Systematic Review

Exploring the Architectural Composition of Cyber Ranges: A Systematic Review

Dionysios Stamatopoulos, Menelaos Katsantonis, Panagiotis Fouliras and Ioannis Mavridis *

Department of Applied Informatics, University of Macedonia, 546 36 Thessaloniki, Greece; diostam@uom.edu.gr (D.S.); mkatsantonis@uom.edu.gr (M.K.); pfoul@uom.edu.gr (P.F.)
* Correspondence: mavridis@uom.edu.gr

Abstract: In light of the ever-increasing complexity of cyber–physical systems (CPSs) and information technology networking systems (ITNs), cyber ranges (CRs) have emerged as a promising solution by providing theoretical and practical cybersecurity knowledge for participants’ skill improvement toward a safe work environment. This research adds to the extant respective literature, exploring the architectural composition of CRs. It aims to improve the understanding of their design and how they are deployed, expanding skill levels in constructing better CRs. Our research follows the PRISMA methodology guidelines for transparency, which includes a search flow of articles based on specific criteria and quality valuation of selected articles. To extract valuable research datasets, we identify keyword co-occurrences that selected articles are concentrated on. In the context of literature evidence, we identify key attributes and trends, providing details of CRs concerning their architectural composition and underlying infrastructure, along with today’s challenges and future research directions. A total of 102 research articles’ qualitative analyses reveal a lack of adequate architecture examination when CR elements and services interoperate with other CR elements and services participating, leading to gaps that increase the administration burden. We posit that the results of this study can be leveraged as a baseline for future enhancements toward the development of CRs.

Keywords: cyber ranges; systematic review; cyber–physical systems; information technology networking systems

1. Introduction

Cyber-attacks on the CPS and ITN infrastructure domains and their intersection points occur daily, many of which are on a large scale and at an alarming rate. Cisco predicts that technology companies and governments will join forces to develop artificial intelligence (AI) systems, ensuring a safer online environment with continuous training for employees to handle risks competently [1]. The advent of advanced computers, cyber–physical systems, and intelligent co-existing heterogeneous networks with extensive communication capabilities yields new vulnerabilities due to high dependency on cyber information and the dynamic nurture of the existing attacks. These advanced capabilities make the assets’ confidentiality, integrity, and availability (the well-known CIA triad [2]) attractive attack targets. Significant efforts are being made to combat threats and protect an enterprise’s critical assets, such as unified CPSs and networks, along with their data and services. These efforts include using CR platforms as a theoretical and practical necessity to extend the participants’ skills. The reason is that cyber-knowledge is best learned by practice on hyper-realistic education and training platforms because practicing on production systems is not considered appropriate [3]. CRs provide cyber activities and live exercises in settings similar to real-world systems and realistic true-to-life scenarios [4–13]. CRs (also referred to as CR environments or in silico platforms where only experimentally generated information is used [14]) are platforms where participants (i.e., students, professionals, etc.) may prepare,
improve their cybersecurity posture (i.e., primary, intermediate, professional, etc.), and hone their knowledge (i.e., security, analytics, etc.).

Most studies aim to enhance many elements (the term element is used in computing to refer to a smaller part of a larger system [15]) and services of CRs [16] to increase automation (e.g., scenario automation), efficient infrastructure deployment, functionality, and resource performance. Current CRs are equipped with many of these features. However, according to the European Cyber Security Organization (ECSO), no single CR unit should simultaneously support all of them [16]. ECSO mentions that functions not natively supported by CRs themselves should be carefully evaluated to address the interoperability and compatibility challenges that would arise (e.g., when an update is needed from an external source affecting them). This is due to their complexity, the different technologies needed to interoperate, the services they support, and the deployment methods and techniques they include [17–20]. To reduce complexity, the authors in [16,17] propose an architecture with a four-layered structure comprising core, infrastructure, services, and front-end technologies. In [20], a six-layer architecture is proposed, in which each layer relies on distinct technologies such as (i) underlying infrastructure, (ii) virtualization, (iii) containerization, (iv) orchestration, (v) configuration management, and (vi) target infrastructure, thereby improving the entire process of element deployment and administration. The authors in [5,21] support CR segmentation into logical parts that could help provide advanced functionality.

