Journal of Internet Services and Applications, 2024, 15:1, doi: 10.5753/jisa.2024.4363

Cybersecurity Testbeds for IoT: A Systematic Literature Review and Taxonomy

Khalil G. Queiroz de Santana [**Universidade do Vale do Itajaí** | ***khalil.santana@edu.univali.br***]
Marcos Schwarz [**Rede Nacional de Ensino e Pesquisa** | ***marcos.schwarz@rnp.br***]
Michelle Silva Wangham [**Universidade do Vale do Itajaí** | ***wangham@univali.br***]

Universidade do Vale do Itajaí (UNIVALI), Campus Kobrasol, São José, Santa Catarina - Brazil - CEP: 88102-700

Received: 9 April 2024 • **Accepted:** 17 July 2024 • **Published:** 04 October 2024

Abstract Researchers across the globe are carrying out numerous experiments related to cybersecurity, such as botnet dispersion, intrusion detection systems powered by machine learning, and others, to explore these topics in many different contexts and environmental settings. One current research topic is the behavior of Internet of Things (IoT) devices, as they increasingly become a common feature of homes, offices, and companies.. Network testing environments which are designated as testbeds, are boosting the effectiveness of network research. However, exploratory studies in IoT cybersecurity may include a wide range of requirements. This article seeks to carry out a survey of IoT cybersecurity testbeds. A critical systematic literature review was conducted to select relevant articles, by applying a novel taxonomy to classify the testbeds. The surveyed testbeds are classified in terms of their primary target domain and other features such as fidelity, heterogeneity, scalability, security, reproducibility, flexibility, and measurability. Furthermore, we have compared the testbeds with regard to each feature. Thus, the main contribution made by this study lies in a) the insights it provides into the state-of-the-art in IoT cybersecurity testbeds, and b) the emphasis laid on the main benefits and limitations that were found in the surveyed testbeds.

Keywords: Testbed, IoT, Cybersecurity, Systematic Literature Review, Taxonomy

Abbreviations

The following acronyms are used in this manuscript:

API	Application Programming Interface
C&C	Command and Control
CPS	Cyber-physical system
DDoS	Distributed DoS
DoS	Denial of Service
GUI	Graphical User Interface
HMI	Human Machine Interface
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IIoT	Industrial Internet of Things
IoT	Internet of Things
NFV	Network Function Virtualization
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SDN	Software Defined Networking
SLR	Systematic Literature Review
TUI	Terminal User Interface

1 Introduction

Given the increasing interconnectivity of today's devices, security is of the utmost importance. This is illustrated by the fact that, according to the Bad Bot Report [Imperva, 2023], 30% of the web traffic worldwide has its origins in bots or other malicious sources, resulting in cybernetic attacks and data breaches that cost, on average, 4.45 million dollars [IBM, 2024]. One of the security trends in the last decade has been the increased use of Internet of Things (IoT) devices, which may be more vulnerable because of their weak passwords, lack of software security updates, or the weak encryption that is generally employed in IoT devices [OWASP, 2018]. In this context, conducting research into cybersecurity mitigation strategies is essential for preserving user privacy, ensuring data integrity, and making such devices available in the broader network, as many unsecured IoT devices perform distributed denial of service (DDoS).

However, research that uses live systems may raise legal, ethical, and availability concerns [Sáez-de Cámara *et al.*, 2023]. For example, it is not recommended to test cyberattacks and mitigations in a critical industrial infrastructure or to collect network traffic data from a campus. Controlled and dedicated research environments may be used instead of a production infrastructure. These kinds of environments may come in distinct forms, such as simulation, emulation, or network testbeds.

A wide-ranging study of cybersecurity network testbeds and their features is required to understand the state-of-the-art and the limitations of the current testbeds, especially when account is taken of the changes brought about by the IoT paradigm, such as low-powered devices, wireless and mesh networks, heterogeneous protocols, edge computing, connected industries, and other factors. In light of this, this paper focuses on conducting a systematic literature review and a network testbed taxonomy. From the lens of this taxonomy, it is possible to analyze each surveyed work and note the unique features of each article. We argue that the lack of a rigorous taxonomy hampers any study, analysis, or comparison being made between the testbeds. Without this kind of taxonomy, the authors describe their works imprecisely and sometimes omit critical information such as the security features of the testbed.

Our contribution to this area can be summarized as follows:

1. We provide a novel structured taxonomy for describing cybersecurity IoT testbeds, by detailing a wide range of features required for a cybersecurity testbed such as fidelity, heterogeneity, scalability, security, reproducibility, flexibility and measurability. After this, we subdivide each feature into multiple characteristics in the form of a questionnaire so that the taxonomy can be applied uniformly across the testbed articles.
2. A survey is conducted of the IoT-based security testbeds described in the systematic literature review, by means of the previously defined taxonomy.
3. A meta-analysis is conducted of the articles, as a means of determining the major trends in the testbeds, and potential limitations observed, as well as possible future directions in the field.

The rest of this paper is structured as follows: Section 2 provides some background information about the network simulators, emulators, and testbeds. Section 3 describes the related works. Section 4 sets out our testbed taxonomy. Section 5 describes the systematic literature review (SLR) methodology and the results obtained. Section 6 introduces the cybersecurity network testbeds for IoT that were identified via the SLR, and arranges the articles in accordance with their primary target domain. In Section 7, we provide a detailed analysis of each testbed previously examined in accordance with our taxonomy. In Section 8, we conduct a meta-analysis of the works, and determine the main trends in testbeds, such as the use of virtualization and hybrid testbed architectures, while taking note of any limitations observed. Finally, Section 9 summarizes the concluding remarks and makes suggestions for future work.

2 Background

Network simulator software are programs that attempt to imitate the real-world behavior of a computer network, by enabling research to be carried out in areas such as routing protocols and interactions between devices [Rampfl, 2013]. Simulation is an efficient approach to investigate physical systems, as it offers a low cost and rapid comprehensive analysis [Siaterlis *et al.*, 2012]. NS-3 Henderson *et al.* [2008] is a famous open-source example. Simulators usually employ discrete-event models in which the simulation runs at discrete steps and logs interactions to meet the user's requirements. Although versatile and scalable, simulators do not model all the features of an environment, as they cannot model real payloads and timings that are present in real-world environments since simulator payloads are generated from an assumed distribution [Veksler *et al.*, 2018], which means that the network simulators impose some constraints with regard to fidelity.

In contrast, emulation combines simulation and testing in real systems. According to Lochin *et al.* [2012], emulation allows a network topology to be created and reproduces patterns of behavior observed in real networks (packet loss, limited bandwidth, etc) through artificial means. The architecture of an emulation system may vary between a centralized system such as Dummynet [Rizzo, 1997], NetEm [Hemminger *et al.*, 2005] or GNS3 [Grossmann and Duponchelle, 2008], where a single node hosts all the parties in communication, or a distributed system such as Emulab [University of Utah and Flux Research Group, 2024], where several nodes are used for emulation. In addition, according to Lochin *et al.* [2012], emulation can be assessed in many different ways, such as (*i*) through a static approach, in which the parameters are kept constant during an experiment, (*ii*) event-driven processes, in which events such as clock ticks govern the emulation, (*iii*) trace-based simulation, in which previously collected traces are reproduced, and (*iv*) virtualization-based network emulation. However, emulation also has its limits. In particular, as Gomez *et al.* [2023] state network emulators deviate from the expected behavior as more nodes are added to the network topology.

Network testbeds can be used to provide a higher degree of fidelity. Testbeds may consist of physical, virtual, or hybrid devices. Some testbeds like FIT-IoT Lab [Adjih *et al.*, 2015] focus on wireless sensor networks (WSNs) and mobility, while others focus on security [Wroclawski *et al.*, 2016], industrial IoT (IIoT) [Al-Hawawreh and Sitnikova, 2020], or vulnerability testing [Siboni *et al.*, 2018]. Furthermore, testbeds may employ various sensors, protocols, or wireless and wired connectivity, and be subject to attacks. This variable architecture, design, and purpose make testbeds versatile environments for different research topics. Some examples include dataset generation for intrusion detection systems (IDS), botnet behavior analysis, and other systems. Security-focused testbeds have extra requirements, as experimentation in this area may involve malicious applications. Thus, these environments must take extra precautions to enable experiments to be carried out safely without endangering the testbed itself or other entities in the network. Network testbeds also have their limitations; for example, the use of physical devices raises scalability and maintenance concerns as the hardware ages and becomes obsolete [Cappos *et al.*, 2018]. However, testbeds are still very useful for research and education; for example DETERLab [Mirkovic and Benzel, 2012] has been employed by more than 2,600 users in 47 institutions spread across 6 countries, with other notable testbeds being FABRIC [Baldin *et al.*, 2019], GENI [Demeester *et al.*, 2022] and PlanetLab [Peterson and Culler, 2002] .

3 Related works

This section describes other secondary studies, including their scope, purpose, findings, and other characteristics. Table 1 provides a summarized comparison between the related works and this survey.

Table 1. A Comparison of Related Works

Cite	SLR^A	Taxonomy	Security	Requirements^C	Purpose	Surveyed Testbeds	Trends	Challenges
Cintuglu *et al.* [2016]	No	Yes	Yes^B	Yes	Smart Grids	60	Yes	Yes
Chernyshev *et al.* [2017]	No	No	No	No	Simulators and Large Scale Testbeds	3	No	Yes
Waraga *et al.* [2020]	No	No	Yes	Yes	Proposing a new Testbed	4	No	No
Gomez *et al.* [2023]	No	Yes	Yes^B	No	Testbeds for Education and Research	14	Yes	Yes
Agrawal and Kumar [2022]	No	No	Yes	Yes	Industrial Cyber-physical systems	59	No	Yes
Kampourakis *et al.* [2023]	Yes	Yes	Yes	No	Wireless security testbeds	46	Yes	Yes
Ukwandu *et al.* [2020]	Yes	Yes	Yes	No	Cyber-ranges and Testbeds	49	Yes	No
Conti *et al.* [2021]	No	No	Yes	Yes	Industrial Control System testbeds	61	Yes	Yes
This Survey	Yes	Yes	Yes	Yes	Cybersecurity IoT testbeds	16	Yes	Yes

^{A} Systematic Literature Review; ^{B} Yes, but not the only focus; ^{C} Testbed requirements;

In Gomez *et al.* [2023], an in-depth survey is conducted with a wide range of network simulators, emulators, and testbeds. The survey briefly delineates the difference between each topic before providing details of its taxonomy and breaks down the network testbed branch into four categories: general purpose testbeds, specific purpose testbeds, production networks as a testbed, and on-demand testbeds. The survey then describes several testbeds that come within those categories, such as Emulab, GENI, DETERLab, P4Campus, and Amlight, among others, and in each section outlines the general architecture, experiment creation workflow, major features, and interconnection with other testbeds. Among the features observed, there are various degrees of isolation, experimental node representation (virtualization, emulation, containerization), network fabric programmability (P4 switches, OpenFlow switches), and available resources (GPUs, NetFPGAs, persistent memory, Infiniband), and thus they provide an overview of the testbed landscape. Lastly, the article compares the testbeds and concludes by noting the identified challenges and making recommendations for future works, such as providing cost-effective solutions and scalable emulated link interfaces, as well as encouraging collaboration.

The work by Waraga *et al.* [2020] carries out a survey that is focused on IoT attacks, and summarizes each article, the IoT devices tested, the tools used in the attacks, and the findings of the research. The authors also surveyed three IoT security testbeds, and compared the hardware and software used, devices tested, attacks, and experiments conducted to determine whether the testbed is automated.

For example, some testbeds are concerned with network packet analysis, while others with vulnerability testing, and employing devices such as network cameras, drones, and smartwatches.

In Chernyshev *et al.* [2017], another survey is conducted with a broad focus. This work describes testbeds (FIT-IoT LAB, Smart Santander, JOSE) and network simulators (IoT-SIM, CupCarbon, Cooja, OMNET++, NS3, among others). The surveyed testbeds are compared, with regard to the following attributes: scale, types of environment, heterogeneity (protocol & node), mobility, concurrency, federation, and primary use case. Although this work includes some security features, it does not focus on this area.

The survey conducted by Cintuglu *et al.* [2016], provides an overview of many distinct types of testbeds and simulators, in particular those related to industrial and power systems, as well as some general-purpose and security testbeds. In its findings, it noted the rarity of wireless systems, and as well as this, it had innovative ideas about large-scale hardware-based testbeds and the prospect of areas such as testbed federation, software-defined networking (SDN), and the need for multipurpose testbeds. Although the authors structured some of their research, their work is not a systematic literature review.

Agrawal and Kumar [2022] carried out a decade-long survey of industrial cyber-physical systems (I-CPS) with an emphasis on security. Comparisons between the surveyed works are provided, such as their general objective, security features, the attack taxonomies provided, and the type of infrastructure used for the testbed. Following this, the security challenges in CPS are divided into three perspectives, I-CPS security (availability, confidentiality, integrity, authenticity), I-CPS components (heterogeneity, interoperability, cohesiveness, coupling), and I-CPS systems (fault-tolerance, maintainability, scalability, reliability).

