Putting Attacks in Context: A Building Automation Testbed for Impact Assessment from the Victim's Perspective

Herson Esquivel-Vargas[1](✉), Marco Caselli[2], Geert Jan Laanstra[1], and Andreas Peter[1]

[1] University of Twente, Enschede, The Netherlands
{h.esquivelvargas,g.j.laanstra,a.peter}@utwente.nl
[2] Siemens AG, Munich, Germany
marco.caselli@siemens.com

Abstract. Cybersecurity research relies on the reproducibility and deep understanding of attacks to devise appropriate solutions. Different kinds of testbeds are typically used to systematically execute attacks and evaluate defenses. Testbeds are widely used to demonstrate Building Automation and Control System (BACS) attacks and defenses, considered too risky to be executed on real infrastructures. However, those testbeds implement arbitrary configurations of building services that do not resemble real-world deployments. In this work, we present the first BACS testbed specially designed to assess the impact of cyberattacks from the victim's perspective. It features general purpose building services such as illumination, ventilation, and temperature control, whose configuration is easily adapted to emulate the requirements of real-world locations. In this way, the context added to our testbed allows us to better understand the impact of BACS attacks through concrete and realistic scenarios. Moreover, by analyzing different configurations of the BACS (i.e., contexts), we found out that identical attacks may have dramatically different impacts. Thus, reinforcing our view on the relevance of adding context to BACS testbeds.

1 Introduction

Cyber Physical Systems (CPSs) refer to a variety of applications where computer systems interact with physical aspects of the world [22]. Those physical aspects include variables such as speed, temperature, and pressure, whose automated control has proved crucial in many industries. The building automation industry is one of them, where physical variables are controlled through building services such as heating, ventilation, and air conditioning. The interconnection and centralized management of building services is achieved through Building Automation and Control Systems (BACSs).

The influence that CPSs exert in the real world has been traditionally regarded as a major security concern. For that reason, the *impact* of CPS

C. Maurice et al. (Eds.): DIMVA 2020, LNCS 12223, pp. 44–64, 2020.
https://doi.org/10.1007/978-3-030-52683-2_3

attacks has been typically measured as the deviation of physical variables from pre-established setpoints [1,14,33]. While such deviations indeed constitute the physical manifestation of an attack, not all of them represent a threat. In fact, several physical changes may naturally occur without noticeable consequences.

Specifically on BACSs, the experience on real-life attacks suggests that the adversaries' goal is typically to leverage the physical capabilities of the system to thwart business processes [8,24]. We deem these attacks as a specialization of physical impact attacks, tailored to drift physical variables beyond a business acceptable threshold (see Fig. 1). Since BACS attacks have a direct effect on organizations' daily operations, we argue that the impact assessment of BACS attacks should be done from the victim's perspective, specifically, the *Business Continuity Impact* (BCI).

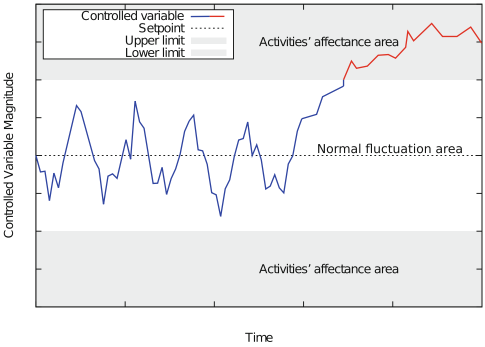


Fig. 1. Controlled variables have limits beyond which the supported activity gets negatively affected. Such limits depend on each specific activity.

From a defensive perspective, BACS attacks must be carefully analyzed by cybersecurity researchers to devise appropriate solutions. The replicability of such attacks is crucial to methodically evaluate defensive approaches. However, attack execution in production infrastructures is risky. To overcome this limitation, testbeds provide a safe experimentation platform that removes the risk of damaging production systems.

Traditional BACS testbeds serve as a demonstration platform for defensive mechanisms [11,13,25] whose capabilities are, in turn, commonly demonstrated in light of two sets of attacks: those that can and cannot be handled by the defensive tool. No context nor special attention to the attacker's goal is needed for such experiments; the focus is placed on the low level details of the attack. Instead, we address the challenge of building the first BACS testbed specialized in the assessment of the BCI of cyberattacks.

The testbed described in this work implements three general building services, namely, illumination, ventilation, and temperature control. Those building services can be reconfigured to fit the requirements of diverse business contexts. Such reconfigurability can hardly be achieved in other kind of CPS testbeds, where one particular scenario is commonly embedded in the hardware itself (e.g., electric grid, water treatment plant, etc.). Leveraging this feature, our testbed allows to reproduce identical attacks on different business contexts and compare their BCI.

Our results show that the context is crucial to properly assess the impact of cyberattacks. The reason being that the BCI is always relative to the victim's use of building services. This insight gets embodied in the testbed by configuring the building services according to the victim's needs. Only then, a context-rich BCI assessment of cyberattacks can be conducted.

Contribution. (i) *A detailed description of the development process of a BACS testbed specialized in the assessment of Business Continuity Impact (BCI) of cyberattacks.* We provide all the engineering materials, custom software, and information sources needed to replicate our testbed.

(ii) *We provide empirical evidence of the context's relevance by exposing remarkably different impacts (BCI) on identical attacks.* Through the implementation of three different emulated environments in our testbed, we provide concrete and realistic examples that show how different organizations under identical attacks suffer the consequences differently.

Organization. Hereafter the paper is organized as follows. The literature review is presented in Sect. 2. We elaborate on the tight relation between business processes and BACSs in Sect. 3. Section 4 describes the process of creating a testbed for BCI assessment from the victim's perspective, followed by our experiments in Sect. 5. Finally, we present the conclusions of our work in Sect. 6.