After analyzing many platforms, the author of [22] suggested using open-source, public tools and technologies for their development and administration. Specifically, for cloud implementations, the author of [23] proposes using RESTful services to minimize the complexity by using specific software such as APIs (Advanced Programming Interfaces) for communication between the different layers capable of managing, monitoring, and controlling various segments. In [24], the authors present a model that incorporates emulation, simulation, data fabrication, and serious gaming tools and is composed of three architectural elements: (i) Sphynx’s Security Assurance Platform, (ii) CTTP models and program editor, and (iii) CTTP Models and Programmes adaptation tool.

The necessity of CRs is demonstrated by the need for well-educated users and confirmed by conducted research. Although the extant literature provides a wide range of solutions, additional research is needed in fields that need to be thoroughly explored. More specifically, we identified only a few publications mentioning the architectural composition of CRs), which play an essential part in this research. This argument is also supported by [19] claiming that it is time to initiate the conversation on sharing the architectural designs and implementation specifications.

Current studies overlook the examination of architectural composition, with most of the proposed solutions being limited, discussed very abstractly, or deploying their selected underlying infrastructure without providing details on the interoperability of the different elements and services. This means that it may be inefficient to follow the procedures adopted for each CR, considering the technological complexity of their structure, and it may be challenging to estimate their behavior sufficiently. Besides this accuracy deficit, CRs often require specific structures used by non-experts [25] who may be unable to assert whether a solution better fits their needs, even if a detailed analysis is provided. Moreover, these deployments need substantial time and physical and technical effort, while financial investments are usually unpredictable [26].

This work explores the architectural environment of CRs, considering their role in theoretical and practical knowledge of cybersecurity. This study reviews the existing articles and open-access data sources related to the architectural composition and underlying infrastructure. To the best of our knowledge, it is the first literature review concentrating on architectural composition based on qualitative analysis of the recent literature. It supports the cyber community and stakeholders by identifying all available open-access sources. It reduces researchers’ search time for suitable information and datasets while providing an up-to-date review of the peer-reviewed literature. Finally, this work highlights the lack of
datasets (i) concerning specific attributes of the CRs composition, capable of achieving a well-composed architecture, and (ii) datasets the community can consult for comparing CRs [19]. Both types of datasets seem equally essential and should be examined.

The identified open issues concerning the composition of CRs can support the cyber community in its efforts to minimize the high levels of uncertainty for errors and control [5] and to establish boundaries between different modules or hierarchy layers, containing potential fault propagation while avoiding platform collapse. All these factors facilitate troubleshooting and problem isolation, thus leading to better administration and management.

The main goal of this work is to explore the architectural composition and underlying infrastructure of CRs, highlighting three major research objectives by examining three *whats* [research questions (RQs)], respectively (which will be discussed in Section 4).

**RQ1:** *What* are the attributes of CR architecture? This RQ examines and analyzes the existing literature, particularly focusing on the architectural attributes of CRs, identifying key elements and services;

**RQ2:** *What* are today’s challenges? This RQ identifies and analyzes the challenges in the literature findings that suffer existing CRs’ implementations;

**RQ3:** *What* needs to be improved? This RQ establishes a comprehensive understanding of the architectural composition of CR platforms to inform future development and research.

This paper’s primary significance and contribution, which differentiates it from previous studies, can be summarized in three main points. First, to the best of our knowledge, this is the first study summarizing all available publications regarding CR architecture in the context of a literature review; it presents all appropriate datasets so that the community can establish a comprehensive knowledge of the architectural composition provided by the researchers. Second, it demonstrates all the dataset collection and the tools we used for our analysis procedures, enabling the information’s comparability. Third, in the context of the evidence of the review results, it discusses the architectural composition of CRs by analyzing selected articles.