The work by Kampourakis *et al.* [2023] provides a systematic literature review of wireless cybersecurity-focused testbeds, and notes that although the use of wireless technology gives greater flexibility to cyber-physical systems, it also poses additional risks because this of wireless communication has a more open nature than more traditional wired networks. The authors then describe the surveyed testbeds, their features such as the wireless protocols in use, and the attacks explored in these works. Furthermore, they classify the attacks as either physical-layer, MAC-layer, transport-layer, application layer, or else attacks that are cross-layer, and these attacks are then also classified against Microsoft's STRIDE model.

The survey in Ukwandu *et al.* [2020] provides an overview of cyber-ranges and testbeds, with a focus on military, educational, and industrial areas. This work then documents the growing research interest in these domains, and underlines the importance of these testbeds and cyber-ranges for cybersecurity readiness and training. The authors also describe likely future trends in the area, such as the use of containerization as a lower-cost and more efficient alternative to traditional virtualization, as well as the use of augmented reality for training. Lastly, a testbed and cyber-range taxonomy is also proposed.

In Conti *et al.* [2021] there is a survey of Industrial Control System testbeds (ICS), which describe the architecture and devices used in these environments, as well as categorizing the testbeds as physical, virtual, or hybrid. The surveyed works are testbeds that are designed to represent a diverse set of sectors, such as power grids, gas pipelines, water distribution, nuclear plants, and cooling plants, as well as more generic CPS testbeds. This survey also includes an overview of datasets in the literature, and examines the various types of attacks carried out, such as network attacks (reconnaissance, Man-in-the-Middle, injection attack, replay attack, Denial of Service) and physical attacks. Finally, this work includes some 'best practices' for creating testbeds, datasets and intrusion detection systems.

The related works are summarized in Table 1, which displays the main features of each work, which are compared with those in this survey. A novel factor in this survey is that it focuses on security aspects unlike Chernyshev *et al.* [2017]. In comparison with Cintuglu *et al.* [2016], this survey only focuses on testbeds. Unlike works such as those by Waraga *et al.* [2020],; Agrawal and Kumar [2022] and; Conti *et al.* [2021] this work provides a testbed taxonomy to guide future research in this area. Moreover, in contrast with most other related works, this survey is a systematic literature review, like the one conducted by Kampourakis *et al.* [2023] and Ukwandu *et al.* [2020]. However, unlike these works and that of Gomez *et al.* [2023], this survey also documents cybersecurity testbed requirements.

4 Cybersecurity Testbed Taxonomy

This section defines the requirements for classifying each testbed based on its purpose, features, and scope. First we examine the general testbed features (Subsection 4.2), the testbed architecture (Subsection 4.3), and then the fidelity (Subsection 4.4), heterogeneity (Subsection 4.5), scalability (Subsection 4.6), security (Subsection 4.7), reproducibility (Subsection 4.8), flexibility (Subsection 4.9) and measurability (Subsection 4.10) requirements. Last of all, the testbed requirements taxonomy are displayed in Figure 1

4.1 Testbed Requirements selection

Network testbeds are complex environments that are designed to provide a high-fidelity testing ground for research, with a number of requirements to achieve this goal. Cybersecurity testbeds are a step beyond that, as their exposure to malicious software and their accompanying disrupting attacks such as DDoS, make additional requirements for this kind of testing environment. These requirements are listed in works such as Sáez-de Cámara *et al.* [2023], Wroclawski *et al.* [2016] and, Siaterlis *et al.* [2012], and the selected requirements for this survey were based on this literature. Furthermore, Appendix A defines in greater detail each aspect of any specific requirement.

4.2 General features

When analyzing testbeds, we first look at their general features, such as purpose, resource management, and other factors. Thus, the questions that have to be answered about these general characteristics can be summed as:

- Testbed purpose: what use cases is the testbed intended for? What are the target audiences, expected results, and main contributions?
- Is the testbed still available?
- Was funding provided for the testbed? If so, by whom?

4.3 Testbed architecture

Testbeds are complex environments with many discrete parts. Thus, there are several different ways to build one. In view of this, in this section, we seek to describe the general testbed architectural concepts, which are summarized in the question below:

- What are the general testbed characteristics regarding (*i*) experiment orchestration (experiment execution controller), (*ii*) resource management, (*iii*) user interaction?

4.4 Fidelity

Fidelity can be defined as the ability to achieve sufficient accuracy when modeling an experiment such as the inherent specific phenomena being studied [Wroclawski *et al.*, 2016]. This may involve scenarios that do not yet exist in the real world, which means that fidelity is a more sophisticated quality than realism [Wroclawski *et al.*, 2016]. For example, this could entail accurately representing an experiment that

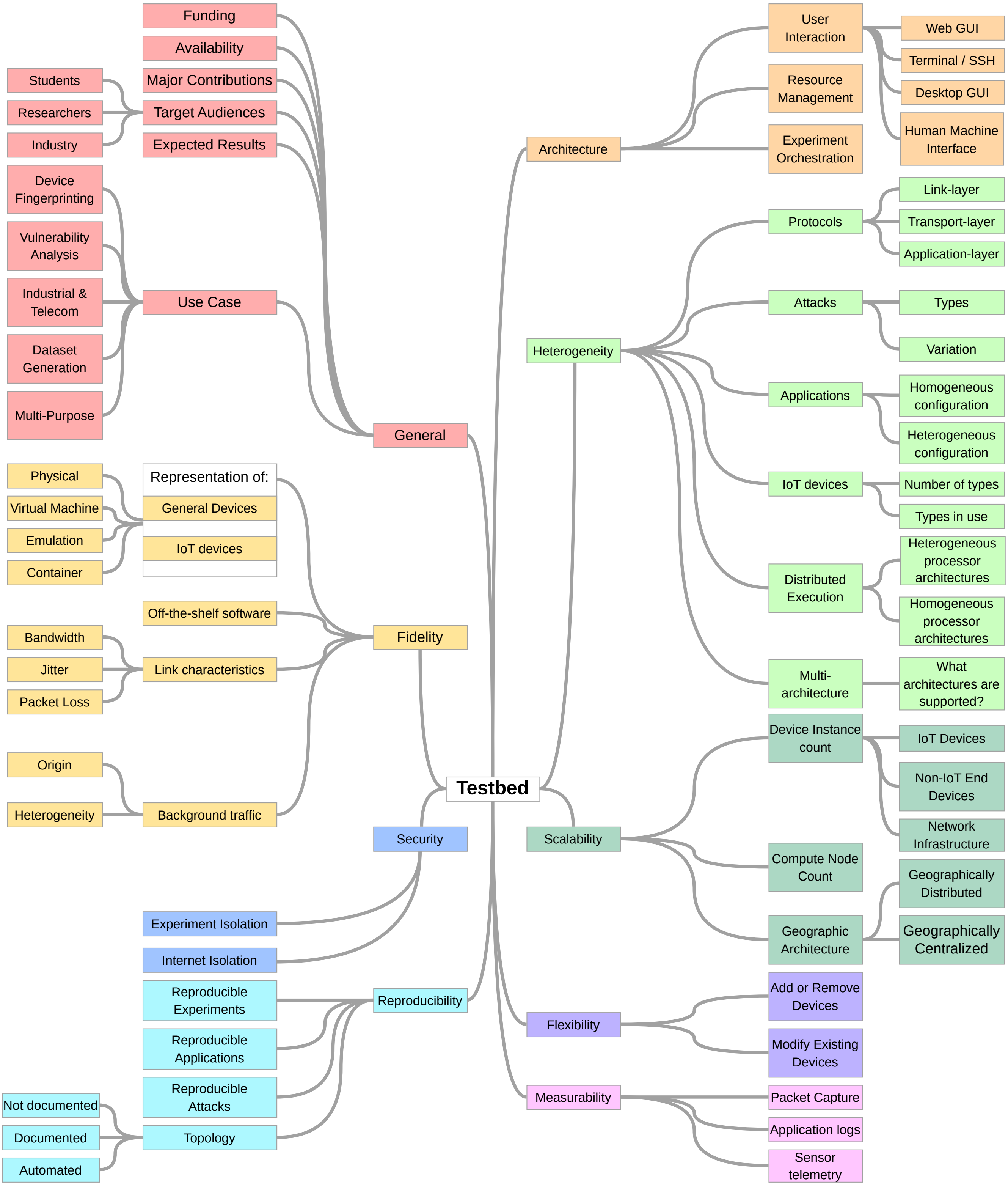

Figure 1. Taxonomy Of Testbed Requirements

is aimed at answering questions such as the capacity of IoT thermostats to perform a DDoS attack on a web server. This may involve studying specific vulnerabilities in the firmware of the devices in question that would allow the launch of such an attack, and for this reason, a real-world firmware may be used. Furthermore, these kinds of IoT devices may be connected to a wireless link such as Wi-Fi, which may limit the effectiveness of the attack from these devices. Thus, a testbed designed to provide fidelity in this situation should provide either the wireless link in question, or at least attempt to model the bandwidth, jitter, and packet loss characteristics of this link. Another important point to be considered in the case of a testbed aimed at creating datasets for further research (such as the use of the data to train machine-learning intrusion detection systems) is the realism of the background traffic. It must be determined whether the testbed accurately simulates real-world traffic by including a diverse set of protocols or if it is artificially generated or even absent. With this context in mind, the questions that have to be answered with regard to fidelity can be summarized as follows:

- How does the testbed represent its devices in experiments? (Physical, Virtualization, Emulation, Containerization)
- Does the testbed use *off-the-shelf* software such as Apache Web Server or malware like Mirai, ZeuS, etc?
- Does the testbed represent link characteristics such as (*i*) bandwidth, (*ii*) jitter, (*iii*) packet loss?
- Does the testbed feature multiple broadcast domains, that represent a more realistic and complex network?
- Does the testbed incorporate background network traffic, and if so, what is its nature? This includes a) assessing whether the traffic is artificial (replayed) or realistic and b) evaluating the heterogeneity, which may range from (*i*) no background traffic, (*ii*) homogeneous/single protocol, or (*iii*) diversified/heterogeneous protocols.
- How does the testbed model the sensors/IoT devices?

4.5 Heterogeneity

Sáez-de Cámara *et al.* [2023] define heterogeneity as a multidimensional quality, for example, in terms of node protocols, attacks, and services that behave differently in the same topology, as node heterogeneity by itself might not be enough to achieve realism. With regard to the context, a testbed aimed at providing a testing ground for corporate network security, may include a diverse set of protocols. This is because these environments include several desktops that carry out activities such as a) web browsing and e-mail, b) servers performing tasks such as serving webpages or acting as databases, c) the use of network infrastructure involving routers providing DHCP services, and other factors. In addition, with regard to IoT devices, these may can make use of different protocols depending on their purpose, such as a network security camera that streams H.264 video to a central location over an Ethernet link, while an environmental sensor may be connected to a Zigbee network and employ a Zigbee2MQTT gateway to transmit data to a cloud endpoint.

Each of these devices may be configured in a wide range of ways, for example, a) by sending data continuously or in specific time intervals, b) by deciding whether to use authentication or not, and c) by deciding whether to deploy encryption or not. Moreover, in such a diverse environment, a number of different types of attacks may be possible, such as ARP spoofing for Man-in-the-Middle attacks, DDoS attacks against web servers, worms, and ransomware targeting desktops and servers. These attacks may also be heterogeneous; for example, a DDoS attack may make use of several techniques such as slow-and-low attacks like Slowloris or more brute-force attacks such as TCP SYN floods. Thus, the heterogeneity requirements can be summarized into the following questions:

- What types of network protocols are employed? - in particular, regarding (*i*) link layer, (*ii*) transport layer, and (*iii*) application layer protocols?
- Does the testbed model have multiple types of attacks?
- Are there varied attack-type instances among them? How heterogeneous are these attacks? (*i*) No variation (*ii*) multiple attack instances with distinct parameters.
- Are the application configurations diverse, such as web servers with or without encryption, different authentication protocols, etc?
- How many types of IoT devices/sensors are used? What are these types?
- In the case of a distributed architecture, does the testbed support the execution with heterogeneous processor architectures?
- Does the testbed demonstrate that it has connected devices of distinct processor architectures (ARM vs. x86, etc.)?

4.6 Scalability

According to Wroclawski *et al.* [2016], a testbed must be able to support experiments at a sufficient scale to be representative and capture complex scale-related effects. Owing to the ever-growing number of IoT devices, these are often weaponized to perform various malicious tasks such as DDoS attacks which usually involve a large number of these devices. In light of this, a testbed that seeks to provide a suitable IoT DDoS testing environment must include enough devices to obtain accurate research results. This scalability may be a hard goal to achieve; for example, the use of physical devices may be too expensive to scale, while other approaches such as virtualization or containerization may only achieve higher scalability at the cost of some fidelity. This kind of containerization or virtualization may be operated in computer servers that run containerization engines such as Docker or virtualization layers such as Virtualbox. A set of servers acting as virtualization or containerization hosts is designated a compute server in this survey, such as a testbed with a three-node Proxmox hypervisor cluster which is a 3-compute-node testbed. It should also be noted that this feature only applies to testbeds that employ these technologies.