2 Related Work and Background

Testbeds. The common objective of all security CPS testbeds is to execute attacks and to evaluate defenses. On top of that, different goals are set which yield different testbed implementations. Not necessarily mutually exclusive, typical testbed goals are demonstration, education, and impact assessment [20,32]. Demonstration testbeds are built to convince stakeholders of the applicability of both offensive and defensive research findings [32]. Education testbeds are skill-development platforms where students, researchers, and practitioners can learn hands-on [2,21]. Finally, impact assessment testbeds use a variety of metrics to quantify the consequences of cyber attacks [1,23,27].

BACS testbeds in particular, have overlooked the relevance of *impact* analyses of cyberattacks to mostly focus on the demonstration of security solutions [11,13,25]. To show the strengths and weaknesses of these tools, they appeal to attack instances to exemplify success and failure cases. Although we acknowledge the illustrative value of such testbeds, the lack of high level context information makes it difficult to recognize the attacks' potential impact in real-world scenarios and to realize the actual value of the proposed defenses. Our testbed addresses this limitation by incorporating context as part of its default operation.

Impact Metrics. Several Industrial Control System (ICS) testbeds have been built to study the *physical impact* of cyberattacks. For instance, a water treatment testbed is used in [33], where the impact is defined as the deviation in the pre-established pH level of the water. In [1], a water distribution testbed is presented where the impact of attacks is measured as the decrease in the supplied water with respect to the normal capacity of the system. Yet another example are the smart-grid testbeds presented in [23,27], where the impact of attacks is measured in terms of voltage (in)stability, generation loss, and load shedding increment.

Other kinds of impact have been analyzed as well. Packet delays have been measured as the impact of communication outages [23], and even the performance decrease after introducing cybersecurity controls has been considered [9].

Most impact metrics do not consider the level of disruption from the organization's perspective. Since the goal of BACSs is to support diverse business processes in organizations, a measure of the business impact of cyberattacks is needed. A BACS BCI metric is described in [12]. In summary, it is based on a methodology that merges business and technical aspects of the BACS. From the technical perspective, this methodology leverages on a graph data structure whose nodes and edges represent BACS components and functional dependencies, respectively. BACS components are then labeled with an initial score that represents their support on business processes. Finally, a centrality measure (called BACRank) is computed on the graph to score BACS components based on their BCI.

Rather than designing a new impact metric, our goal in this project is to build a testbed that works as a BACS reference implementation whose properties (e.g., design, emulated business processes, etc.) are used to instantiate existing BCI metrics.

Attacks. There are two main types of attacks in CPSs: those inherited from the IT domain and the attacks that exploit the physical capabilities of the system. Examples of attacks inherited from the IT domain are packet flooding, packet spoofing, and password attacks [15]. On the other hand, attacks that leverage the physical capabilities of the CPS include triggering alarms, opening/closing valves, blinking lights, etc. [1,29]. IT attacks often serve as a first step towards cyber-physical attacks. The scope of our work is focused on cyber-physical attacks.

Specifically on the BACS domain, different attacks have been described in literature. Many of them targeting BACnet (ISO 16484-5) [3], one of the most popular BACS protocols currently in use [13]. A condensed list of attacks is shown in Table 1.

An attack classification framework is needed to methodically analyze attacks and defenses. The threat intelligence community has developed a number of taxonomies to structure knowledge about cyberattacks. One of such taxonomies is Mitre's ATT&CK framework [31]. This framework describes Tactics, Techniques, and Procedures used by adversaries. *Tactics* are the high-level goals of the attacker, whereas *techniques* refer to the expected actions required to achieve those goals. Finally, the *procedures* describe specific details about how to

Table 1. BACS attacks described in literature.

Protocol	Attack	Reference
MS/TP	DoS via frame desynchronization	[16]
ICMP	DoS via smurf attack	[15]
IP	DoS via packet flooding	[15]
PPP	Backdoor via modem connection	[15]
BACnet	DoS via malformed packet injection	[19]
BACnet	DoS via Initialize-Routing-Table command	[15]
BACnet	DoS via Reinitialize-Device command	[11]
BACnet	Snooping via I-Am-Router-To-Network command	[15]
BACnet	DoS via depletion of CoV subscriptions	[26]
BACnet	Firmware corruption via File object writing	[10]
BACnet	Data manipulation via WriteProperty attack	[19]

instantiate the techniques. Although the ATT&CK framework was initially created for standard IT enterprise environments, an analogous framework for ICSs has been published recently.[1] Since the ICS version of the framework fits more accurately the BACS attacks discussed in this paper, we use it to categorize the type of attacks we aim to study and the concrete instances of attacks executed.

3 Business Processes and BACSs

According to the Information Systems Audit and Control Association (ISACA), a *business process* is a set of inter-related activities that deliver a specific product or service to a customer [17]. Building services play an important role in organizations, supporting the execution of their business processes [12]. However, the configuration of the BACS is different depending on the supported business processes. Each business process location has a set of desired or, in many cases, *required* environmental conditions it must comply with in order to fit its purpose. Ventilation, temperature, illumination, among other conditions, are specified for different locations in diverse documents such as standards, regulations, and best practices guides. Thus, setpoints, thresholds, and control algorithms change depending on the particular setting. Examples of regulated environmental conditions are shown in Table 2, taken from [4–7,28,30].

Ventilation requirements are commonly expressed as liters per second (L/s) or cubic feet per minute (CFM). Those requirements are intended to keep the CO_2 level below the specified values. Details on how to convert such measures to CO_2 parts per million (ppm) can be found in Appendix A.