The remainder of this paper is organized as follows. Section 2 presents the literature survey’s methodology and preliminary results. Section 3 discusses the results of the research. Section 4 discusses the three RQs in the context of the literature research evidence. In Section 5, a short conclusion is provided, along with our future study goals.

**2. Methodology**

The overall research methodology adopted in this literature review is the new Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework statement [27], as updated in 2020. The methodology was considered appropriate as it is the most suitable because of its extensive data collection and relatively rigorous and detailed process and because it can support mixed-method systematic reviews, including quantitative and qualitative studies. It focuses on the development and performance improvement of an advanced evidence-reporting process for screening and selecting relevant publications. The methodology, which consists of three phases, includes an expanded checklist of keyword items segmented into sections that researchers can follow and match in their research.

Beginning from the start of this paper, the structure of items contained in our search, namely (i) title, (ii) introduction, (iii) inclusion process and eligibility criteria, (iv) information sources, (v) search strategy, (vi) selection process, (vii) data items, (viii) study selection (ix) synthesis of the results, and (x) discussion, is located on the respective items of PRISMA. A visual presentation of the flow is given in Figure 1.
2.1. Inclusion Process and Eligibility Criteria

This work aims to identify, collect, and present usable datasets based on specific criteria regarding the composition of CRs. Initially, an article fulfills the criterion if it is recognizable that it has contributed to our datasets, but as a final criterion, only articles that use the term *cyber ranges* meet the inclusion criteria. A new search was deemed necessary during the article study due to missing outcomes during the initial search. To explore possible causes of heterogeneity among eligible search outcomes, an expert panel of three people working together finally decided whether they should be used.

In the context of evidence, we follow a structured report of the available outcomes using tables and visual display diagrams. Note that during the search process, the datasets extracted from the study were preconfigured, produced, and saved in a file. To synthesize and visualize the presented outcomes, we uploaded the collected datasets into the last version of open-access software [28,29] designated for this purpose to conduct the meta-analysis. In our study, we produced two separate files, one for each process. Keeping resulting datasets open for sensitivity analyses on contested aspects, a detailed explanation of the content of the two files and how the datasets are utilized will be found in Section 2.4.

2.2. Information Sources

To cover a wide range of articles in the literature, a set of well-known databases (DBs) were queried to collect relevant datasets. Due to the spread of articles across multiple databases, the literature search was carried out using the following seven, namely ACM [30], Google Scholar [31], IEEE Xplore [32], MDPI [33], ScienceDirect [34], Semantic Scholar [35], and Springer Link [36]. However, articles relevant to our research outside those seven DBs were considered, demonstrating extensive research in the CR field.
2.3. Search Strategy

To ensure that our search was efficient and reproducible, the following keyword Boolean expression (1) was applied to the seven DBs listed previously:

\[
\text{"Cyber Ranges" AND "cybersecurity" OR "survey" OR "platform"} \quad (1)
\]

The filters applied for CRs span 15 years, i.e., from 2000 to 2024, including journal and conference articles and proceeding papers written in English. The articles were searched for relevant data, including the title of the article, the name of the authors, the year, and the name of the DB, which were collected and extracted from the selected studies for the study selection process. It was also determined whether those datasets were access-free, controlled, available for purchase, or unavailable, and domain characteristics of the datasets were identified. The search yielded a number \(n = 312\) of preliminary articles utilizing the seven DBs mentioned previously. In addition, our extensive study led us to carry out additional searching via other sources \[37–39\] using combinations of the above terms to include any relevant papers left out from the initial search.

2.4. Selection Process

After the initial search process (Identification phase), two more phases were involved: the Screening phase and the Included phase. The aim was to determine whether an article was relevant to this literature review.

