Figure 5. Prototype in action: automated conversion of CVE records into a knowledge base.

Table 2. Test results of the developed prototype.

	Test No. 1	Test No. 2	Test No. 3
The actual number of CVE entries	186,741	186,851	187,614
The number of CVE entries read by the prototype	186,741	186,851	187,614
Reading data as a percentage	100%	100%	100%
Target number of converted posts	175,255	175,378	176,362
Number of CVE records converted	175,255	175,378	176,362
Conversion of data into percentages	100%	100%	100%
Number of rejected CVE entries	11,486	11,473	11,252
Conversion time with download from NVD	6 min 36 s.	6 min 42 s.	6 min 58 s.

It is observed that the developed prototype works correctly—it successfully reads all CVE records and converts all CVE records according to the set conditions. Also, the prototype program works efficiently because it converts CVE records directly from NVD to Jess ES knowledge.

For further investigation, three tests of knowledge import quality were performed to evaluate the correctness of the data. During the tests, an attempt was made to import the automatically generated knowledge base into Jess and check the number of converted facts with the number of actual imported facts. Also, during the tests, the time of importing the automatically generated knowledge into Jess was measured. The results of the tests performed with the automatically formed ES knowledge base are summarized in Table 3.

The experiments have revealed that all the automatically generated knowledge is successfully imported and read in Jess ES—the converted data show 100% correctness.

Table 3. Attempts to import data into the ES.

	Test No. 1	Test No. 2	Test No. 3
Number of CVE entries converted by prototype	175,255	175,255	175,378
Number of facts successfully imported into Jess ES	175,255	175,255	175,378
Import as a percentage	100%	100%	100%
Time to import data into Jess ES	19 s	20 s	20 s

5. Conclusions

The research into the existing automated methods for building expert system knowledge bases in the field of information security risk management has revealed the potential to convert CVE data into the knowledge base of expert systems. By supplementing the knowledge base of the expert system with CVE data, it is possible to assess the risks posed by the software used for information security.

There are several fields of application used by other researchers, focusing on mapping the CVE vulnerability descriptions to certain security frameworks and facilitating natural language processing models to automate certain data transformations, but our approach focuses more on the data transformation of CVE data to expert system rules.

An analysis of CVE sources has revealed the NVD's data source suitability for the development of the knowledge base from CVE data. Analysis of the CVE data provided by NVD found that not all CVE records have CVSS estimates. Also, it was found that CVSS version 2 has nearly twice as many records as CVSS version 3 and that CVSS version 3 records also have CVSS version 2 estimates. An additional source of CVE data for the emerging knowledge base of expert systems was selected—the CISA Known Exploited Vulnerabilities Catalog, which provides additional information about the importance of the vulnerability and the probability of its exploitation. After analyzing the CVE sources and the data they provide, a new automated method is proposed, which automatically converts CVE data from NVD with CVSS version 2 data into the knowledge base of the expert system and marks those CVE records that are included in the CISA Known Exploited Vulnerabilities Catalog.

A program prototype has been created in the Python programming language that implements the proposed method idea. After the experiments, it was found that the prototype efficiently and successfully transforms 100% of selected CVE data, and the formed database includes more than 175 thousand records about vulnerabilities.

The use of the CVE data-converting method for the formation of the knowledge base of expert systems for information security risk analysis is superior to other existing methods in that this method uses continuously updated sources, thus ensuring the actuality of the knowledge base of the expert system without additional user effort, and with the knowledge base formed automatically by this method, it is possible to assess the risk posed by the software used for information security.