[1] https://collaborate.mitre.org/attackics/.

Table 2. Required environmental conditions for diverse business process locations.

Business process	Business process location	Illumination (lux)	Ventilation (CO_2 ppm)	Temperature (°C)
Surgeries	Operating room	[500–600]	≤770	[20–24]
Teaching	Lecture hall	[300–500]	≤1400	[20–27]
Server hosting	Data center	[50–100]	–	[18–27]
Blood tests	Laboratory	[750–1200]	≤1400	[20–27]
Physical conditioning	Fitness gym	[200–300]	≤880	[20–22]

4 Testbed for Business Continuity Impact Assessment

Current security testbeds in the BACS domain focus on the demonstration of the protection capabilities of defensive tools. Such demonstrations typically compare the set of attacks that the tool can handle with the set of attacks it cannot. While these attack-based demonstrations draw all attention to the technicalities of the attack, no context information is given to illustrate the attack's potential impact in real-world scenarios and the best use cases for the proposed tools. No BACS testbeds up to now, have used contextual information to analyze the impact of cyberattacks.

To fill this gap, the focus of our testbed is the assessment of attacks impact from the business perspective. We refer to such impact as Business Continuity Impact (BCI). Our aim is to analyze attacks that exploit the physical capabilities of the BACS and whose ultimate goal is to hinder business processes in the targeted organization. In this section, we describe the development process of our testbed, covering its requirements, design, and implementation.

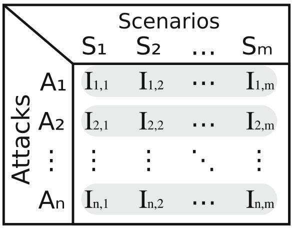
4.1 Requirements

Scenarios. The goal of our testbed is to provide a platform to assess the impact of attacks launched against different business scenarios. We define *scenario* as the combination of a business process location (in previous sections regarded as *the context*) and its supporting building services. Our observation, as can be derived from Table 2, is that a reduced subset of core building services can abstractly represent different business process locations. Based on this observation, the requirement for our testbed is to implement automated illumination, ventilation, and temperature control, so it can reproduce the environmental conditions of diverse locations such as those listed in Table 2. Finally, a software tool is needed to reconfigure and adapt these building services to the requirements of different locations.

Attacks. We focus on attacks that take advantage of the physical capabilities of the BACS. According to Mitre's tactics, techniques, and procedures hierarchy, those attacks correspond to the *impair process control* tactic,[2] in which "[t]he adversary is trying to manipulate, disable, or damage physical control processes.". From a high level perspective, the requirement for our testbed is to provide the technical means to reproduce *impair process control* attacks.

Impact Assessment. The required impact assessment metric must consider the business process where the BACS is deployed. In particular, we are concerned with attacks that can affect the normal execution of business processes. Such metric is commonly known as BCI and allows to assess the impact of the attacks launched against our testbed from the business perspective. Since our testbed should be easily reconfigured to emulate different scenarios, the impact metric can be used to compare identical attacks on many of them. The goal of such experiments is to figure out to what extent the context influences the BCI.

To summarize our requirements, Fig. 2 shows the relation between attacks, scenarios, and the impact assessment metric, where $I_{i,j}$ is the BCI of attack i under scenario j.



		Scenarios			
		S_1	S_2	...	S_m
Attacks	A_1	$I_{1,1}$	$I_{1,2}$	$\cdots$	$I_{1,m}$
	A_2	$I_{2,1}$	$I_{2,2}$	$\cdots$	$I_{2,m}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	A_n	$I_{n,1}$	$I_{n,2}$	$\cdots$	$I_{n,m}$

Fig. 2. Summary of our experimental setup requirements: (1) Reproducibility of *impair process control* attacks; (2) Reproducibility of diverse scenarios modeled through building services; and (3) A BCI metric to compare the impact of attacks on multiple scenarios.

4.2 Design

The minimal experimental setup needed to launch attacks and compute the corresponding BCI, must implement one scenario comprised of at least one business process location and one building service. The building service embodies the technical attack surface that will be targeted by the adversary. The business process tunes the impact metric so that its magnitude reflects the consequences of the attack.

Our testbed integrates illumination, ventilation, heating, and cooling as building services. Figure 3 depicts those services as implementations of an abstract *BuildingService*. Whereas each *Scenario* uses one or more *BuildingService*(s),

[2] https://collaborate.mitre.org/attackics/index.php/Impair_Process_Control.

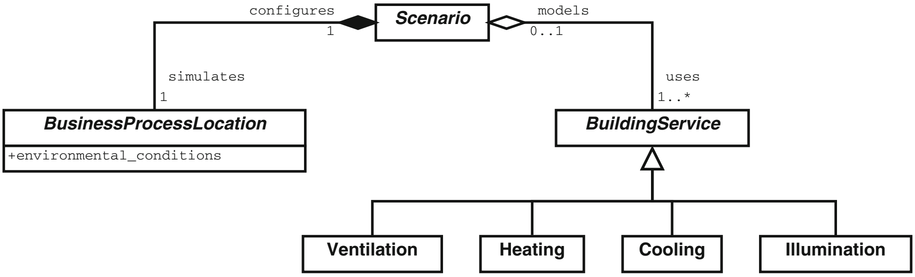


Fig. 3. High level design of our experimental setup.

a *BuildingService* might not necessarily model a *Scenario*. This is how BACS testbeds have been built in the past. It is only by configuring the environmental conditions of a *BusinessProcessLocation* that the overall *Scenario* required to compute the BCI is added to the testbed. From a design perspective, we do not limit the business process locations that can be emulated in our testbed.