During the Identification phase, the Rayyan software \[28\], a cloud-based bibliometric analysis web tool, helped us conduct the meta-analysis by maintaining reviews, speeding up this process, and selecting studies. We uploaded a preconfigured, produced, and saved file \([.txt]\) with 123-kilobytes data capacity on the tool’s main page during our previous search of the seven DBs mentioned in Section 2.2.

Removing the duplicate records (13 articles), the necessary adjustments (10 articles) were made concerning the abstract part of the articles, and 289 articles remained. In the Screening phase, each article’s abstract was thoroughly examined for relevance to determine its potential contribution to our research, filtering out articles outside our research scope (i.e., not meeting the inclusion criteria of relevancy). Specifically, the inclusion and exclusion criteria shown in Table 1 were used for literature selection. After applying the criteria, this phase reduced the reports to \(n = 131\).

<table>
<thead>
<tr>
<th>Inclusion Criteria (IC)</th>
<th>Exclusion Criteria (EC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1 Articles related to the general capabilities and functionality of CRs</td>
<td>EC1 Articles generally focus only on the cybersecurity domain</td>
</tr>
<tr>
<td>IC2 Articles particularly related to CRs in CPS and ITN domains of interest</td>
<td></td>
</tr>
</tbody>
</table>

Next, only reports using the term cyber ranges met the inclusion eligibility during the Screening phase, excluding 35 reports. This phase narrowed the selected reports to \(n = 96\). The new search that was made, as it was deemed necessary for our study’s completeness, increased the number of reports by 6, culminating in the Included phase, with a total corpus of \(n = 102\) reports.

During the Included phase, we used the VOSviewer tool v.1.6.2 \[29\], which is an open-access software that finds hot keyword co-occurrences of the high-frequency words among the researchers’ preferences for constructing and analyzing bibliometric networks by creating keyword co-occurrence map that can produce quality analysis, co-citation analysis, and keyword analysis. In this phase, a 14-kilobyte text file (.txt) with the full name title of the 102 articles was used. The title of an article defines keywords that are generally the enrichment and refinement of the article’s core idea.
Next, three expert people conducted a quality appraisal of those 102 articles, and the discrepancies were discussed and resolved. An article fulfilled the criterion if the information in its full-text review contains useful datasets, concentrating on those articles that included detailed information on the architectural composition and the underlying infrastructure of CRs. This process led to a number of articles, with $n = 11$.

2.5. Data Items

During our study, various datasets compatible with the CRs’ architectural composition were identified and searched, and they were then analyzed to identify common themes, patterns, and gaps in the literature that need further research. The identified data items extracted in those 11 articles are summarized in 4 sectors (labeled A, B, C, and D, respectively) in Table 2. The third column presents the numbers and ratios of the respective articles.

**Table 2. Sector focus of articles in existing CR deployments.**

<table>
<thead>
<tr>
<th>Sector Label</th>
<th>Dataset Description</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Refers to datasets concerning a wide range of CR issues such as automation, scenario mechanisms, scoring systems, orchestration mechanisms, performance, infrastructure provisioning and programmability, resource allocation and efficiency, hardware and software tools, runtime and safe execution, and type of exercise</td>
<td>6/11 (~55%)</td>
</tr>
<tr>
<td>B</td>
<td>Refers to datasets based on the methodology used, the deployment methods, and mechanisms when they need to be conducted and executed</td>
<td>3/11 (~27%)</td>
</tr>
<tr>
<td>C</td>
<td>Refers to datasets on standardized models that simplify the process of design methods</td>
<td>1/11 (~9%)</td>
</tr>
<tr>
<td>D</td>
<td>Refers to datasets on the federation deployment, which can offer users a one-stop shop for all their needs</td>
<td>1/11 (~9%)</td>
</tr>
</tbody>
</table>

3. Results

3.1. Study Selection

Figure 1 depicts the results of the search and selection process, segmented into three phases, from the number of records identified in the search to the number of studies included and excluded in the review. For the search, the keyword Boolean expression (I) was applied to the seven DBs listed in Section 2.3. During the final phase (Included), no other reports were excluded.