Author Contributions: Conceptualization, D.V. and N.G.; methodology, D.V. and A.Č.; software, D.B.; validation, D.B. and D.V.; formal analysis, N.G.; investigation, D.B.; data curation, D.B.; writing—original draft preparation, J.J. and D.B.; writing—review and editing, N.G. and A.Č.; supervision, D.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available in a publicly accessible repository. Available online: <https://github.com/dvitkus/CVE2JESS/> (accessed on 27 May 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Kühn, P.; Relke, D.N.; Reuter, C. Common vulnerability scoring system prediction based on open source intelligence information sources. *Comput. Secur.* **2023**, *131*, 103286. [CrossRef]
- Dawson, M.; Bacias, R.; Gouveia, L.B.; Vassilakos, A. Understanding the challenge of cybersecurity in critical infrastructure sectors. *Land Forces Acad. Rev.* **2021**, *26*, 69–75. [CrossRef]
- Hernandez, Z.; Hernandez, T.H.; Velasco-Bermeo, N.; Monroy, B. An expert system to detect risk levels in small and medium enterprises (SMEs). In Proceedings of the Fourteenth Mexican International Conference on Artificial Intelligence (MICAI), Cuernavaca, Mexico, 25–31 October 2015.
- Lee, Y.; Woo, S.; Song, Y.; Lee, J.; Lee, D.H. Practical vulnerability-information-sharing architecture for automotive security-risk analysis. *IEEE Access* **2020**, *8*, 120009–120018. [CrossRef]
- Azzazi, A.; Shkoukani, M. A Knowledge-based Expert System for Supporting Security in Software Engineering Projects. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 395–400. [CrossRef]
- Atymtayeva, L.; Kozhakhmet, K.; Bortsova, G. Building a knowledge base for expert system in information security. *Adv. Intel. Syst. Comput.* **2014**, *270*, 57–76. [CrossRef]
- Tripathi, K.P. A review on knowledge-based expert system: Concept and architecture. *IJCA Spec. Issue Artif. Intell. Tech. -Nov. Approaches Pract. Appl.* **2011**, *4*, 19–23.
- Colson, A.R.; Cooke, R.M. Expert elicitation: Using the classical model to validate experts' judgments. *Rev. Environ. Econ. Policy* **2018**, *12*, 113–132. [CrossRef]
- Tecuci, G.; Marcu, D.; Boicu, M.; Schum, D.A. *Knowledge Engineering: Building Cognitive Assistants for Evidence-Based Reasoning*; Cambridge University Press: Cambridge, UK, 2016.
- Ogu, E.C.; Adekunle, Y.A. Basic Concepts of Expert System Shells and an Efficient Model for Knowledge Acquisition. *Int. J. Sci. Res.* **2013**, *2*, 554–559.
- McGoo Software. ES-Builder Web Expert System Shell. Available online: <http://www.mcgoo.com.au> (accessed on 15 January 2024).
- Frederiksen, B. Applying Expert System Technology to Code Reuse with Pyke. In Proceedings of the PyCon, Birmingham, UK, 12–14 September 2008.
- Wen, Q. Drools Rules Engine Used in Management Accounting System Design Research. In Proceedings of the 4th International Conference on Management Science and Engineering Management (ICMSEM 2023), Nanchang, China, 2–4 June 2023.
- Riley, G. *Adventures in Rule-Based Programming: A CLIPS Tutorial*; Secret Society Software, LLC: AZ, USA, 2022.
- Yurin, A.Y.; Dorodnykh, N.O. Personal knowledge base designer: Software for expert systems prototyping. *SoftwareX* **2020**, *11*, 100411. [CrossRef]
- Orbst, L.; Chase, P.; Markeloff, R. Developing an Ontology of the Cyber Security Domain. In Proceedings of the Seventh International Conference on Semantic Technologies for Intelligence, Defense, and Security, Fairfax, VA, USA, 23–26 October 2012; pp. 49–56.
- Sicilia, M.A.; Garcia-Barriocanal, E.; Bermejo-Higuera, J.; Sanchez-Alonso, S. What are information security ontologies useful for? *Commun. Comput. Inf. Sci.* **2015**, *544*, 51–61.
- Fenz, S.; Plieschnegger, S.; Hobel, H. Mapping information security standard ISO 27002 to an ontological structure. *Inf. Comput. Secur.* **2016**, *24*, 452–473. [CrossRef]
- Ramanauskaitė, S.; Olifer, D.; Goranin, N.; Čenys, A. Security ontology for adaptive mapping of security standards. *Int. J. Comput. Commun. Control* **2013**, *8*, 878. [CrossRef]
- Vitkus, D.; Salter, J.; Goranin, N.; Čeponis, D. Method for attack tree data transformation and import into risk analysis expert systems. *Appl. Sci.* **2020**, *10*, 8423. [CrossRef]
- ISO/IEC 27001:2005; Information Technology—Security Techniques—Information Security Management Systems—Requirements. International Organization for Standardization: Geneva, Switzerland, 2005.
- PCI DSS 3.2.1; Payment Card Industry Data Security Standard. PCI Security Standards Council: Wakefield, MA, USA, 2018.
- ISSA 5173; The Security Standard for SMEs. 2|SEC: London, UK, 2012.
- NISTIR 7621; Small Business Information Security. The National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016.
- Kopena, J.; Regli, W.C. DAMLJessKB: A Tool for Reasoning with the Semantic Web. *IEEE Intell. Syst.* **2003**, *18*, 74–77. [CrossRef]
- Meditkos, G.; Bassiliades, N. DLEJena: A practical forward-chaining OWL 2 RL reasoner combining Jena and Pellet. *J. Web Semant.* **2010**, *8*, 89–94. [CrossRef]
- Vitkus, D.; Steckevičius, Ž.; Goranin, N.; Kalibatiene, D.; Čenys, A. Automated expert system knowledge base development method for information security risk analysis. *Int. J. Comput. Commun. Control* **2019**, *14*, 743–758. [CrossRef]
- Grigorescu, O.; Nica, A.; Dascalu, M.; Rughinis, R. CVE2ATT&CK: BERT-Based Mapping of CVEs to MITRE ATT&CK Techniques. *Algorithms* **2022**, *15*, 314. [CrossRef]
- Manjunatha, A.; Kota, K.; Babu, A.S. CVE Severity Prediction From Vulnerability Description—A Deep Learning Approach. *Procedia Comput. Sci.* **2024**, *235*, 3105–3117. [CrossRef]

30. Dodiya, B.; Singh, U.K.; Gupta, V. Trend analysis of the CVE classes across CVSS metrics. *Int. J. Comput. Appl.* **2021**, *183*, 23–30. [\[CrossRef\]](#)
31. Czarnowski, I. A framework for the clustering and categorization of CISA reports. *Procedia Comput. Sci.* **2022**, *207*, 4369–4377. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.