Last of all, the testbed resources may be either geographically distributed or centralized; in the case of the former, some factors such as realistic latencies and wireless signal degradation may be represented, while in the case of the latter, these features need to be emulated. With this context in mind, regarding scalability requirements, the following questions can be formulated:

- How many devices are represented in the testbed in total, including non-IoT devices?
- How many compute nodes are used in the testbed?
- Is the testbed architecture locally or geographically distributed?

4.7 Security

Conducting cybersecurity experiments in real production devices and networks incurs risks with regard to security and availability. Furthermore, collecting network traffic may impose a legal and ethical barrier to research [Sáez-de Cámara *et al.*, 2023]. According to Siaterlis *et al.* [2013], safe execution is required for a cybersecurity-oriented testbed, as most security experiments assume that some form of malicious software is always present.

This means a security testbed must be able to protect potentially disruptive experiments safely. An example of such a danger might be the research into the activity of computer worms. These viruses can spread through vulnerable systems outside the testbed environment if not properly contained, and cause damage to other vulnerable devices such as the university campus or other hosts on the wider Internet. There are other risks that degrade the service of other legitimate applications that may share the infrastructure with the testbed. For example, a DDoS attack that cuts across a shared Ethernet link may cause other services to slow down or become unavailable, even if the attack itself is confined to a specific VLAN. Finally, a testbed can provide isolation between experiments, in such a way that several researchers can simultaneously study distinct phenomena without either experiment affecting each other. Thus, the security features can be summarized in the following questions:

- Does the testbed feature isolation between experiments?
- Does the testbed feature isolation between the testbed and the Internet?

4.8 Reproducibility

The authors Wroclawski *et al.* [2016] define reproducibility as the ability to reproduce and build upon the (experimental) results of others. They add that a deterministic execution may not be viable for scenarios with physical systems of significant complexity that evolve over time.

Siaterlis *et al.* [2012] argue that reproducibility represents the capacity to repeat an experiment and obtain the same or a statistically consistent result; in light of this, repeatable experiments require a controlled environment with a well-defined initial state and include all the events until the final states. An example of how this requirement may be achieved is the documentation of the testing environment and parameters, coupled with automation such as scripts that configure this environment. An interesting approach to meet this requirement is the use of an infrastructure as code, such as Docker, which enables researchers to deploy applications and services into a well-defined container image. Lastly, the network topology needs to be well documented to enable this reproducibility to occur, and a reproducible topology can be undertaken via network diagrams or may even be created programmatically via scripting. Given this context, the reproducibility requirements can be summarized in the following questions:

- Does the testbed enable the execution of reproducible experiments?
- Are applications reproducible?
- Are attacks reproducible?
- Which of the following best describes the network topology? (*i*) not documented, (*ii*) documented, (*iii*) automatized or otherwise scripted

4.9 Flexibility

This quality is defined as the ability to modify the algorithms or behavior of the modeled devices [Wroclawski *et al.*, 2016], by conducting an experiment to study similar phenomena based on other parameters, starting conditions, runtime events, or other variables. For example, a DDoS experiment may have a varying number of devices participating in an attack, and each of those devices may behave differently by being closer (latency-wise) to the attack target or may have access to other computing resources. Furthermore, given the vast number of areas in which IoT devices are employed, there might not be a one-size-fits-all topology. In other words, the testbed should provide some level of flexibility concerning the arrangement of the network topology itself, to enable experiments to model environments ranging from small office and home networks to industrial networks. Thus these flexibility requirements can be boiled down to the following questions:

- Is it possible to modify the testbed to perform new experiments? In particular, (*i*) by adding or removing devices and (*ii*) by modifying the behavior of existing devices?
- How is the topology assembled? (*i*) manual assembly, (*ii*), configurable (SDN,NFV), (*iii*) scripted

4.10 Measurability

Sáez-de Cámara *et al.* [2023] defines measurability as a) the capacity to measure raw network packets from any node, including multiple locations at the same time, and b) the ability to measure application-level logs, as a means of enabling the creation of datasets with diverse sources. Another key requirement is that experiments should be accurately monitored, and the measurements should not interfere with or alter the outcome of the experiment [Siaterlis *et al.*, 2012]. For example, testbeds may collect information such as the network traffic during a computer worm propagation experiment to later determine how a set of IDS may react, or even design a new IDS system that is tailored to detecting that kind of attack. Testbeds may also collect application logs such as Apache Web Server to evaluate existing or newly built Web Application Firewalls (WAFs). Lastly, given the presence of IoT systems in today's networks, sensor telemetry is also an essential factor that must be analyzed, as events such as anomalies may serve as an indicator of compromise. In light of this, the measurability functions can be summarized in the following question:

- What kinds of artifacts are collected by the testbed? (*i*) network captures (pcaps), (*ii*) application logs, (*iii*) sensor telemetry

5 Systematic Literature Review

A systematic literature review (SRL) should be carried out to evaluate and interpret the research papers relevant to a specific, previously established research question [Kitchenham, 2007]. This kind of evaluation must be thorough and employ a rigorous methodology for synthesizing the appropriate studies. According to the author, the systematic literature review involves several distinct activities, including review planning, execution, and review reporting. In summary, the research questions and the search protocol are specified in the planning stage of the review. The principal works on the research question are identified during the execution phase with the aid of the derived search terms, starting from a set of 824 articles, which are then analyzed in phases to reach 16 selected works. Finally, the reporting phase of the review covers the presentation of the results obtained and their formatting and dissemination [Kitchenham, 2007]. Thus, the executed SRL process is explained in the following sections and summarized in Figure 2.

5.1 Research Question

Cybersecurity testbeds are an important tool for research, and, given the adoption of the IoT paradigm and the new risks and challenges it involves, security testbeds must evolve to model real-world environments and situations accurately. Hence, the research question for this survey is: What cybersecurity testbeds can enable the execution of experiments that include IoT devices? More specifically, after these works have been compiled, other research questions are broken down into testbed requirements in the form of a questionnaire, such as, how these testbeds model IoT devices (physical, virtualized, emulated, etc), or which aspects of the testbed are measurable (packet captures, application logs, sensor telemetry), and others. A complete list of these sub-research questions is provided in Section 4, and further application instructions and context are provided in the questionnaire in Appendix A.

5.2 Research Keywords and Query String

The keywords were extracted from the search question, followed by the addition of other related (synonymous) keywords. From this set a query string was created:

> ”Testbed” AND (”Security” OR ”Cyber-security” OR ”Cybersecurity”) AND (”IoT” OR ”Internet of Things” OR ”Internet-of-Things”)

5.3 Inclusion and Exclusion criteria

In view of the large size of online article databases, some filtering is required to find articles of interest efficiently. In the same vein, some works may only briefly mention the keywords of interest without going into detail about those topics. Thus, additional filtering is required to remove these kinds of articles that match the search query but do not answer the research question.

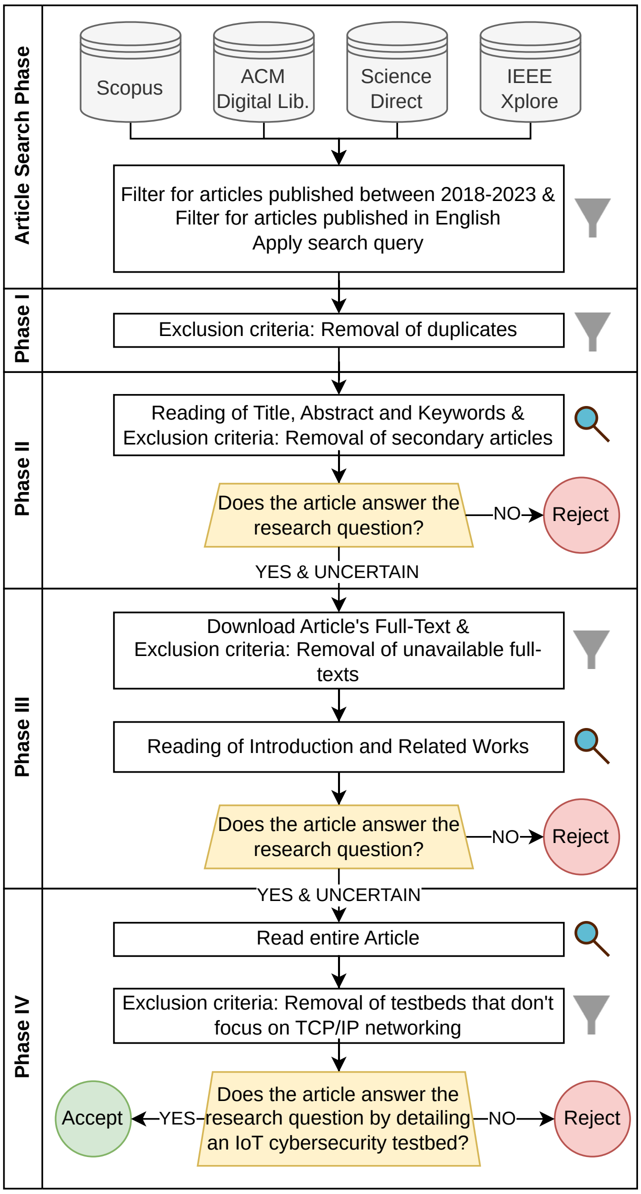
Article Search Phase
Scopus
ACM Digital Lib.
Science Direct
IEEE Xplore
Filter for articles published between 2018-2023 & Filter for articles published in English Apply search query
Phase I
Exclusion criteria: Removal of duplicates
Phase II
Reading of Title, Abstract and Keywords & Exclusion criteria: Removal of secondary articles
Does the article answer the research question?
NO
Reject
YES & UNCERTAIN
Phase III
Download Article's Full-Text & Exclusion criteria: Removal of unavailable full-texts
Reading of Introduction and Related Works
Does the article answer the research question?
NO
Reject
YES & UNCERTAIN
Phase IV
Read entire Article
Exclusion criteria: Removal of testbeds that don't focus on TCP/IP networking
Does the article answer the research question by detailing an IoT cybersecurity testbed?
YES
Accept
NO
Reject

Figure 2. Systematic Literature Review process

5.3.1 Inclusion Criteria

- Studies that give evidence of a testbed with IoT and security focus.
- Articles published between 2018 and 2023.
- Only articles published in English were considered.

5.3.2 Exclusion Criteria

- Articles that do not focus on IoT and security.
- Duplicated articles between databases.
- Articles where the full text was not available.
- Books, surveys, and other secondary studies.
- Testbeds that exclude TCP/IP networking, focusing solely on link layers. It should be made clear that the exclusion factor is the absence of TCP/IP.

5.4 Selected Article databases

The research query was then executed with the aid of four major databases: IEEE Xplore, ACM Digital Library, Elsevier ScienceDirect, and Scopus. This search was performed between April and July 2023 and concentrated on articles up to five years before that. This limit was imposed so that more up-to-date trends and challenges in the area could be searched, such as the current technologies in use, as well as attempts to locate testbeds that are still available for use. Appropriate filters were applied during this query to filter out secondary knowledge, such as books, or superficial works, such as posters.

5.5 Results

The methodology was systematized and divided into phases, as described below, and is summarized in Table 2. Table 3 goes into more detail by listing the SCIMAGO publication score, the user interface type employed, and the way the work was categorized.

Phase I

Duplicate removal was executed as a first step in pruning the results; in total, 285 duplicates were found, divided between IEEE Xplore (4), ScienceDirect (18), and Scopus (263).

Phase II

This phase focused on reviewing and filtering articles on the basis of their title, abstract, and keywords. Many articles were removed in this stage, such as articles that only mention security or IoT in passing, or articles clearly beyond the scope of this literature review.

Phase III

The introduction of each article and the related works section were analyzed to determine their relevance to the research topic, which involved filtering the results further.

Phase IV

In the case of articles that could not yet be removed, an entire reading was executed, acting as a final delimiter, to ensure that articles that fit the research criteria remained. In this phase, we removed some articles on the following grounds: a) the lack of a testbed as the focal point (7), b) a failure to include IoT devices (5), c) a lack of a security focus (5), d) the fact that it only included Wireless Sensor Networks (3), e) it was only a testbed proposal without a concrete implementation (1), f) it did not include IP networking or only focused on radio-frequency (8), g) and other articles were removed because they failed to refer to the research question in other ways. Table 2 shows that 16 articles were selected in this last phase.

6 The Classification of Iot Cybersecurity Testbeds

This section introduces each paper selected in the systematic literature review and groups them by their main goals and general features. Hence we categorized the surveyed works as (*i*) testbeds for vulnerability analysis, (*ii*) testbeds for IIoT and telecom use, (iii) testbeds for fingerprinting, (*iv*) testbeds for dataset generation and, (*v*) multi-purpose testbeds. There is some overlap between the categories as some testbeds target more than one use case. This categorization is displayed in Figure 3.

6.1 Testbeds for vulnerability analysis

Some testbeds have a narrower focus, such as investigating exploits and other attack vectors in IoT devices. These testbeds usually study one device at a time and run automated network scans to probe the device for vulnerabilities, and then produce a report based on its findings. An example of these testbeds is Bettayeb *et al.* [2019], which runs network port scans, network traffic analysis, man-in-the-middle attacks, decompiling mobile applications, and interactions with the IoT device. This testbed only studied two devices, a smart socket, and a thermostat, and found vulnerabilities in the former.