The control algorithm differs per building service. Whereas some are activated upon specific time conditions, others require feedback from the environment. The former is known as *open control loop* (see Algorithm 1) and the latter as *closed control loop* (see Algorithm 2). In our testbed, the illumination service is handled by an open control loop. The ventilation and temperature control use closed control loops.

Algorithm 1. Simplified open control loop.

```
while $True$ do
    if $time\_for\_action = True$ then
        $take\_action()$
    else
        $stop\_action()$
    end if
end while
```

Algorithm 2. Simplified closed control loop.

```
while $True$ do
    if $controlled\_var > upper\_limit$ then
        $decrease\_controlled\_var()$
    else if $controlled\_var < lower\_limit$ then
        $increase\_controlled\_var()$
    end if
end while
```

The design of BCI metrics is a complex task beyond the scope of this work. Instead, we use self-evident scenarios whose attacks' BCI can be deduced by domain experts from the business processes' technical requirements. We back up the expert-based assessment with the BCI metric proposed in [12]. We do not design new BCI metrics nor enhance existing ones.

In [12], the components to be assessed are represented as nodes in a graph data structure. The edges of the graph represent dependencies between components. The components' impact scoring is executed in three steps. First, each node is annotated with an initial score that, among other information, considers the relevance of the component from the business perspective. During the second step, the edges are annotated with an estimation of the dependency strength. Finally, after all nodes and edges have been annotated, a graph centrality algorithm (called BACRank) is executed on the graph to assign the final impact score.

The granularity of the components to be assessed depends on the needs of the organization. To simplify our discussion, we use *building services* as high level components to be assessed. This decision reduces the graph size to only three nodes: illumination, ventilation, and temperature control, which includes the heating and cooling services.

4.3 Implementation

Hardware. BACSs comprise diverse components in a 3-layered hierarchical arrangement. At the bottom, there are sensors and actuators, commonly referred to as *field devices*. In the middle, embedded computers in charge of taking inputs from the sensors and sending output signals to the actuators make up the *control layer*. On top, there is a *management layer* which provides unified control and monitoring to BACS administrators.

In our testbed, we use the BACnet communication protocol at the control and management layers [3]. Although at these layers we use software and hardware commonly used in real BACS deployments, at the field level we use smaller actuators than those used in real buildings. This is due to our down-scaled version of building rooms.

We built two physical modules that represent real building rooms. The first module is a *mechanical room* that contains heating and cooling hardware that emulates a building's boiler and chiller, respectively. The second module is a generic *building room* that requires heating and cooling services from the first module. Moreover, it has a thermostat, illumination, and ventilation hardware. The thermostat contains temperature, humidity, occupancy and CO_2 sensors (inputs) and relays to interact with the actuators (outputs). Both modules are physically connected to allow the heat/cold transfer. Figure 4 shows a picture of both physical modules.[3]

[3] The 3D CAD designs, schematics of custom electronics, and bill of materials are published in https://www.utwente.nl/en/eemcs/scs/downloads/2020_BACS_testbed/.

Since the illumination service must adapt to different lighting requirements, it is controlled by an *analog output* that regulates the light intensity. The analog output provides a maximum of 20 mA at [0–12] VDC, which is too low to feed the high power LEDs installed in the building room. A customized electronic circuit was designed to dim the lights according to the driving analog output. The other actuators are controlled using *binary outputs* connected to relays. Thus, avoiding the need for additional circuitry.

The cost of the project can be divided in three parts. The structural components, which includes the aluminum base, profiles, plexiglass, among others, have an approximate cost of $700 USD. The BACnet specific hardware and software has an approximate cost of $3.500 USD (see Table 3). Finally, other components including power supplies, actuators, relays, etc. have an approximate cost of $500 USD. After considering outsourced services (e.g., plexiglass laser cutting), the overall cost of the physical components of the testbed is about $5.000 USD. We consider this as reasonable costs for a small testbed and it should allow other research groups to replicate our testbed.

Table 3. BACnet components used in our testbed.

Vendor	Product	BACnet profile	Approximate cost
KMC	BAC-5050	Router	$1.000 USD
KMC	FlexStat BAC-131136CEW	B-ASC	$1.000 USD
MBS	BACeye version 2.1.0.15	B-OWS	$500 USD
Janitza	UMG 604-PRO	B-SA	$1.000 USD

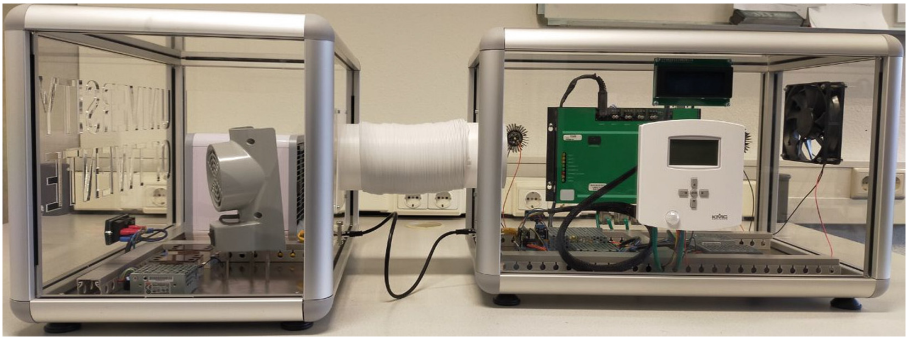


Fig. 4. Testbed modules. The building room (on the right) is physically connected to the mechanical room (on the left) to allow air flow.