3.2. Synthesis of Results

In this subsection, we present the outcomes of all the sensitive analyses previously conducted to assess the robustness of the results.

Figure 2 screens the overall distribution of those $n = 289$ articles (records screened of the Screening phase) per annum and the trend that is constantly increasing until today (blue dashed line). With few exceptions, we can see the interest in CRs increasing steadily; however, the number dropped in 2015 for a year, and after 2016, the number increased again. In 2021, the number of publications slightly dropped for two consecutive years.

After the Screening phase, the review identified 102 articles (Included phase) relating to approaches proposed to address the CRs’ composition and their underlying infrastructure. During the full-text review of those 102 articles, we identified that the researchers addressed various solutions conserving the CRs’ architecture as they did the following:

- Shared similar logic and interpreted this logic in the same manner [20,40];
- Were manipulated differently in different platforms (for instance, techniques about infrastructure deployments; domain of application; scenario mechanism and scoring system; and teams, objectives, geolocation, etc.) [17,22,41];
- Used various design developments covering a wide range of technologies, methods, functionality, and heterogeneous objectives, each within a specific domain of expertise (e.g., CPSs and ITNs) [16].

**Figure 2.** Number and trend of relevant publications per annum (records screened in Figure 1).

In the end, researchers proposed numerous approaches, highlighting the importance of their solutions and aiming to improve many of these points; however, each researcher has different research preferences and finally highlights various aspects of them in their study. To extract and visualize their preferences, we conscript the VOSviewer tool v.1.6.20 [29], and the output is shown in the map of Figure 3. The four (4) red-colored connections (close to each other) of the four keywords, namely, cybersecurity exercise scenario, modeling, deployment, and case study, with the circle keyword cyber range in the center, are considered strong relationships, whereas the green-colored connections namely, research, and cybersecurity training, are weak ones.

**Figure 3.** The results of keyword co-occurrence analysis (Included phase of Figure 1).

The result creates a cluster of simple keywords that reflects the overall research field of the 102 articles, producing a relationship map of the input’s most frequent co-occurrence keywords. Shortly, the map summarizes common articles and emerging trends concerning researchers’ preferences in the CRs field.

Note that one tool parameter is to configure the threshold node (by means of the maximum/minimum number of occurrences of an item in the file) [29,42], which could reduce the occurrence of low-importance words, accelerating literature analysis and exacting more representative hot keywords. In our case, the configuration was set to 3, that is, a medium configuration number of occurrences.

Finally, through the dataset’s thorough qualitative examination, we observed that only 11 of those 102 articles, as shown in the Included phase of Figure 1, examine the composition of the architecture and underlying infrastructure since our study mostly concentrated on this field. Those 11 articles are shown in Table 3 in chronological order (starting with the oldest one) next to the particular sector label of Table 2 where they belong.
4. Discussion

A comprehensive systematic literature review was conducted to examine the current state of research on the composition of CRs and to provide a comprehensive understanding of the field, considering the three RQs stated at the beginning of this work. In this section, we present an overview of multiple findings to support this.

4.1. General Interpretation of the Literature Findings

Reviewing 102 scholarly articles, we identified many CRs in seven selected DBs. Some platforms were open-source and freely available, while others were not. All of them are developed by companies, agencies, researchers, and professionals.

Our research strategy does have some limitations since the articles came from only seven selected DBs. For instance, some studies used old datasets that can no longer be used for future implementations. Moreover, many CRs are private; thus, many details about their architectural composition and underlying infrastructure are not disclosed or are mentioned very abstractly, and most developers need to reveal all the details, especially those in production environments.