In Waraga *et al.* [2020] another testbed is described that targets vulnerability analysis. This work also analyses two IoT devices (smart bulb and wireless camera) for vulnerabilities in 16 test cases using tools such as Skipfish, Nmap, SSLScan, Nikto, Metasploit, among others. The wide array of data collected is shown in a Graphical User Interface (GUI), which contains the logs from each test, graphs of the network traffic generated, the last external IP connections, and a summary of the results of a test case.

Another example of this category of testbed is Siboni *et al.* [2018], which also studies flaws in IoT devices. This testbed examines devices in a shielded room, where stimulators interact with devices, security tools launch attacks, and probes and measurement devices collect data from the device under test. This testbed includes network analyzers for Wi-Fi, Bluetooth, ZigBee and also includes stimulators for GPS positioning, NTP/Timezone data, movement, lighting, and audio, which enables an analysis to be conducted of IoT devices in many distinct scenarios.

6.2 Testbeds for Industrial and Telecommunication systems

Oliver *et al.* [2018] provide a testbed for telecommunication environments powered by OpenStack. This testbed features network function virtualization (NFV), SDN, and 5G to create a telecommunications infrastructure for safety-critical environments such as remote medicine, that includes the use of a 'robot surgeon' powered by a Universal Robotics UR3 arm. This work aims to ensure a) that the application data is not tampered with, b) that applications continue to function even during an attack, and c) that applications fail or degrade safely. The authors also list some of the areas of

Table 2. Search results and works preserved/accepted per phase

Database	Results	Phase I	No access	Phase II	Phase III	Phase IV
IEEE Xplore	318	314	1	48	27	8
ACM Digital Library	23	23	0	5	5	1
ScienceDirect	34	16	0	2	1	1
Scopus	449	186	3	45	22	6
TOTAL	824	539	4	100	55	16

Table 3. Overview of accepted works from the survey

Cite	Availability	Purpose	Source	Type*	User Interface
Al-Hawawreh and Sitnikova [2020]	Yes	IIoT Testbed	IEEE	J (Q1)	Web GUI, SSH
Sáez-de Cámara *et al.* [2023]	Yes	General Purpose	IEEE	J (Q1)	Desktop GUI
Beauchaine *et al.* [2021]	No	Vuln. Analysis	IEEE	C	SSH
Koroniotis *et al.* [2021]	Yes	Dataset Generation	SCOPUS	J (Q1)	Web GUI
Siboni *et al.* [2018]	No	Vuln. Analysis	ACM	J (Q1)	Web GUI, SSH
Moustafa [2021]	No	Dataset Generation	SCOPUS	J (Q1)	Unknown
Gardiner *et al.* [2019]	Yes	IIoT testbed	IEEE	C	HMIs
Kumar and Lim [2019]	No	General Purpose	SCOPUS	C	Unknown
Oliver *et al.* [2018]	No	Telecom Research	IEEE	C	Unknown
Lee *et al.* [2021]	No	Dataset generation	ScienceDirect	C	Unknown
Lee *et al.* [2018]	No	IIoT Testbed	SCOPUS	J (Q2)	Unknown
Thom *et al.* [2021]	No	General Purpose	IEEE	C	Unknown GUI
Waraga *et al.* [2020]	No	Vuln. analysis	SCOPUS	J (Q1)	Desktop GUI
Nock *et al.* [2020]	No	General Purpose	IEEE	J (Q1)	WebGUI
Bettayeb *et al.* [2019]	No	Vuln. Analysis	IEEE	C	Desktop GUI
Babun *et al.* [2020]	No	Fingerprinting	IEEE	C	Unknown

$*$ 'C' for Conference paper and 'J' for Journal. In the case of Journals, their publication score according to SCIMAGO/SJR is present between parentheses

research assisted by the testbed such as DDoS and network anomaly protection, and they give extensive details of mechanisms for attestation using TPM (Trusted Platform Modules). Lastly, the authors give a practical example of the testbed's built-in protections by executing a DDoS attack between the robot and its control center. This shows that an attack is only successful if its defenses are disabled.

In Gardiner *et al.* [2019] there is an Industrial Control System (ICS) testbed that represents a water treatment plant. This testbed is used for security analysis, intrusion detection, and dataset generation and is based on the author's previous experience in designing an ICS testbed. The testbed is formed of three zones: Experiment Level, Control Level and Production Level, segregated by VLANs. Each contains an infrastructure such as network appliances, SCADA workstations, sensors, programmable logic controllers (PLCs), human-machine interfaces (HMIs), and remote terminal units (RTUs). Virtual machines based on VMWare vSphere are used to install and manage software such as ClearSCADA, Windows and Linux. The authors also define five requirements for an IIoT and ICS testbed: diversity, scalability, complexity, data capture and safety. Lastly, the authors compare their work with other testbeds with regard to their physical device diversity, industrial protocol diversity, process diversity, flexibility, scalability, fidelity, simulation support, support for security analysis, monitoring and openness. It should be noted that this testbed also includes dataset generation within its scope.

Lee *et al.* [2018] describes a cybersecurity IIoT testbed and an intrusion detection system (IDS). The testbed only features physical devices, (16 in total), which are distributed among units such as PLCs, HMIs, RTUs, and SCADA controllers, as well as IDS and firewall and controller units. The authors then demonstrate 5 separate attack scenarios (data exfiltration, DoS, buffer overflows and DNP3 exploits). These attacks are detected in the simulations executed by the IDS, which did not result in extra delays, packet loss or instability. The IDS inputs are from a packet duplicator, while the detection method is based on three systems: a protocol-specific whitelist model, a packet-diversity anomaly detection system, and a one-class support vector machine learning model.

Al-Hawawreh and Sitnikova [2020] outline their IIoT testbed (Brown-IIoTbed) which features physical and virtualized components such as pfSense firewalls, Raspberry Pi gateways, Arduino MEGA, Iphone and Android mobile devices, as well as virtual machines of Windows, Kali Linux, Ubuntu. These devices may either feature a set of sensors (pressure, temperature) or run applications such as CoAP servers, SQL servers, txThing, mail servers and web browsers.

The testbed features 6 attack types based on Microsoft's STRIDE model: Spoofing (ARP spoofing), Tampering (MiTM), Repudiation (sending fake data), Information disclosure (sniffing), DoS and Elevation of Privilege. The authors then examined the use of machine learning models (Random Forest, Decision Tree, Naive-Bayes, Logistic Regression, and k-nearest neighbor) for intrusion detection (IDS) and, obtained accuracy scores ranging from 42.4%

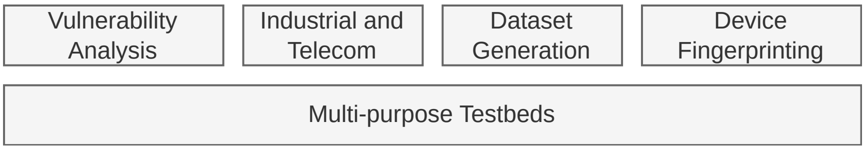

Figure 3. Testbed classification

(Naive-Bayes) to 99.99% (Random Forest). The authors conclude their article with a comparison against 10 other testbeds regarding their usability, fidelity, heterogeneity, interoperability, and other features, as well as the limitations of the testbed. For example, the edge gateway has limited connectivity which may affect its performance in scenarios with other PLC devices.

The SAir-IIoT testbed [Koroniotis *et al.*, 2021] focuses on providing a multipurpose cyber-twin of an airport using heterogeneous devices and protocols. This testbed features multiple zones with physical IoT devices that use wireless protocols such as Wi-Fi, Bluetooth, ZigBee, LoRa, and Z-Wave, as well as other IoT application layer protocols such as MQTT. Alongside those devices, there are 20 virtual machines in use with a wide range of operating systems (Ubuntu 18.04 & 14.04, Windows 7 & 10, Metasploitable3, pfSense firewall), 10 of them being benign and 10 of them employed for malicious scenarios by means of software suites such as Kali Linux.

6.3 Testbeds for fingerprinting

In Babun *et al.* [2020] there is a testbed specialized in fingerprinting IoT devices, specifically those that rely on Zigbee and Z-wave protocols. This testbed features 39 IoT devices such as smart outlets, switches, locks, motion sensors, and dimmers. The authors then describe their use of AVR RS UZB and Z-Wave 500 Zniffer as sniffers for Zigbee and Z-wave protocols, which can capture data such as timestamps, source and destination addresses, protocol type, packet length, and the time elapsed from the previously captured frame. This information is then processed and filtered to maximize the accuracy of the machine-learning models tested, which obtained over 90% average accuracy, precision, and recall for Zigbee and Z-wave protocols.

6.4 Testbeds for Dataset generation

The TON_IoT testbed [Koroniotis *et al.*, 2021] is used to create the TON_IoT dataset, which incorporates IoT devices such as Iphones 7s and a smart-TV, as well as many virtual machines with various operating systems (Windows 7 & 10, Ubuntu 18.04 & 14.04, Kali Linux). The testbed also features some vulnerable endpoints (Metasploitable3 and Damm Vulnerable Web App) and a plethora of ancillary services (DHCP, E-mail, DNS, FTP, MQTT Broker). Lastly, a Netsniff-NG tool was used to record the network traffic from the entire testbed in .pcap files, which were then used in conjunction with the Zeek utility to extract data features from these files. This article then goes on to describe a data analytics pipeline to train various machine learning models (Gradient Boosting Machine, Random Forest, Naive Bayes, Deep Neural Networks), as well as to classify normal and attack events.

The fifth generation (5G) connectivity is an emerging system that is adopted in many industries. Similarly, the testbed in Lee *et al.* [2021] covers an environment for collecting and labeling critical 5G data from a replayed network capture of a model IIoT factory. Both benign and malicious network flows were replayed in order to produce this kind of dataset, with the malicious flows containing simulated attacks. This work also describes a dataset conversion utility with a GUI to assist in labeling the data records. In addition, it takes the extra steps required to handle the encapsulated data in a 5G network, including stripping tunnel headers and manipulating fields such as source IP and MAC addresses.

6.5 Multi-purpose testbeds

This section examines testbeds that do not fit a single purpose, or that do not target a single use case. The IoT-CR (Cyber Range) [Nock *et al.*, 2020] is a testbed built upon the Cooja emulator. This testbed assists in the creation of physical and virtual IoT networks and the concurrent execution of multiple scenarios. The physical devices employed are 20 Zolertia RE-MOTE IoT devices, that form a peer-to-peer mesh network over IEEE 802.15.4, and are connected to the testbed via a USB-tree cabled topology, while the virtual component is powered by Cooja and Contiki-NG. The testbed also features a CLI/TUI tool, a Web Interface and an API endpoint powered by Python 3.7, Flask and SQLite3. Among the actions enabled by the testbed are login/signup flows, uploading topologies, scripts, and the ability to view created jobs and logs, which are available via API and a CLI wrapper also written in Python. Lastly, the authors create a scenario that mimics a MiTM example of a red/blue ”pass the token” where one team attempts to increment a value by one while the other tries to do the opposite. When the scenario reaches either its upper or lower limit, the victory or defeat of each team is determined.

In Thom *et al.* [2021] there is a multipurpose testbed that models a small-scale city with a range of physical and virtual IIoT and IoT devices such as Raspberry Pis, as well as an SDN area that is reconfigurable. The testbed features Cowire, Conpot and Dionaea honeypots to gather SSH and FTP login attempts, traffic tunneling, and other functions. Among the activities enabled by this testbed is research and training for students in security techniques (e.g. scanning, fingerprinting), as well as more general studies about IoT communications. The testbed is also notable for featuring wireless and wireless links, as well as some industrial protocols such as PLC and MODBUS which are suitable for interacting with certain mechanical elements of the model smart city.

The iBOT_IoT testbed [Beauchaine *et al.*, 2021] represents a small testbed with two scenarios: (*i*) a home testbed with two Raspberry Pi 2 and one Raspberry Pi 3 device connected to a router, which is then connected to the victim server and a botnet control and command (C&C) endpoint; (*ii*) an enterprise scenario with many access points in a mesh setup featuring RADIUS authentication with the three Raspberry Pi devices connected similarly to the first scenario. This testbed then studies various attacks against a web server by the Vivid botnet and the UFONet deployments, which resulted in this server failing to handle requests after a minute of DoS. The authors note that no firewalls or extra hardening were configured, and that the Vivid botnet has one advantage over the more traditional Mirai botnet, as the former does not require DNS. Finally,, the botnet server operates on a virtual machine using bridged networked adapters and VirtualBox as the hypervisor.