Communication. As stated above, the chosen BACS communication protocol is BACnet [3]. The underlying protocols include UDP, IP, ICMP, Ethernet, and

MS/TP. A PPP connection to the Public Switched Telephone Network (PSTN) is also added since it has been documented as an important attack vector for BACS networks [15]. The variety of protocols available provides a considerable attack surface. A network diagram of our testbed is shown in Fig. 5.

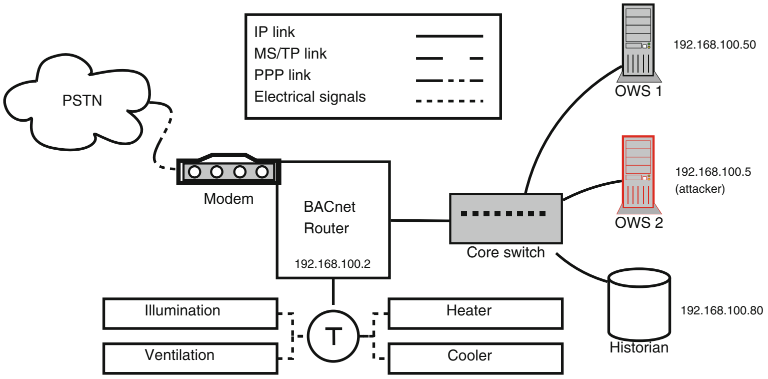


Fig. 5. Network topology (including electrical signals to actuators).

The IP network implements a star topology. The core switch has been configured with a mirroring port to collect all the network traffic exchanged during the experiments.

Software. The most important software applications used in our testbed are BACeye 2.1.0.15 and bacnet-stack 0.8.6.[4] Using bacnet-stack we implement a Linux-based Operator Work Station (OWS 1 in Fig. 5). It runs a custom application developed to quickly reconfigure the testbed to meet the environmental requirements of predefined business process locations.

BACeye runs on a Windows-based OWS, which we assume under control of the attacker (OWS 2 in Fig. 5). Network captures taken during the experiments might show legitimate and malicious traffic from this computer.

The firmware version of the FlexStat controller is R2.1.0.18. The BAC-5050 router runs firmware build R1.8.0.1.

5 Empirical Analysis of BACS Attacks

We execute attacks against the illumination (I), ventilation (V), and temperature control (T) services implemented in our testbed. As specified in our testbed's requirements (Sect. 4.1), the attacks considered is this work correspond to Mitre's *impair process control* tactic. One step down in the ATT&CK hierarchy, there are 11 techniques to implement such tactic. Since we want to replicate the same attack conditions on different scenarios, we chose the *Unauthorized Command*

[4] https://sourceforge.net/projects/bacnet/.

Message technique for all the attacks. According to Mitre's website,[5] following this technique "[a]dversaries may send unauthorized command messages to instruct control systems devices to perform actions outside their expected functionality for process control.". Further down in the hierarchy, we also fix the attack *procedure*. We chose one of the attacks listed in Table 1, specifically, data manipulation via the WriteProperty attack [19]. This attack consists of a syntactically valid BACnet message that changes a property in a BACnet object.

To achieve our goal of comparing the BCI of identical attacks on different scenarios, we pick three business process locations from Table 2, namely the operating room, lecture hall, and data center. Those locations are chosen due to their diverse building service requirements. During the experiments, the testbed is configured to fit the environmental conditions defined for each business process location.

The BCI of cyberattacks can be computed in advance by understanding the requirements of business processes on building services. Building services that are essential for business processes will have larger BCI than other services. The impact levels assigned to building services are technically-backed choices made by domain experts. In what follows, we present a short description of such technically-backed choices for each location assessed. A summary is presented in Table 4.

Operating Room. The World Health Organization deems illumination as "one of the major nonstructural elements in a hospital" [35]. While most people would agree that all environmental conditions in operating rooms are important, the severity and immediacy of an attack on the illumination service are key factors to consider it as the highest priority service, above the ventilation and temperature control, both considered of medium impact.

Lecture Hall. The concern for air quality is common in densely occupied indoor spaces [4]. A high concentration of CO_2 (e.g., $\geq$1400 ppm) might lead to illness symptoms such as headaches and dizziness. Moreover, the ventilation is considered a high priority service in lecture halls since it has been shown that improving the air quality increases the students performance [34]. Although illumination and temperature are also relevant, they have been scored as medium impact services.

Data Center. Data centers are extremely sensitive to temperature [6]. Whereas low temperatures increase the chances of electrostatic discharges, high temperatures might damage the servers' hardware, or trigger safety mechanisms to automatically power them off. For those reasons, temperature control is by far considered the most important building service for the continuity of operations in a data center. Data centers do not have ventilation requirements (see Table 2) mainly because servers do not produce CO_2 *in-situ*. Finally, illumination is primarily used to enable video surveillance. For those reasons, the ventilation and illumination are deemed as low impact services.

[5] https://collaborate.mitre.org/attackics/index.php/Technique/T855.

Table 4. BCI levels of building service attacks on different contexts. Highlighted in bold font the *high* impact services per location.