Considering RQ1 and the datasets extracted from the articles’ background and shown in Table 2, the findings connote that existing research predominantly concentrates on six attributes, and many articles strongly emphasize them. These attributes play a key role in the CRs’ composition. Our analysis locates that attributes include a wide range of key elements, both technical (such as—physical or/and virtual—computer node, network device, IoT device, device controller, database, connection port, connection link, virtual machine controller, topology loader, and so on) and non-technical (such as participant/user, role, method, configuration file, and so on), and services. The analysis shows that CR platforms have attributes that make every platform unique.

Moreover, the analysis of the articles shows that every attribute corresponds to one or more values. In our effort to find the greatest degree of convergence in the vocabulary used throughout the fragmented literature and assuming accurate utilization, we present the attributes with the corresponding values. The utilization of attributes depends on the developer’s preferences and the rapidly changing environment, often making it unclear how to leverage their role. By assembling the most important points of the articles in the literature below, we illuminate their role.

4.1.1. Target Group

This attribute, along with the multiple values of students and professionals discussed in the literature [16], represents end-user engagement in a particular platform. One or more end-users, professionals, or security teams can be involved at the same time.
4.1.2. Sector

Articles [5,46] support that CRs were used for military and governance or universities in the past. However, currently, they can be deployed by many other organizations [17,47–52], encompassing values such as academia, military, private, public, and federated organizations.

4.1.3. Domain of Application

Article [53] supports that, regarding technology, CRs can be considered a single-domain if they address only a particular type of network or system (e.g., ITNs or CPSs corresponding to the attribute’s values) or a multi-domain spanning many networks and systems.

4.1.4. Technology

The literature discusses [16,17,54–56] three basic technological types to construct CRs, corresponding to the multiple values of simulation, emulation, conventional virtualization, and container-based virtualization. These types are (i) simulation and emulation; (ii) conventional virtualization segmented into two subtypes, (a) Type 1 and (b) Type 2; and (iii) container-based virtualization. In order to operate correctly all the different technologies, the authors in [57] state that they utilize an integrated infrastructure orchestrator layer that combines and supports these technologies based on Network Function Virtualization (NFV) and Software-Defined Networking (SDN) environments to set up customized solutions and training activities on CR platforms. The virtualized (software-based) infrastructure forms the basis for setting up and deploying CRs’ dynamic services and automating platform management, orchestration, security mechanisms, and data collection configurations to support different use cases.

Figure 4 highlights the layered architecture of these types that can be used in combination or separately, depending on applied circumstances. The two green dashed lines show the two heterogeneous groups of applications installable on the operating system (OS), and the red dashed line depicts the two subcategories of conventional virtualization.

**Simulation and Emulation (Host-based Solutions):** Simulation involves using a model, a virtual instance, to recreate a complex environment based on the real elements’ behavior. Emulation is when CRs run on the dedicated physical infrastructure [40]. Popular simulation examples are Simulink [58] for modeling and analyzing multi-domain systems; Cisco Packet Tracer (PT) [59]; Cisco Modeling Lab (CML) [60], formerly known as VIRL; NS2 [61] and NS3 [62]; and OMNet++ discrete event network simulators [63]. Popular emulation examples are Mininet [64], GNS3 [65], Emulab [66], and QEMU [67] as a network emulation testbed.

**Conventional virtualization:** Virtualization is achieved through virtual machines (VMs) with a real OS emulated by the hypervisor, which controls all access to the underlying physical hardware. There are two hypervisor types:

- **Type 1** (or bare-metal or native [68]) interacts with the underlying physical resources, replacing the traditional OS (e.g., MacOS, Windows, and Linux) altogether. Popular examples are VSphere [69], ESXi [70], Vagrant [71], Hyper-V [72], and XenServer (now known as Citrix Hypervisor [73]);

- **Type 2** runs as an application on an existing OS (MacOS, Windows, or Linux). It is used on endpoint devices to run alternative OS, and the host OS must be used to access and coordinate the underlying hardware resources. Popular examples are VirtualBox [74], Fusion [75], and Workstation [75].