Sáez-de Cámara *et al.* [2023] displays a testbed based on the open-source GNS3 network emulator (Gotham Testbed). This testbed extends GNS3's functionality using scripting, APIs and middle-ware, and allows experiments to be carried out in containers and virtual machines. As a means of building the experimental topology, QEMU-based virtual machines are employed as routers, and IoT devices are represented by Docker containers. Each instance of these devices is created by Gotham's template creation engine, which allows RAM, CPU, and disk limits to be established, as well as other parameters such as environment variables and start commands. From this point, the topology builder can handle the network configuration files of each device, or even install the appropriate software on the routers from scratch. Afterwards the scenario generator starts all devices in a specific order and enforces the CPU, RAM and bandwidth limits that have been previously established. As well as this, the scheduling network captures, attacks and runs arbitrary scripts on each node. It should be noted that this testbed also includes dataset generation within its scope.

The work by Kumar and Lim [2019] outlines a testbed based on DETER-Lab [Wroclawski *et al.*, 2016] for IoT botnet research and mitigation. More specifically, this work focuses on the well-known Mirai botnet, which is famous for its record-breaking 1.2Tbps DDoS attacks. The authors list the adaptations needed on the Mirai source code to ensure the bots are connected to their C&C server, as Mirai uses DNS resolution with Google's public resolvers (8.8.8.8), and this resolver is not available in the isolated testbed setup. This testbed uses Network Simulator (NS) scripts to orchestrate the experiment (start, stop, modify), and emulate Raspberry Pi devices in QEMU virtual machines connected to the underlying network via a Layer 2 virtual switch (bridge). Finally, the authors perform TCP SYN and UDP flood DDoS attacks and release 46-48,000 packets per second against the victim server, which results in a reduction of legitimate client traffic.

7 Cybersecurity Testbed Landscape

This section outlines the existing cybersecurity testbeds and their characteristics in areas previously designated in our taxonomy. Table 4 provides a summarized comparison of fidelity features while Table 5 contains the heterogeneity feature summary comparison, and finally, Table 6 provides a summary of scalability, security and measurability.

7.1 General characteristics

With regard to general testbed features, the testbeds have been divided into five distinct groups regarding their main purpose, as seen in Subsection 6.1 through Subsection 6.5. In the same context, we have discovered that the surveyed testbeds usually focus on researchers [Al-Hawawreh and Sitnikova, 2020; Sáez-de Cámara *et al.*, 2023; Moustafa, 2021; Gardiner *et al.*, 2019; Waraga *et al.*, 2020; Nock *et al.*, 2020], while some also focus on students [Beauchaine *et al.*, 2021; Kumar and Lim, 2019; Thom *et al.*, 2021], with their major contributions ranging from complex multi-layer IoT scenarios [Sáez-de Cámara *et al.*, 2023; Moustafa, 2021] for dataset generation, to more restricted areas such as vulnerability analysis devices in isolation [Siboni *et al.*, 2018; Waraga *et al.*, 2020; Bettayeb *et al.*, 2019] or device fingerprinting Babun *et al.* [2020].

In the matter of availability and financing, we've found that only four testbeds [Al-Hawawreh and Sitnikova, 2020; Sáez-de Cámara *et al.*, 2023; Koroniotis *et al.*, 2021; Gardiner *et al.*, 2019] may be available for researchers, with 13 testbeds declaring that they had received funding.

7.2 Testbed Architecture

Experiment orchestration is an important aspect of a testbed, and consists of scheduling, executing tests, and collecting and storing data. In light of this, we found that most of the testbeds surveyed do not meet all aspects of this requirement, especially scheduling, which is an important activity for enabling multi-user testbeds. With regard to the execution stage, some testbeds employ scripting, which is sometimes based on NS3 scripts, in order to complete their tests. In this context, it was found that about half of the testbeds employ some level of scripting or automation to carry out the tests.

Resource management is another aspect of testbeds that was examined during our survey, as testbeds make use of multiple end devices and are interconnected by a number of switches, routers and abstraction layers, while some testbeds employ resource management software. However, we found that this resource management is often limited in the surveyed testbeds. This is because it is often confined to just virtualization layers, with some testbeds employing SDN/NFV technology for the interconnecting hosts. With regard to interfacing with the testbed, there is a wide range of approaches, with some works being confined to more manual approaches such as SSH when they carry out experiments and interact with the testbed, while other works use built-in interfaces for managing their systems (Human Machine Interfaces), and some testbeds employ desktop GUI or WebUI applications. Finally, some testbeds only use configuration files and scripts, such as the aforementioned NS3 scripts, when they carry out their experiments. The question of limitations and constraints in the use of testbeds is discussed in Section 8.

7.3 Fidelity requirements

All the examined testbeds use (or are capable of using) physical devices to some degree, with virtualization being a commonly used technology, while containerization and emulation are rarely employed. Sáez-de Cámara *et al.* [2023] is notably the only testbed that does not employ physical devices, although it is likely that it could support these kinds of devices to some degree. On the question of the IoT device representation, most testbeds only use physical devices to model these devices, while Sáez-de Cámara *et al.* [2023] uses containers and, Koroniotis *et al.* [2021] features physical and simulated devices. Finally, the works of Nock *et al.* [2020] and Kumar and Lim [2019] employ emulated devices, the former also using physical devices.

The use of real-world (off-the-shelf) software in a testbed is an important metric, as simulated software may behave differently. Likewise, using real malware samples increases fidelity. However, we found that only some works employ live malware, backdoor or botnet samples [Sáez-de Cámara *et al.*, 2023; Beauchaine *et al.*, 2021; Siboni *et al.*, 2018; Moustafa, 2021; Kumar and Lim, 2019] such as Merlin, Mirai, Vivid, Gafgyt, Reaper, Satori, and most other testbeds only employ tools such as Kali Linux, Nmap, Zmap, SSLStrip, Ethercap, Wireshark and others. However, some testbeds without malware samples may be flexible enough to allow these kinds of scenarios to be explored, thus should not be regarded as a limitation but rather a different focal point in the works surveyed.

Link features play a key role in faithfully representing scenarios, for example, flaky or faulty connections with packet loss (Wi-Fi/LTE) or increased latency (Satelite). Concerning this component, only Sáez-de Cámara *et al.* [2023] features configurable bandwidth and latency modeling by using the `tc` Linux command line utility. It should be noted that the works of Moustafa [2021]; Oliver *et al.* [2018]; Thom *et al.* [2021] feature SDN/NFV technology and may also meet this requirement for variable link characteristics. However, this feature was not demonstrated. Thus, we consider such works as uncertain concerning this requirement.

Non-flat network topologies, which involve more than one broadcast domain are an interesting aspect of a testbed, since real-world networks such as corporate networks involve multiple layers of isolation including firewalls, DMZs, VLANs and other such security features. For this reason, a realistic testbed should include these features so that they can model modern network deployments. In fact, the works by Sáez-de Cámara *et al.* [2023]; Beauchaine *et al.* [2021]; Kumar and Lim [2019]; Lee *et al.* [2021]; Thom *et al.* [2021]; Nock *et al.* [2020]; Babun *et al.* [2020] employ networks with more than one broadcast domain, which was used as a basic requirement for more complex networks.

Background network traffic adds realism to a testbed, as real-world cyber infrastructure features many ancillary services carrying out regular tasks such as IP assignment (DHCP), data exchanges between devices (MQTT, HTTP) or event-based actions such as sensor activations. Within this context, the testbeds of Al-Hawawreh and Sitnikova [2020]; Sáez-de Cámara *et al.* [2023]; Beauchaine *et al.* [2021]; Moustafa [2021] feature some level of background traffic beyond attacks, while the testbeds of Siboni *et al.* [2018]; Waraga *et al.* [2020]; Bettayeb *et al.* [2019]; Babun *et al.* [2020] are not concerned with background traffic as these testbeds focus on analyzing a single IoT device at a time. Finally, Lee *et al.* [2021] features replayed traffic instead of live network captures, and the work by Koroniotis *et al.* [2021] seems to only feature HTTP and MQTT as background traffic.

7.4 Heterogeneity requirements

Wired and Wireless connections are used in real-world IoT deployments, and these connections can suffer from interference, drop packets, induced jitter, limited throughput, and any number of other limitations and characteristics. Given the diverse nature of the current networks, a representative testbed should model this kind of diversity. For example, the works of Al-Hawawreh and Sitnikova [2020]; Beauchaine *et al.* [2021]; Koroniotis *et al.* [2021]; Moustafa [2021]; Kumar and Lim [2019]; Lee *et al.* [2021]; Siboni *et al.* [2018]; Thom *et al.* [2021]; Waraga *et al.* [2020]; Bettayeb *et al.* [2019]; Oliver *et al.* [2018] include Wi-Fi, (Koroniotis *et al.* [2021] being a notable example as it includes the most diverse set of link-layer protocols, including Ethernet, Wi-Fi, Bluetooth, Zigbee, LoRa and Z-Wave). Another testbed that includes Z-Wave and Zigbee protocols is Babun *et al.* [2020]. Other notable works include Gardiner *et al.* [2019] and Oliver *et al.* [2018] as the only testbeds to feature 4G/mobile connectivity.

Application level protocols vary between the purpose and scope of the testbeds, with the IIoT testbeds of Al-Hawawreh and Sitnikova [2020]; Koroniotis *et al.* [2021]; Gardiner *et al.* [2019]; Lee *et al.* [2018]; Thom *et al.* [2021] featuring industrial protocols such as MMS, GOOSE, PLC, MODBUS among others. Other testbeds (Sáez-de Cámara *et al.* [2023]; Beauchaine *et al.* [2021]; Koroniotis *et al.* [2021]; Moustafa [2021]; Kumar and Lim [2019]; Waraga *et al.* [2020]) focused on more general IoT deployments and employ protocols such as MQTT, CoAP, RSTP, IMAP, DNS and NTP. Some testbeds like those of Gardiner *et al.* [2019]; Oliver *et al.* [2018] do not state which application-layer protocols are used, while Bettayeb *et al.* [2019] investigates custom IoT protocols. In particular, the testbed described in Babun *et al.* [2020] is not concerned with application layer protocols, but instead focuses on metadata such as the inter-arrival times of packets. Finally, concerning transport layer protocols, most testbeds employ TCP and UDP, although some testbeds do not disclose which protocols are used.

As regards heterogeneity in attacks, generally speaking, testbeds featured between 1 and 16 distinct attack classes. The work by Waraga *et al.* [2020] includes the most types of attacks, although, some of them are non-standard such as reverse engineering through physical access to serial ports. Some commonly used attack types are DDoS/DoS [Sáez-de Cámara *et al.*, 2023; Moustafa, 2021; Kumar and Lim, 2019; Koroniotis *et al.*, 2021; Beauchaine *et al.*, 2021; Oliver *et al.*, 2018; Al-Hawawreh and Sitnikova, 2020], MiTM ([Nock *et al.*, 2020; Moustafa, 2021; Al-Hawawreh and Sitnikova, 2020]), ARP Spoofing (Bettayeb *et al.* [2019]; Al-Hawawreh and Sitnikova [2020]), network scans (Bettayeb *et al.* [2019];

Table 4. Comparison of Fidelity Features

Publication	**Fidelity**			
	Device Types	**Complex Topologies**	**Background Traffic**	**Sensor devices**
Al-Hawawreh and Sitnikova [2020]	Phy, Virt	No	Realistic	Phy
Sáez-de Cámara *et al*. [2023]	Virt, Container	Yes	Realistic	Containers
Beauchaine *et al*. [2021]	Phy	Yes	Realistic	Phy
Koroniotis *et al*. [2021]	Phy, Virt	No	HTTP & MQTT only	Phy, Sim
Siboni *et al*. [2018]	Phy	No	Not applicable	Phy
Moustafa [2021]	Phy, Virt	No	Realistic	Phy
Gardiner *et al*. [2019]	Phy, Virt	Uncertain	Unknown	Phy
Kumar and Lim [2019]	Phy, Virt	Yes	Unknown	Emu/Virt
Oliver *et al*. [2018]	Phy	Uncertain	Unknown	Phy
Lee *et al*. [2021]	Phy	Yes	Artificial	Phy
Lee *et al*. [2018]	Phy	No	Uncertain	Phy
Thom *et al*. [2021]	Phy, Virt	Yes	Unknown	Phy
Waraga *et al*. [2020]	Phy	No	Not applicable	Phy
Nock *et al*. [2020]	Phy, Emu	Yes (mesh)	Unknown	Phy, Emu
Bettayeb *et al*. [2019]	Phy	No	Not applicable	Phy
Babun *et al*. [2020]	Phy	Yes (mesh)	Realistic	Phy

Phy Physical; $Virt$ Virtualization; Emu Emulation; Sim Simulation

Table 5. Comparison of Aspects of Heterogeneity

Publication	**Heterogeneity**			
	Protocols		**Attack Types**	**Sensors**
	Link-layer	**Application**	**Count**	**N. Types**
Al-Hawawreh and Sitnikova [2020]	Wi-Fi, Ethernet	MODBUS, DNS, SSH, Websockets, DNS, COAP, MQTT, SQL, IMAP, SMTP	Uncertain	3
Sáez-de Cámara *et al*. [2023]	Ethernet	MQTT, CoAP, RSTP, DNS, NTP	13	9*
Beauchaine *et al*. [2021]	Wi-Fi, Ethernet	HTTP, SSH, TELNET	2	3
Koroniotis *et al*. [2021]	Wi-Fi, Bluetooth, Zigbee, Lora, Z-wave, Ethernet	MQTT, HTTP	7	7
Siboni *et al*. [2018]	Ethernet, Wi-Fi, BLE, ZigBee	HTTP, TELNET, SSH	Unknown	Uncertain
Moustafa [2021]	Wi-Fi, Ethernet	HTTP, HTTPS, MQTT	9	7
Gardiner *et al*. [2019]	4G, Ethernet	Unknown	1	Uncertain
Kumar and Lim [2019]	Uncertain	DNS, DHCP, TELNET	2	1
Oliver *et al*. [2018]	5G, 4G, Wi-Fi	Unknown	1	Unknown
Lee *et al*. [2021]	Ethernet, Wi-Fi	Unknown	0	6
Lee *et al*. [2018]	Ethernet	MMS, GOOSE, Modbus, DNP3, GSE, SV, MMS	5	8
Thom *et al*. [2021]	Ethernet, Wi-Fi	PLC, SSH, FTP, Modubs	3	8
Waraga *et al*. [2020]	Wi-Fi	HTTP	16	2
Nock *et al*. [2020]	6LoPWAN, 6TiSCH, RPL, 802.15.4	CoAP	1	1
Bettayeb *et al*. [2019]	Wi-Fi	Custom IoT protocols	2	2
Babun *et al*. [2020]	Zigbee, Z-wave	Custom IoT protocols	0	19

∗ The testbed has 9 groups of sensors, each containing various discrete sensors

Moustafa [2021]; Thom *et al.* [2021]; Sáez-de Cámara *et al.* [2023]) and botnet backdoors (Moustafa [2021]; Beauchaine *et al.* [2021]; Sáez-de Cámara *et al.* [2023]).