Attack	Operating room	Lecture hall	Data center
Illumination	**High**	Medium	Low
Ventilation	Medium	**High**	Low
Temperature	Medium	Medium	**High**

5.1 Attacks

We configured our testbed according to the chosen scenarios to launch three attacks in each of them: turning the illumination off, stopping the ventilation service, and stopping the temperature control service. All attacks are executed against the thermostat FlexStat BAC-131136CEW (BACnet Application-Specific Controller). The specifics of each attack are detailed in Table 5. These attacks do not respond to vulnerabilities particular to the device but to the BACnet protocol itself.[6]

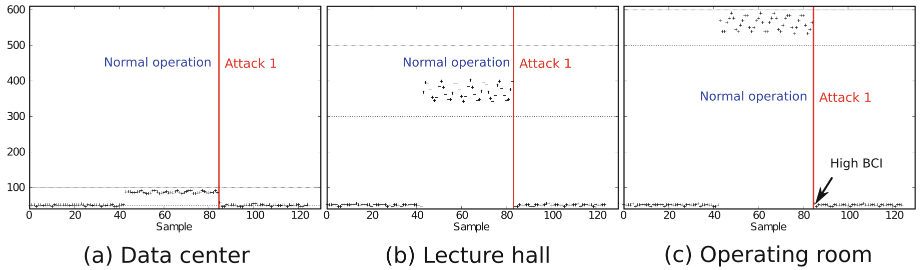
Table 5. Attack procedures against the building controller. Object types and instance numbers provided to ease the analysis of the corresponding pcap files.

No	Attack	BACnet service	Object type	Object instance	Written value
1	Illumination	WriteProperty	Analog output	5	0
2	Ventilation	WriteProperty	Binary output	1	0
3	Temperature	WriteProperty	Binary output	2,1	0

Illumination. The ambient light in the room where the testbed is located is measured in the range of [46, 52] lux. All the illumination experiments start with sensor readings in this range. After approximately 40 samples of ambient light, the illumination service is turned on at the intensity needed to meet the requirements of each specific scenario. Approximately 40 samples later the first attack is executed, which causes the sensor to report the ambient light intensity again, confirming thus the attack. Figure 6 shows the illumination samples collected during our experiments for each scenario.

Ventilation. Unlike lecture halls and operating rooms, data centers do not have CO_2 requirements (see Table 2). Since there are no consequences from the business perspective, we did not execute a ventilation attack on the data center scenario.

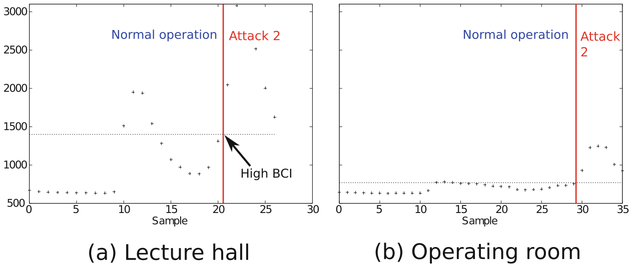
[6] Network captures of each attack are published in pcap format at https://www.utwente.nl/en/eemcs/scs/downloads/2020_BACS_testbed/.



(a) Data center (b) Lecture hall (c) Operating room

Fig. 6. Illumination attack on different scenarios. The y-axis represents lux units for all scenarios. Data points collected during the experiment are shown using the "+" character. Dashed lines show the minimum and maximum allowed values.

During the experiments, the ambient CO_2 level is in the range of [632, 674] ppm. For both ventilation attacks we take approximately 10 sensor readings before leaking CO_2 inside the testbed's building room. We use 16 g cylinders of CO_2 commonly found in bike shops to inflate tires. As expected, the CO_2 values increase but are quickly brought back to normal by the ventilation service. Once the CO_2 values are below the threshold, the fan is automatically deactivated which causes the CO_2 level to rise above the maximum limit again. The maximum limit violation triggers the ventilation service a second time. At this point, the attack is executed (i.e., the ventilation is turned off) which causes the CO_2 level to keep increasing. Finally, the CO_2 source depletes its content which drops the sensor readings again. Figure 7 shows the CO_2 level in our testbed during both experiments simulating the lecture hall and operating room locations.

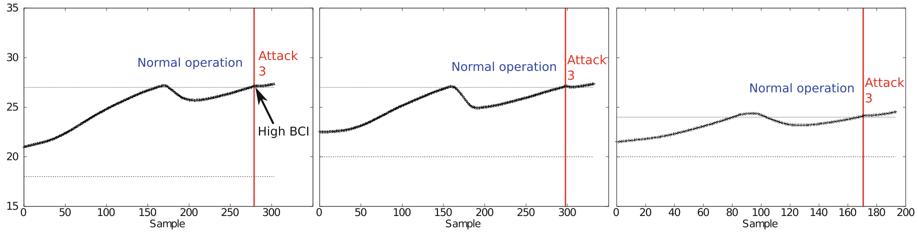


(a) Lecture hall (b) Operating room

Fig. 7. Ventilation attack on the lecture hall and operating room scenarios. The data center scenario is excluded since it does not have specific CO_2 requirements. The y-axis represents CO_2 ppm units for both scenarios. Data points collected during the experiment are shown using the "+" character. The dashed line shows the maximum allowed values.

Temperature Control. Each experiment starts by recording the ambient temperature of the testbed's *building room*. Afterwards, a source of heat is placed inside the room. For these experiments, three anti-spill aluminum bottles filled with boiling water are used as heat source.

As in the previous experiments, we first let the system react as it was designed to work. Later on, the third attack is executed which turns off both the cooler, physically located in the testbed's *mechanical room*, and the fan, located in the testbed's *building room*. Both devices are controlled from the thermostat by *binary output* object instances 2 and 1, respectively. Although the attack comprises two components, the goal is to increase the temperature regardless of the CO_2 level measured by the ventilation service. Figure 8 shows the temperature plots of our three experiments.



(a) Data center (b) Lecture hall (c) Operating room

Fig. 8. Temperature attack on different scenarios. The y-axis represents degrees Celsius for all scenarios. Data points collected during the experiment are shown using the "+" character. Dashed lines show the minimum and maximum allowed values.