**Container-based virtualization:** Using containers as a solution when there are issues with conventional virtualization [6,76], such as maintenance costs and usage of adequate hardware [56]. Popular examples are Docker [77], configured as Platform-as-a-Service (PaaS); OpenStack [78], which can be deployed as Infrastructure-as-a-Service (IaaS); LXC [79]; and OpenVZ [80].
4.1.5. Geolocation

CRs can be deployed and accessed in one of the following three locations.

**On-premises:** In this case, the architectural composition of the platform is simplest, using virtualization resources provided by developers with full awareness of the operating nodes [16,54].

**Cloud-based:** The architecture is based on appropriate web-based APIs. ECSO [16] reports three types: (i) private, (ii) public, and (iii) federated. Popular examples are AWS Elastic Compute Cloud (EC2) [81], IBM Cloud Computing [82], and Microsoft Azure [83].

**Federated CRs:** The authors in [16,84] refer to a unified, interoperable combination of cloud CRs integrating different types of DBs, applications, and systems. This interoperability allows for easier data sharing and collaboration across elements.

4.1.6. Scenario Mechanism

CR architecture includes true-to-life scenarios, which are something alive that should be produced, changed, and improved. The authors in [41] support the idea that scenarios can be created, designed, and saved in files. Scenarios (a set of preconfigured challenges, where each challenge contains a level of difficulty [85] that the users must follow and execute) are usually one or more configuration files written in YAML data modeling language or any other Domain Scenario Language (DSL) [46,86] used to model configuration, data, and orchestrated actions manipulated and controlled by protocols such as NETCONF and RESTCONF [68].

4.2. Challenges in the Literature Findings

Considering RQ2, data collection, analysis, and interpretation of the review, we detected that existing scholarly research on the CRs’ composition suffers from four points, as highlighted below.

**Architecture Considerations:** The systematic data collection process reveals that only some articles have available and public datasets concerning the composition of CRs, their elements and services interoperability, and data flows. From those that do present such information (cf. Table 3), it is discussed very abstractly, and thus many concerns arise. Specifically, the authors in [20,87] claim that CRs are composed of multiple independent elements that interoperate through a primary element; however, the authors in [22,88] support that in some cases, this leads to ineffectiveness during exercise execution if the correct
information from/or other elements is not achieved because interoperability facilitates timely, efficient, and effective completion of exercises.

**Cost Considerations:** While articles [5,46] state that CRs are not a cost-effective solution because they require substantial time, months of preparation, and financial investments [26,89], other articles (see next) support that a solution for this is automation.

**Automation Considerations:** The authors in [46,90–92] support that CRs should generate scenarios automatically, but articles [19,20] claim that automation adds complexity to their architecture development process. Generally, it is difficult to create a fully automated exercise platform [93,94], including VM cloning, host preparation, security-content installation, and the necessary features such as network topology. However, many automechanisms try to achieve this, like CyberVAN [95], Cyris (a tool inside the CyTrONE platform) [86], Astoria [96], AIT CR for gaming [19], CRATE Exercise Control (CEC) [43], and the tool proposed in [97] (using model-driven approaches as in [98]).

**Federation Considerations:** The literature concludes that building CRs where one can gain real-world cybersecurity experience is almost impossible. A single platform would be extremely complicated if it were to include all the necessary features, services, and functionality. Many articles state [40,57,84,99,100] that it would be realistic if multiple CRs, each within a specific domain of expertise, could collaborate to offer various activities. However, it is the most challenging architecture because interoperability orchestration and data sharing among them are complex since CRs are located in different places.

### 4.3. Future Research Directions

Finally, considering RQ3, although the reviewed literature provides a comprehensive overview of many fields, the architectural composition of CRs remains a field requiring additional research. Specifically, additional research is needed focusing on country-specific datasets to provide further insights. This will help to ensure that CRs are more manageable, reducing the configuration burden among multiple technologies and complex implementations.