Another facet of heterogeneity is the number of diverse attack instances within the same attack class, for example, variants of a DDoS attack. In this context, we found that the works of Sáez-de Cámara *et al.* [2023]; Koroniotis *et al.* [2021]; Kumar and Lim [2019] feature this kind of heterogeneity in at least one attack class, and only Sáez-de Cámara *et al.* [2023] makes configurations of applications heterogeneous.

A testbed that seeks to be representative of a wide range of scenarios must include diversified sensors and IoT devices. Most works use physical sensors/IoT devices such as air quality (temperature, humidity, pressure), light sensors, presence sensors, cameras, gas sensors, cooling engine monitors, hydraulics and household appliances (smart TVs, Amazon Echo, Nest Cams, Philips Hue, among others). However, some testbeds do not specify which sensors are in use [Kumar and Lim, 2019; Oliver *et al.*, 2018]. Lastly note should be taken of Babun *et al.* [2020] who have the testbed with the largest number of different types of devices (19).

IoT devices commonly employ architectures other than AMD64/x86, such as MIPS, ARM and more recently RISCV. Moreover, we have found that the testbeds of Al-Hawawreh and Sitnikova [2020]; Beauchaine *et al.* [2021]; Koroniotis *et al.* [2021]; Moustafa [2021]; Kumar and Lim [2019]; Thom *et al.* [2021] demonstrate the use of these architectures, while some other testbeds such as those of Sáez-de Cámara *et al.* [2023]; Gardiner *et al.* [2019]; Oliver *et al.* [2018]; Lee *et al.* [2021, 2018]; Nock *et al.* [2020]; Babun *et al.* [2020] fail to specify their use. Lastly, since it has been determined that no testbed uses a distributed architecture (as discussed in Subsection 7.5), there are no testbeds among the works surveyed that have a clustering of devices with heterogeneous architectures.

7.5 Scalability requirements

Scalability is an important factor when researching certain types of attacks such as DDoS attacks, botnets and worms. This attribute has multiple dimensions, such as the number of devices modeled in a given experiment, whether the testbed architecture is distributed or local and in the case of testbeds that employ virtualization or clustering, the number of computer nodes that are utilized.

We found that Sáez-de Cámara *et al.* [2023] is the testbed with the most device nodes modeled. This testbed relies on containerization and virtualization to achieve 140 nodes in its topology, which includes several sensors (100x), switches (30x), routers (10x) and a cloud layer. This degree of scalability is to some extent possible because containers efficiently share host resources, such as processor time and memory, and may incur lower overhead than virtualization [Xavier *et al.*, 2013]. Another notable testbed with this metric is the SAir-IIoT [Koroniotis *et al.*, 2021] testbed, which features 82 devices ranging from general environment sensors, to physical access control and drones.

With regard to the testbed architecture, no testbeds were found that have a geographically distributed architecture, although some employ virtualization and clustering which may enable this distribution in conjunction with other techniques for interconnecting hosts. Of the testbeds that employ either clustering or virtualization[Al-Hawawreh and Sitnikova, 2020; Sáez-de Cámara *et al.*, 2023; Beauchaine *et al.*, 2021; Moustafa, 2021; Kumar and Lim, 2019; Thom *et al.*, 2021], the work Al-Hawawreh and Sitnikova [2020] employs the largest number of computer nodes (3), with most other works employing a single computer node or in the case of Kumar and Lim [2019] the number of nodes in the cluster is unknown. It should also be noted that only testbeds that employ virtualization or clustering were considered for this requirement.

7.6 Security requirements

According to Sáez-de Cámara *et al.* [2023] security testbeds have extra requirements for isolation and the safe execution of malware. Our comparison in this area centers on two key factors: whether testbeds are isolated or otherwise firewalled to the Internet and whether the experiments are isolated from each other, and can allow experiments to be conducted at the same time.

A unique security feature is a shielded room that was noted in Siboni *et al.* [2018], although the testbed does not deny the devices' access to the internet. Other testbeds such as Al-Hawawreh and Sitnikova [2020]; Sáez-de Cámara *et al.* [2023]; Koroniotis *et al.* [2021]; Gardiner *et al.* [2019]; Kumar and Lim [2019] isolate their experiments from the Internet, while the works of Oliver *et al.* [2018]; Lee *et al.* [2021, 2018]; Waraga *et al.* [2020]; Nock *et al.* [2020]; Bettayeb *et al.* [2019] do not state whether their testbed ensures this kind of isolation. TON_IoT [Moustafa, 2021] has network function virtualization and SDN capabilities, although, it is unclear whether these features were used to ensure isolation. The works of Beauchaine *et al.* [2021] and Babun *et al.* [2020] are other similar examples with unclear network isolation features. On the question of isolation between the experiments, most testbeds do not offer this feature; the work of Sáez-de Cámara *et al.* [2023] may inherit such a feature from GNS3 while Kumar and Lim [2019] may acquire it from its virtualization architecture. However, this functionality was not described in detail.

7.7 Reproducibility requirements

Reproducibility is a key aspect of research and can be achieved in actions such as the following: documenting network node topology, standardizing attack and application tools or scripting events so that they can be executed in a repeatable manner. We found that the works of Sáez-de Cámara *et al.* [2023]; Kumar and Lim [2019] have their topology defined by scripts, while the works of Sáez-de Cámara *et al.* [2023]; Koroniotis *et al.* [2021] and Kumar and Lim [2019] feature reproducible applications and software in a similar fashion. The works of Sáez-de Cámara *et al.* [2023]; Kumar and Lim [2019]; Waraga *et al.* [2020]; Nock *et al.* [2020]; Bettayeb *et al.* [2019] feature scripted or otherwise reproducible attacks. With regard to application reproducibility, we found that Al-Hawawreh and Sit-

Table 6. A comparison of Scalability, Security and Measurability features

Publication	Scalability		Security	Measurability
	No. devices	No. Compute nodes	Internet Isolation	Artifacts Logged
Al-Hawawreh and Sitnikova [2020]	14	3	Yes	Pkt.; Sensor
Sáez-de Cámara *et al*. [2023]	140	1	Yes	Pkt.; App.
Beauchaine *et al*. [2021]	13	1	Uncertain	Pkt.; Sensor
Koroniotis *et al*. [2021]	82	Unknown	Yes	Pkt.; Sensor
Siboni *et al*. [2018]	Unknown	Not applicable	No	Pkt.
Moustafa [2021]	17	Unknown	Unknown	Pkt.; App.; Sensor
Gardiner *et al*. [2019]	Unknown	Unknown	Yes	Pkt.; Sensor
Kumar and Lim [2019]	53	Unknown	Yes	Pkt.
Oliver *et al*. [2018]	Unknown	Unknown	Unknown	Unknown
Lee *et al*. [2021]	27	Not applicable	Unknown	Pkt.
Lee *et al*. [2018]	16	Not applicable	Unknown	Pkt.; App.
Thom *et al*. [2021]	42	Unknown	Partial^A	Pkt.; App.
Waraga *et al*. [2020]	2	Not applicable	Unknown	Pkt.
Nock *et al*. [2020]	20	Not applicable	Unknown	App.
Bettayeb *et al*. [2019]	4	Not applicable	Unknown	Pkt.
Babun *et al*. [2020]	39	Not applicable	Unknown	Pkt.

A Uses a firewall to isolate the networks, but is not completely isolated; Pkt Packet Captures; $App.$ Application Logs; $Sensor$ Sensor Telemetry

nikova [2020]; Sáez-de Cámara *et al.* [2023]; Koroniotis *et al.* [2021]; Kumar and Lim [2019] feature reproducible applications by means of scripting or containerization. Likewise, Sáez-de Cámara *et al.* [2023]; Kumar and Lim [2019]; Waraga *et al.* [2020]; Nock *et al.* [2020]; Bettayeb *et al.* [2019] feature reproducible attacks in the same manner.

7.8 Flexibility requirements

Experiments should be adaptable so that researchers can explore other scenarios with distinct parameters. These parameters may come in many different forms, such as a change of topologies, and by adding or removing devices, among others. In light of this, we documented these characteristics across the surveyed testbeds, and found them in the works of Al-Hawawreh and Sitnikova [2020]; Koroniotis *et al.* [2021]; Moustafa [2021]; Gardiner *et al.* [2019]; Kumar and Lim [2019]; Thom *et al.* [2021]; Babun *et al.* [2020] make it possible to add or remove either physical or virtualized nodes, while Sáez-de Cámara *et al.* [2023] provides node flexibility using virtualization and containers. Nock *et al.* [2020] uses a mix of physical and emulated devices to modify experimental node counts. Lastly, this metric is not applicable to the works of Siboni *et al.* [2018]; Waraga *et al.* [2020]; Bettayeb *et al.* [2019] as these do not focus on testing multiple devices at the same time.

On the question of network topology flexibility, manually assembling a network infrastructure for a single experiment is the most common method observed However, this may prove too costly or slow to carry out experiments to scale and to address this, some testbeds employ automation in the form of scripting or configuration files [Sáez-de Cámara *et al.*, 2023; Kumar and Lim, 2019; Nock *et al.*, 2020] or SDN/NFV [Oliver *et al.*, 2018; Thom *et al.*, 2021] to create multiple topologies. Moustafa [2021] partly creates its topology with SDN and partly with physical infrastructure.

7.9 Measurability requirements

Data is a core aspect of research, and in the field of testbeds, this data may come in many different forms such as network traffic captures, application logs or sensor telemetry. The use cases of these data sources vary, for example in the case of anomaly detection or intrusion detection and prevention. Thus, the testbeds can be classified in accordance with the data each one collects. Most testbeds feature some level of network capture or logging, usually in the form of packet captures such as `.pcapng` files. However, application-level logs and sensor telemetry captures are not always available. For instance, only the works Sáez-de Cámara *et al.* [2023]; Moustafa [2021]; Lee *et al.* [2018]; Thom *et al.* [2021]; Nock *et al.* [2020] feature application logs, and the works Al-Hawawreh and Sitnikova [2020]; Beauchaine *et al.* [2021]; Koroniotis *et al.* [2021]; Moustafa [2021]; Gardiner *et al.* [2019] feature sensor telemetry data.

8 Final remarks

In light of the comparisons made above, this section is devoted to carrying out a meta-analysis of the works discussed. First, we note that 12 out of the 16 testbeds considered are not available for public testing or else this information is not disclosed. This can be seen as a serious barrier to reproducibility, as without a testbed available, researchers are unable to carry out their experiments in the same environment without also building (reproducing) the entire testbed, which is a time-consuming and resource-intensive task.

[Sáez-de Cámara *et al.*, 2023] furthers this point and notes that although many datasets from emulated testbeds are available, the testbeds themselves are rarely published. The author also states that sharing machine learning datasets alone is not enough to narrow the gap between experimental and deployment environments, and argues in favor of a reproducible and extendable testbed that allows researchers to create scenarios that more accurately reflect real environments.

Testbeds that mainly rely on physical devices [Beauchaine *et al.*, 2021; Siboni *et al.*, 2018; Oliver *et al.*, 2018; Lee *et al.*, 2021, 2018; Waraga *et al.*, 2020; Bettayeb *et al.*, 2019; Babun *et al.*, 2020] have some scalability constraints, as adding more physical devices to a testbed often involves manual work involving the infrastructure, while virtualized and containerized testbeds can take advantage of modern cloud architecture, by instantiating nodes as required via a pre-existing infrastructure and abstraction layer. The flexibility constraints of physical-only devices are also evident, in particular those concerning the creation of new network topologies, as the network topologies used in these testbeds are often also manually configured and assembled. Lastly, it should be noted that the overwhelming majority of testbeds opted to use physical devices to represent their IoT/sensors, which inherits the scalability and flexibility constraints previously discussed.