5.2 BACRank Scoring

To back up the intuitive BCI of attacks discussed in the previous section, here we follow the methodology described in [12] to measure it. As discussed in Sect. 4.2, the BACS must be modeled as a graph data structure, where the nodes represent the building services implemented: illumination, ventilation, and temperature control.

The edges of the graph model the way in which the BACS is programmed and built. In our testbed, the illumination and ventilation services do not have external dependencies. The temperature control service, on the other hand, depends on the ventilation service to make the heat/cold transfer from the mechanical room to the building room. From the implementation point of view, the strength of such dependency is 100%. A graphical representation of the graph is shown in Fig. 9.

According to [12], each asset m of the BACS is represented as a vertex in the graph, where the granularity of such assets can range from specific data points to entire building services. The initial score given to each vertex (denoted as δ) at time t is defined as:

$$\delta(m,t) = \begin{cases} \max_{1\leq i\leq n}(\beta(p_i)\cdot\gamma(s_j,p_i)) & \text{if time}(p_i,t)=\text{time}(m,t)=1,\\ 0 & \text{otherwise,}\end{cases}$$

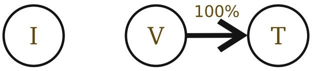


Fig. 9. Graph used to compute the BCI of the illumination (I), ventilation (V), and temperature control (T).

where function β returns the Business Impact Analysis (BIA) [18] score of business process p_i, out of n business processes in the organization. Moreover, function γ encodes how relevant building service s_j (of which m is part) is to business process p_i. Finally, *time* is a binary function that is overloaded to take as input a business process or a BACS asset. $time(p_i, t) = 1$ means that business process p_i is running at time t, and $time(m, t) = 1$ means that asset m is needed at time t.

Three components of the δ function are simplified when using the BACRank methodology in our testbed:

Assets. Unlike [12], that considers software modules as the assets to evaluate, we use building services as coarse grained assets. This decision simplifies our discussion while preserving all the properties of the original methodology.

Business Processes. Since we consider only one business process per organization (i.e., hospital→operating room, hosting company→data center, and university→lecture hall), subscripts are not needed for business processes. Furthermore, since BIA scores of different organizations are not comparable, we assume each business process to have identical values for β.

Time. We assume that all building services and business processes are needed/active at the time of the assessment.

These changes lead to a simplified version of the original function:

$$\delta(s_j) = \gamma(s_j, p).$$

Thus, it is clear that the initial scoring of each building service is a function of the business process p. Table 6 specifies the initial scores of the building services implemented in our testbed. Moreover, it contains the final BCI score of each service and, consequently, of the attacks targeting them. Details on how to compute the BACRank score are described in [12]. A brief summary is also provided in Appendix B.

Table 6. Initial and BCI scores of the implemented building services. Highlighted in bold font the BCI values considered *high* in Table 4.

Location	$\delta(I)$	$\delta(V)$	$\delta(T)$	BCI(I)	BCI(V)	BCI(T)
Operating room	1.0	0.5	0.5	**1.0**	1.0	0.5
Lecture hall	0.5	1.0	0.5	0.3	**1.0**	0.3
Data center	0.1	0.1	1.0	0.1	1.0	**1.0**

The BACRank-based BCI score is normalized in the range [0–1] per organization. By comparing the BCI scores from Table 6 with the BCI scores from Table 4, it is possible to observe a match in the most important building services. That is not the case for some services previously considered of *medium* or *low* impact. This is because in addition to business aspects, BACRank considers technical aspects omitted in the first assessment. The BACRank-based BCI tends to increase the ventilation service score because other building service (i.e., temperature control) as a strong dependency on it.

6 Conclusion

We have presented the first BACS security testbed focused on the assessment of Business Continuity Impact (BCI) of cyberattacks. The unique feature of our testbed is its capability to reconfigure the implemented building services to fit the requirements of different business process locations. Its BACS design and emulated business processes are used to instantiate existing BCI metrics, which shed light on the impact of identical attacks on different scenarios. We have made available all the materials needed for other research groups to replicate our testbed and experiments.

The hardware of our testbed is essentially similar to the hardware found in existing testbeds. In the same way that we abstractly represent business processes as specific configurations of the BACS, existing testbeds could incorporate context by configuring their building services to fit the needs of business processes of choice. Regardless of the original purpose of their testbed (e.g., education, demonstration, etc.), the addition of context would enable them to analyze attacks from the victim's perspective.

Although simple BCI assessments could be done independently of a physical testbed, more sophisticated BCI metrics require additional information such as the BACS design. In these cases, the BACS design of the testbed could be used as an input of the BCI metric. We have presented both kinds of assessments in this work. The development of new BCI metrics was beyond our scope.

Using our testbed, we showed that the addition of context is required to properly assess the BCI of BACS attacks. More than a requirement, such context is a crucial aspect that can swing an attack evaluation from high impact (e.g., illumination in an operating room) to low impact (e.g., the same illumination attack in a data center).

A key aspect of our BCI assessments is that the impact scores are linked to the targeted *physical variables* (and their corresponding building services) but not to the attack procedures. This approach decouples our reasoning about cyberattacks from the low level details of their implementation. The impact materializes only after the variable crosses a predefined threshold, whatever the means.

The selection of security controls should be based on the concept of *risk*, commonly defined as the product of impact and probability of attacks. By identifying the impact of physical variables on business processes, it is possible to make a better assessment of the defensive tools needed to protect the business continuity. This aspect is typically overlooked by current BACS testbeds focused on the demonstration of security solutions.