While articles [20,22,41] propose solutions such as CR segmentation into layers, where each sub-layer, such as the core, virtualization/containerization, orchestration, and so on, defines different purposes and relies on different services, the issue arises about what elements and which services of those layers are responsible for establishing collaboration with other elements and services belonging to different or the same architectural layers do not reveal. Note that all these layers consist of various elements and services. Although a layered architecture serves different purposes, being independent of one another, the elements and the services running among the layers are dependent on each other. Manipulating different elements and services is a task that often requires complex administration of all elements (e.g., a network device, an interface, etc.) participating in a service. For example, if any change needs to be made, it should be concurrently across all elements and services participating in the platform. Moreover, during the dry run of the CR cycle life or an exercise, a change in these two factors should be either completely successful or rolled back to the starting state in case of failure. Changes need to be controlled and kept in sync across all layers of CRs. However, further investigation is necessary to achieve synchronization among the services and the elements belonging to different or the same layers in the entire platform.

As current platforms leverage a service-oriented approach where physical and virtual multi-vendor technology support complex services, the interoperability of the elements and services needs to be accurate and constant. Research in this direction is needed as this approach can support features like fine-grained configuration commands, bidirectional element configuration synchronization, node groups (e.g., network devices) and services, and compliance monitoring. The availability of these factors reduces complexity and identifies key points, helping to benchmark the aggregation points of layers, containment of potential fault propagations, and collapse, acting as a control boundary between them. Moreover, adopting the procedures followed for each platform is impractical, considering
the CRs’ complexity. A reason for this is that it may be difficult to find an architecture that will work across all cases, plus the fact that existing implementations are not intended to be deployed in many different environments.

5. Conclusions

In this work, we conducted a literature review of all available articles on CRs to the best of our knowledge. We collected and assessed the articles’ datasets by searching across seven DBs. The search term used for the search is the keyword Boolean expression (1) in Section 2.3. The articles were evaluated based on specific criteria and publication trends across time. The presented literature was examined in the topic analysis to study the composition of CRs with software help, following the updated PRISMA methodology based on its principles where sufficient detail should be reported to allow the trustworthiness and applicability of the review findings through three phases. This process helps to avoid subjectivity issues since other research teams may identify distinct overarching themes or interpret the findings differently. In the beginning (cf. Figure 1), after a preliminary search, 312 articles came up, and finally (after the Screening phase), 102 articles were fully reviewed. Next, these articles were analyzed with software help, which allowed us to gather the appropriate datasets and explore the composition of CRs. In the end, only 11 of the 102 articles examined CR architecture and underlying infrastructure, and even those only did so to a lesser degree.

The qualitative analysis of article outcomes concentrated on finding hot keyword co-occurrences among these researchers’ preferences, as depicted in the map of Figure 3 and segmented into datasets in Table 2. Next, we identify the attributes of CR architecture. Research on those attributes shows that CRs face many challenges. Through this process, we detected that we still have a limited grasp of interoperability and data flow among the elements and services belonging to different or the same architectural layer, as a small range of studies covered this field. This may be due to the difficulty of managing relationships between services and elements, which creates interoperability issues. Existing implementations often consist of methods without clear directions on the relationships, making it inherently challenging to implement intelligent technical solutions effectively. This lack is an important knowledge gap that needs further investigation, if not redefinition, and poses a serious issue in the architecture as the cybersecurity community seeks to tackle the quest for robust CR platforms. This venture’s challenge is to efficiently manipulate all the elements and services participating in a CR platform. As we navigate the ever-changing landscape of technologies and methodologies deployed by CRs, the diversity of their use (e.g., training, education, certification, defense, serious gaming, etc.) and complexity continue to grow; a holistic strategy is imperative. We strongly believe this study provides directions for several paths for future work.


Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References


85. Brown, R. Computer Security Threats: A Brief History; Dell Blog; Dell: Round Rock, TX, USA, 2014.


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.