With regard to fidelity factors, it was clear that most of the surveyed testbeds do not provide configurable network links, and feature characteristics such as configurable latency, bandwidth shaping, and packet loss emulation. The sole exception to this observation is Sáez-de Cámara *et al.* [2023], although, some testbeds employ SDN/NFV [Moustafa, 2021; Oliver *et al.*, 2018; Thom *et al.*, 2021] technologies, which can provide much of the required tooling to emulate variable link qualities; nonetheless, these features were not demonstrated in those works.

Another important fidelity factor that is often absent in some testbeds, is the ability to create complex network topologies involving multiple broadcast zones. This may be achieved via physical (hard-coded) infrastructure such as in Beauchaine *et al.* [2021]; Lee *et al.* [2021] or software-based solutions such as those demonstrated in Sáez-de Cámara *et al.* [2023]. We also found that Thom *et al.* [2021] uses both software-based and hardware-based network topologies, including multiple broadcast zones on each. However, we also discovered that many testbeds lack this fidelity characteristic, and thus cannot accurately represent scenarios such as enterprise networks, which often involve multiple VLANs, firewalls, and other systems to secure and isolate office workstations and servers.

When compared with the previously discussed testbeds, some limitations could be observed in these works. First, it should be pointed out the fidelity constraints observed in the Brown-IIoT testbed [Al-Hawawreh and Sitnikova, 2020], which shows that they cannot demonstrate that the developed testbed behavior is similar to that found in real-world systems. This limitation is at the core of many testbeds, since they are rarely compared with real-world systems, especially given the scope of the Brown-IIoT testbed. This work also reveals another limitation, regarding scalability and fidelity trade-offs, as the use of physical IoT devices such as sensors, increases realism; however, it also introduces scalability issues since the IoT gateways can only tolerate a certain number of connected devices. By comparison, the Gotham testbed [Sáez-de Cámara *et al.*, 2023] balances this trade-off more on the side of scalability, by providing containerized sensors and applications that mimic real-world devices, even though these may not possess all the features of physical devices. For example, the Gotham testbed lacks the ability to model wireless protocols and mobility features, which may be important factors in some research areas. The Gotham testbed also brings to light some other important limitations, such as the re-usability of created scenarios, since there is some coupling between the configuration files. Furthermore, another constraint of the testbed is related to security, as Gotham's backend executes containers in privileged mode by default, which may pose security risks for the testbed. Z-IoT [Babun *et al.*, 2020] notes that it does not take into account clones from authorized devices or authorized devices that have been compromised. Thus, this work focuses on fingerprinting unauthorized devices that attempt to spoof those that are authorized.

Our last remark drew attention to some trends in the cybersecurity testbed landscape. Hybrid testbeds involving a mix of physical and virtual devices are common, since they provide the flexibility, scalability and resource management of virtualization, together with the fidelity and heterogeneity of physical devices. Moreover, a clear trend has been found in the use of IDS solutions based on machine-learning models built from testbed datasets, which may increase accuracy and thus improve the security of systems where such a solution is employed. We also found that many testbeds employ a hybrid architecture, which relies on both virtualization and physical devices to balance scalability and fidelity, with the use of containerization also showing great scalability potential. Lastly, the softwareization of networks (SDN) in testbeds is an emerging phenomenon that should enable greater flexibility in the design and modification of testbed scenarios.

9 Conclusion

This paper establishes a comprehensive taxonomy for IoT cybersecurity testbeds, by addressing questions of fidelity, heterogeneity, scalability, security, reproducibility, flexibility, and measurability. Through a rigorous systematic literature review, we identified and analyzed 16 prominent testbeds from an initial pool of 824 articles. These testbeds were thoroughly examined, and their main features were summarized and compared. Our findings are that the current testbeds have diverse and sometimes overlapping focal points and features. This variability makes some testbeds highly specialized and suitable for research in fields such as industrial networks, while other testbeds have a more generalized approach, since they are able to accommodate a broader range of scenarios. We also found that some areas such as SDN and NFV have a great potential for adding flexibility to testbeds, while LTE (mobile) connectivity is still an area that needs research, as most testbeds do not include it even though 5G networks have become more commonplace in infrastructure and industry. Lastly, we found that although some testbeds reach more than a hundred devices, this is still unsuitable for modeling scenarios where hundreds of devices co-exist, as can be seen in other testbeds like DeterLab [Wroclawski *et al.*, 2016].

Moving forward, we plan to continually monitor advances in the field and expand our review to track the influence of these testbeds in future literature reviews. Additionally, we aim to conduct bibliometric analyses to reveal trends and correlations within the surveyed works, including insights into key academic centers and collaborative networks. Finally, we also intend to create an IoT cybersecurity testbed, by applying the lessons learned from the surveyed works to ensure it is a novel, realistic and scalable testbed.

Declarations

Acknowledgements

This work was partially supported by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) - Grant 88887.808340/2023-00, *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP) - Grant #2018/23098-0 and *Rede Nacional de Ensino e Pesquisa* (RNP).

Authors' Contributions

KS was responsible for drafting the manuscript, conducting the systematic literature review, and analyzing and classifying the selected articles in accordance with the established taxonomy.

MS critically reviewed the manuscript for clarity, coherence, and structural integrity, and provided expert guidance on the subject of testbeds. Additionally, MS contributed to the revision process.

MW designed the systematic literature review process, oversaw its implementation, and reviewed articles identified as borderline. Additionally, MW contributed to defining the overall structure of the manuscript, revising its content, and assisting in the writing process.

All authors have read and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

References

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., *et al.* (2015). Fit iot-lab: A large scale open experimental iot testbed. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 459–464. IEEE. DOI: 10.1109/WF-IoT.2015.7389098.

Agrawal, N. and Kumar, R. (2022). Security perspective analysis of industrial cyber physical systems (i-cps): A decade-wide survey. *ISA transactions*, 130:10–24. DOI: 10.1016/j.isatra.2022.03.018.

Al-Hawawreh, M. and Sitnikova, E. (2020). Developing a security testbed for industrial internet of things. *IEEE Internet of Things Journal*, 8(7):5558–5573. DOI: 10.1109/JIOT.2020.3032093.

Babun, L., Aksu, H., Ryan, L., Akkaya, K., Bentley, E. S., and Uluagac, A. S. (2020). Z-iot: Passive device-class fingerprinting of zigbee and z-wave iot devices. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE. DOI: 10.1109/ICC40277.2020.9149285.

Baldin, I., Nikolich, A., Griffioen, J., Monga, I. I. S., Wang, K.-C., Lehman, T., and Ruth, P. (2019). Fabric: A national-scale programmable experimental network infrastructure. *IEEE Internet Computing*, 23(6):38–47. DOI: 10.1109/MIC.2019.2958545.

Beauchaine, A., Macchiaroli, M., and Yun, M. (2021). ibot: Iot botnet testbed. In *2021 16th International Conference on Computer Science & Education (ICCSE)*, pages 822–827. IEEE. DOI: 10.1109/ICCSE51940.2021.9569298.

Bettayeb, M., Waraga, O. A., Talib, M. A., Nasir, Q., and Einea, O. (2019). Iot testbed security: Smart socket and smart thermostat. In *2019 IEEE Conference on Application, Information and Network Security (AINS)*, pages 18–23. IEEE. DOI: 10.1109/AINS47559.2019.8968694.

Cappos, J., Hemmings, M., McGeer, R., Rafetseder, A., and Ricart, G. (2018). Edgenet: a global cloud that spreads by local action. In *ACM Symposium on Edge Computing (SEC)*, pages 359–360. DOI: 10.1109/SEC.2018.00045.

Chernyshev, M., Baig, Z., Bello, O., and Zeadally, S. (2017). Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647. DOI: 10.1109/JIOT.2017.2786639.

Cintuglu, M. H., Mohammed, O. A., Akkaya, K., and Uluagac, A. S. (2016). A survey on smart grid cyber-physical system testbeds. *IEEE Communications Surveys & Tutorials*, 19(1):446–464. DOI: 10.1109/COMST.2016.2627399.

Conti, M., Donadel, D., and Turrin, F. (2021). A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials*, 23(4):2248–2294. DOI: 10.1109/COMST.2021.3094360.

Demeester, P., Van Daele, P., Wauters, T., and Hrasnica, H. (2022). Fed4fire–the largest federation of testbeds in europe. In *Building the future internet through FIRE*, pages 87–109. River Publishers. Available at:`https://www.taylorfrancis.com/chapters/oa-edit/10.1201/9781003337447-5/fed4fire-largest-federation-testbeds-europe-piet-demeester-peter-van-daele-tim-wauters-halid-hrasnica`.

Gardiner, J., Craggs, B., Green, B., and Rashid, A. (2019). Oops i did it again: Further adventures in the land of ics security testbeds. In *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, pages 75–86. DOI: 10.1145/3338499.3357355.

Gomez, J., Kfoury, E. F., Crichigno, J., and Srivastava, G. (2023). A survey on network simulators, emulators, and testbeds used for research and education. *Computer Networks*, 237:110054. DOI: 10.1016/j.comnet.2023.110054.

Grossmann, J. and Duponchelle, J. (2008). Graphical network simulator-3. Available at:`https://gns3.com/`. Accessed in: 20-02-2024.

Hemminger, S. *et al.* (2005). Network emulation with netem. In *Linux conf au*, volume 5, page 2005. Available at:`https://www.rationali.st/blog/files/20151126-jittertrap/netem-shemminger.pdf`.

Henderson, T. R., Lacage, M., Riley, G. F., Dowell, C., and Kopena, J. (2008). Network simulations with the ns-3 simulator. *SIGCOMM demonstration*, 14(14):527. Available at:`https://conferences.sigcomm.org/sigcomm/2008/papers/p527-hendersonA.pdf`.

IBM (2024). Cost of a data breach 2023 | ibm — ibm.com. Available at:`https://www.ibm.com/reports/data-breach`. Accessed in: 15-02-2024.

Imperva (2023). 2023 Imperva Bad Bot Report | Resource Library — imperva.com. Available at:`https://www.imperva.com/resources/resource-library/reports/2023-imperva-bad-bot-report/`. Accessed in: Accessed 15-02-2024.

Kampourakis, V., Gkioulos, V., and Katsikas, S. (2023). A systematic literature review on wireless security testbeds in the cyber-physical realm. *Computers & Security*, page 103383. DOI: 10.1016/j.cose.2023.103383.

Kitchenham, B. (2007). *Guidelines for performing Systematic Literature Reviews in software engineering. EBSE Technical Report EBSE-2007-01*. Technical report, EBSE Technical Report EBSE-2007-01. Book.

Koroniotis, N., Moustafa, N., Schiliro, F., Gauravaram, P., and Janicke, H. (2021). The sair-iiot cyber testbed as a service: A novel cybertwins architecture in iiot-based smart airports. *IEEE Transactions on Intelligent Transportation Systems*. DOI: 10.1109/TITS.2021.3106378.

Kumar, A. and Lim, T. J. (2019). A secure contained testbed for analyzing iot botnets. In *Testbeds and Research Infrastructures for the Development of Networks and Communities: 13th EAI International Conference, TridentCom 2018, Shanghai, China, December 1-3, 2018, Proceedings 13*, pages 124–137. Springer. DOI: 10.1007/978-3-030-12971-2_8.

Lee, G., Lee, J., Kim, Y., and Park, J.-G. (2021). Network flow data re-collecting approach using 5g testbed for labeled dataset. In *2021 23rd International Conference on Advanced Communication Technology (ICACT)*, pages 254–258. IEEE. DOI: 10.23919/ICACT51234.2021.9370561.

Lee, S., Lee, S., Yoo, H., Kwon, S., and Shon, T. (2018). Design and implementation of cybersecurity testbed for industrial iot systems. *The Journal of Supercomputing*, 74:4506–4520. DOI: 10.1007/s11227-017-2219-z.

Lochin, E., Perennou, T., and Dairaine, L. (2012). When should i use network emulation? *annals of telecommunications-annales des télécommunications*, 67:247–255. DOI: 10.1007/s12243-011-0268-5.

Mirkovic, J. and Benzel, T. (2012). Teaching cybersecurity with deterlab. *IEEE Security & Privacy*, 10(1):73–76. DOI: 10.1109/MSP.2012.23.

Moustafa, N. (2021). A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. *Sustainable Cities and Society*, 72:102994. DOI: 10.1016/j.scs.2021.102994.

Nock, O., Starkey, J., and Angelopoulos, C. M. (2020). Addressing the security gap in iot: towards an iot cyber range. *Sensors*, 20(18):5439. DOI: 10.3390/s20185439.

Oliver, I., Kalliola, A., Holtmanns, S., Miche, Y., Limonta, G., Vigmostad, B., and Muller, K. (2018). A testbed for trusted telecommunications systems in a safety critical environment. In *Computer Safety, Reliability, and Security: SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings 37*, pages 87–98. Springer. DOI: 10.1007/978-3-319-99229-7_9.

OWASP (2018). IoT Top 10. Technical report, OWSAP. Available at:`https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project`.

Peterson, L. and Culler, D. (2002). PlanetLab | An open platform for developing, deploying, and accessing planetary-scale services. Available at:`http://www.planet-lab.org/`.

Rampfl, S. (2013). Network simulation and its limitations. In *Proceeding zum seminar future internet (FI), Innovative Internet Technologien und Mobilkommunikation (IITM) und autonomous communication networks (ACN)*, volume 57. Citeseer. DOI: 10.2313/NET-2013-08-$1_0$8.

Rizzo, L. (1997). Dummynet: a simple approach to the evaluation of network protocols. *ACM SIGCOMM Computer Communication Review*, 27(1):31–41. DOI: 10.1145/251007.251012.

Sáez-de Cámara, X., Flores, J. L., Arellano, C., Urbieta, A., and Zurutuza, U. (2023). Gotham testbed: a reproducible iot testbed for security experiments and dataset generation. *IEEE Transactions on Dependable and Secure Computing*. DOI: 10.1109/TDSC.2023.3247166.

Siaterlis, C., Garcia, A. P., and Genge, B. (2012). On the use of emulab testbeds for scientifically rigorous experiments. *IEEE Communications Surveys & Tutorials*, 15(2):929–942. DOI: 10.1109/SURV.2012.0601112.00185.

Siaterlis, C., Genge, B., and Hohenadel, M. (2013). Epic: A testbed for scientifically rigorous cyber-physical security experimentation. *IEEE Transactions on Emerging Topics in Computing*, 1(2):319–330. DOI: 10.1109/TETC.2013.2287188.

Siboni, S., Sachidananda, V., Meidan, Y., Bohadana, M., Mathov, Y., Bhairav, S., Shabtai, A., and Elovici, Y. (2018). Security testbed for internet-of-things devices. *IEEE transactions on reliability*, 68(1):23–44. DOI: 10.1109/TR.2018.2864536.

Thom, J., Das, T., Shrestha, B., Sengupta, S., and Arslan, E. (2021). Casting a wide net: An internet of things testbed for cybersecurity education and research. In *2021 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pages 1–8. IEEE. DOI: 10.23919/SPECTS52716.2021.9639278.

Ukwandu, E., Farah, M. A. B., Hindy, H., Brosset, D., Kavallieros, D., Atkinson, R., Tachtatzis, C., Bures, M., Andonovic, I., and Bellekens, X. (2020). A review of cyber-ranges and test-beds: Current and future trends. *Sensors*, 20(24):7148. DOI: 10.3390/s20247148.

University of Utah and Flux Research Group (2024). Emulab.Net - Bibliography. Available at:`http://www.emulab.net/expubs.php`. Accessed in: 20-02-2024.

Veksler, V. D., Buchler, N., Hoffman, B. E., Cassenti, D. N., Sample, C., and Sugrim, S. (2018). Simulations in cyber-security: a review of cognitive modeling of network attackers, defenders, and users. *Frontiers in psychology*, 9:691. DOI: 10.3389/fpsyg.2018.00691.

Waraga, O. A., Bettayeb, M., Nasir, Q., and Talib, M. A. (2020). Design and implementation of automated iot security testbed. *Computers & security*, 88:101648. DOI: 10.1016/j.cose.2019.101648.

Wroclawski, J., Benzel, T., Blythe, J., Faber, T., Hussain, A., Mirkovic, J., and Schwab, S. (2016). Deterlab and the deter project. *The GENI Book*, pages 35–62. DOI: 10.1007/978-3-319-33769-2$_3$.

Xavier, M. G., Neves, M. V., Rossi, F. D., Ferreto, T. C., Lange, T., and De Rose, C. A. (2013). Performance evaluation of container-based virtualization for high performance computing environments. In *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pages 233–240. IEEE. DOI: 10.1109/PDP.2013.41.

Appendix A Questionnaire

Number	Question	Application Instructions
	Article Metadata	
Meta.1	Title	Full article title
Meta.2	Authors	All article authors
Meta.3	Year	Year of publication
Meta.4	Abstract	Article's abstract
Meta.5	Keywords	Article's keywords
Meta.6	DOI	Article's DOI identifier
	General Questions	
Gen.1	Testbed purpose: what use cases is the testbed intended for? Categories: Vulnerability Analysis, Industrial and Telecom use, Dataset Generation, Device Fingerprinting	Classify the testbed according to its primary purpose. If the testbed fulfills multiple roles, then classify it as a multipurpose testbed.
Gen.2	What are the target audiences, and expected results?	Does the testbed focus on providing a lab environment for teachers or some other well-defined group? And what results are expected by targeting that group?
Gen.3	What are the major contributions of the testbed?	Does the testbed provide any major contributions or distinguishing factors when compared to other similar works in the literature?
Gen.4	Is the testbed still available?	Does the article in question imply that the testbed is available for other researchers?
Gen.5	Was funding provided for the testbed? If so, by whom?	Whether the article enumerates funding sources such as grants.
	Testbed architecture	
Arch.1	What are the general testbed characteristics regarding (i) experiment orchestration (experiment execution controller), (ii) resource management, and (iii) user interaction?	(i) Experiment orchestration is the ability to create, schedule, start, stop, and otherwise manage the lifecycle of an experiment beyond manual means. For example, batch executions of experiment definitions. (ii) Whether the testbed employs some level of resource management, such as allocating resources for a specific experiment, using slicing, virtualization, or containerization to effectively divide or multiplex resources. (iii) The means through which the user interacts with the testbed, for example, by using a web interface, manually executing scripts inside SSH sessions, through some GUI, TUI, or any other kind of interface.
	Fidelity	
Fid.1	How does the testbed represent its devices in experiments? (Physical, Virtualization, Emulation, Containerization)	Whether the testbed uses one of more of the following options: Physical devices, Virtualization, Emulation, Containerization
Fid.2	Does the testbed use off-the-shelf software such as Apache Web Server or malware like Mirai, ZeuS, etc?	Certain testbeds utilize purpose-built scripts or applications that mimic the operation of off-the-shelf software. For example, a testbed may utilize an off-the-shelf HTTP server such as Nginx or a custom-built Python script that behaves as an HTTP server. This is also applicable to malware. Does the testbed employ real malware strains such as Mirai, ZeuS, etc.?

Fid.3	Does the testbed represent link characteristics such as (i) bandwidth, (ii) jitter, (iii) packet loss?	Whether the connections between devices can be controlled or otherwise defined by the researcher. For example, one experimenter might want to observe certain phenomena on devices connected to a 3G network, by emulating the bandwidth, jitter, and packet loss of the connection to mimic the desired connection.
Fid.4	Does the testbed feature multiple broadcast domains, that represent a more realistic and complex network?	Multiple broadcast domains are a critical requirement for emulating networks beyond a flat topology, such as a single home's LAN. For example, corporate networks may utilize complex hierarchical topologies with multiple broadcast zones. To apply this criterion, the testbed topology should be analyzed. Topologies containing multiple routers are considered to contain multiple Layer 2 broadcast domains.
Fid.5	Does the testbed incorporate background network traffic, and if so, what is its nature? This includes a) assessing whether the traffic is artificial (replayed) or realistic and b) evaluating the heterogeneity, which may range from (i) no background traffic, (ii) homogeneous/single protocol, or (iii) diversified/heterogeneous protocols.	Background traffic is classified as any traffic beyond attacks such as brute force attempts or network scans. This background traffic may be homogeneous such as HTTP clients performing periodic requests, or it may be heterogeneous, containing more than three protocols, such as DHCP, DNS, SMTP, NTP, and more. Regarding whether the traffic is artificial, it should be considered whether the origin of such traffic is from replaying a packet capture. If the device is running the software for that protocol (for example, a DNS client crafting and sending a DNS query), then the traffic is not artificial.
Fid.6	How does the testbed model the sensors/IoT devices?	Whether the testbed uses one or more of the following types to represent IoT devices: Physical devices, Virtualization, Emulation, Containerization
	Heterogeneity	
Het.1	What types of network protocols are employed? - in particular, regarding (i) link layer, (ii) transport layer, and (iii) application layer protocols?	Given the OSI model, enumerate which protocols are used for the link, transport, and application layers. For example: (i) Ethernet, Wi-Fi, Bluetooth, Zigbee, Lora, Z-Wave, (ii) TCP, UDP, QUIC, (iii) HTTP, SMTP, DNS, DHCP, MQTT, TELNET
Het.2	Does the testbed model have multiple types of attacks?	Distinct types of attacks such as DDoS attacks, worm propagation, ransomware, data exfiltration, man-in-the-middle
Het.3	Are there varied attack-type instances among them? How heterogeneous are these attacks? (i) No variation (ii) multiple attack instances with distinct parameters.	For example, consider a DDoS attack against a target host, it may be homogenous such as an HTTP Flood, or it may be heterogeneous, including GRE, TCP SYN, Slowloris, and others. Likewise, each of these attacks may have some degree of configurability, such as the rate of the packets being sent, the size of the payload, etc.
Het.4	Are the application configurations diverse, such as web servers with or without encryption, different authentication protocols, etc?	Analyze whether an application in the testbed (web server, database, file server, etc) has multiple instances with distinct parameters. For example, a web server with and without encryption, a persistent and a non-persistent CoAP connection, a file server with and without authentication, etc.

Het.5	How many types of IoT devices/sensors are used? What are these types?	The types (models) of IoT/sensor devices in use. For example, there may be two distinct types of a network security camera, and for each of those types, multiple instances of such an model may exist. In this question, we are concerned with the types only, as the number of devices is a Scalability aspect. Example of application: 10x Vendor-A Model Foo Cameras + 10x Vendor-B Model Bar Cameras + 10x Vendor-B Model Baz equals 3 types.
Het.6	In the case of a distributed architecture, does the testbed support the execution with heterogeneous processor architectures?	Does the testbed feature a distributed architecture, and if so, does it feature devices with distinct processor architectures? For example, a testbed featuring AArch64 + AMD64 + RISC-V support.
Het.7	Does the testbed demonstrate that it has connected devices of distinct processor architectures (ARM vs. x86, etc.)?	Whether the testbed demonstrates the use of multiple processor architectures. If such information is unclear or unknown, describe it as such.
Scalability		
Sca.1	How many devices are represented in the testbed in total, including non-IoT devices?	The number of instances of devices in the testbed. From end systems such as servers, laptops, sensors, to network infrastructure such as routers, switches, firewalls, etc.
Sca.2	How many compute nodes are used in the testbed?	This question is only applicable to testbed that employ virtualization or containerization. Does the testbed use more than one node acting as a hypervisor/container host? For example, a testbed built upon a 3-node Proxmox cluster has 3 compute nodes.
Sca.3	Is the testbed architecture locally or geographically distributed?	Whether the testbed has devices distributed geographically outside a single room. For example, sensors spread across a university campus.
Security		
Sec.1	Does the testbed feature isolation between experiments?	Whether two experiments can co-exist without the actions of one tainting the other, for example, a testbed that utilizes virtualization, slicing, VLANs, and other segregation techniques may enable multiple experiments in parallel.
Sec.2	Does the testbed feature isolation between the testbed and the Internet?	Is the testbed isolated from the Internet in such a way that attacks or viruses from an experiment are unlikely or unable to escape the testing environment? This may be accomplished by air-gapping the testbed or using strict firewall rules at the edge of the testbed.
Reproducibility		
Rep.1	Does the testbed enable the execution of reproducible experiments?	Whether the testbed has some means of achieving reproducibility, such as scripted environment configurations or infrastructure as code.
Rep.2	Are applications reproducible?	Whether application configurations are detailed enough to enable others to reproduce their behavior, such as providing a container image, configuration files, software versions, etc.
Rep.3	Are attacks reproducible?	Attack instances are detailed enough so that to enable others to reproduce their behavior? For example, by documenting attack tool versions and the command-line arguments used.
Rep.4	Which of the following best describes the network topology? (i) not documented, (ii) documented, (iii) automatized or otherwise scripted	Can the testbed network topology can be reconstructed from the article to reproduce it? This goal may be achieved by textual documentation or via a diagram.
Flexibility		

Flex.1	Is it possible to modify the testbed to perform new experiments? In particular, (i) by adding or removing devices and (ii) by modifying the behavior of existing devices?	Whether new devices can be added to the testbed, if is so, how easily can such action be done? For example, testbeds may allow adding devices, but if physical devices are involved, they most likely require extra steps for them to work. Plus, such physical device may be costly, not only in terms of monetary cost, but also the space and time cost it requires to set up. Lastly, such devices may be reconfigurable. Does the testbed have any resources for enacting such configurations?
Flex.2	How is the topology assembled? (i) manual assembly, (ii), configurable (SDN,NFV), (iii) scripted	Whether the testbed network topology is manually assembled by the researcher each time a new experiment topology is needed, or if it is reconfigurable via SDN/NFV or scripting.
	Measurability	
Mea.1	What kinds of artifacts are collected by the testbed? (i) network captures (pcaps), (ii) application logs, (iii) sensor telemetry	Whether the testbed collects any artifacts for future use, and if so, what are the data sources of such artifacts?