As future work, we will use the context added to our testbed to experiment with context-aware intrusion detection systems. Moreover, we will address the execution of automated attacks as an optimization problem that tries to maximize the impact in each particular scenario.

Acknowledgments. This work is partially funded by the Costa Rica Institute of Technology.

A Ventilation Rate

The ventilation rate Q, commonly measured in liters per second (L/s), is computed using Eq. 1, where:

- G is the CO_2 generation rate per person (assumed 0.005 L/s).
- C_i is the acceptable indoor CO_2 concentration, measured in parts per million (ppm) and is different per business process location.
- C_a is the ambient CO_2 concentration (assumed 350 ppm).

$$Q = \frac{G}{(C_i - C_a)} \tag{1}$$

The CO_2 values in Table 2 refer to the C_i parameter, which can be obtained rearranging Eq. 1, given Q.

B BACRank Centrality Measure

The BACRank centrality measure is defined as:

$$\text{BACRank}(m, t; i) = \begin{cases} \delta(m, t), \text{ at iteration } i = 0, \\ \delta(m, t) + \sum_{n \in N^+(m)} \text{BACRank}(n, t; i-1) \cdot \omega(e_{m,n}), \text{ for } i > 0. \end{cases}$$

The BACRank score measures the BCI of node m at time t through several iterations i. At iteration 0, each node in the graph gets as score the initial value assigned by the δ function (see Sect. 5.2). For all the following iterations, node m gets as score the initial value δ plus a contribution from the nodes that depend on m. We call this set $N^+(m)$. The contribution consists on a fraction of current BACRank score of all nodes $n \in N^+(m)$. The fraction of the transferred score depends on the weight of the edge (denoted ω) between nodes m and n. After a number of iterations, depending on the complexity of the graph, the BACRank score converges for all nodes in the graph. This is then considered the final BCI score of each *element* in the BACS.

References

1. Ahmed, C.M., Palleti, V.R., Mathur, A.P.: WADI: a water distribution testbed for research in the design of secure cyber physical systems. In: Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER@CPSWeek 2017, Pittsburgh, Pennsylvania, USA, 21 April 2017, pp. 25–28 (2017). https://doi.org/10.1145/3055366.3055375
2. Almgren, M., et al.: RICS-el: building a national testbed for research and training on SCADA security (short paper). In: Critical Information Infrastructures Security - 13th International Conference, CRITIS 2018, Kaunas, Lithuania, 24–26 September 2018, Revised Selected Papers, pp. 219–225 (2018). https://doi.org/10.1007/978-3-030-05849-4_17
3. ANSI/ASHRAE STANDARD 135–2016: A Data Communication Protocol for Building Automation and Control Networks (2016)
4. ANSI/ASHRAE STANDARD 62.1-2016: Ventilation for Acceptable Indoor Air Quality (2016)
5. ANSI/ASHRAE/ASHE STANDARD 170–2017: Ventilation of Health Care Facilities (2017)
6. ANSI/TIA: ANSI/TIA-492-A Telecommunications Infrastructure Standard for Data Centers (2012)
7. ANSI/TIA: ANSI/TIA-569-C Telecommunications Pathways and Spaces (2012)
8. Bilefsky, D.: Hackers use new tactic at Austrian hotel: locking the doors. https://www.nytimes.com/2017/01/30/world/europe/hotel-austria-bitcoin-ransom.html. Accessed 22 Oct 2019
9. Candell, R., Stouffer, K., Anand, D.: A cybersecurity testbed for industrial control systems. In: Proceedings of the 2014 Process Control and Safety Symposium (2014)
10. Chipkin: The 18 Attack Types Using the Vulnerabilities of BACnet. https://store.chipkin.com/articles/the-18-attack-types-using-the-vulnerabilities-of-bacnet. Accessed 10 Sept 2019
11. Esquivel-Vargas, H., Caselli, M., Peter, A.: Automatic deployment of specification-based intrusion detection in the BACnet protocol. In: Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, Dallas, TX, USA, 3 November 2017, pp. 25–36 (2017). https://doi.org/10.1145/3140241.3140244
12. Esquivel-Vargas, H., Caselli, M., Tews, E., Bucur, D., Peter, A.: BACRank: ranking building automation and control system components by business continuity impact. In: Computer Safety, Reliability, and Security - 38th International Conference, SAFECOMP 2019, Turku, Finland, 11–13 September 2019, Proceedings, pp. 183–199 (2019). https://doi.org/10.1007/978-3-030-26601-1_13
13. Fauri, D., Kapsalakis, M., dos Santos, D.R., Costante, E., den Hartog, J., Etalle, S.: Role inference + anomaly detection = situational awareness in BACnet networks. In: Detection of Intrusions and Malware, and Vulnerability Assessment - 16th International Conference, DIMVA 2019, Gothenburg, Sweden, 19–20 June 2019, Proceedings, pp. 461–481 (2019). https://doi.org/10.1007/978-3-030-22038-9_22
14. Hadziosmanovic, D., Sommer, R., Zambon, E., Hartel, P.H.: Through the eye of the PLC: semantic security monitoring for industrial processes. In: Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, 8–12 December 2014, pp. 126–135 (2014). https://doi.org/10.1145/2664243.2664277

15. Holmberg, D., Evans, D.: BACnet wide area network security threat assessment. US Department of Commerce, National Institute of Standards and Technology (2003)
16. HVACR control: Attack BACnet MSTP by frame desynchronization. http://www.hvacrcontrol.com/attack-bacnet-mstp-by-frame-desynchronization/. Accessed 13 Sept 2019
17. ISACA: Cybersecurity fundamentals glossary (2018). https://www.isaca.org/Knowledge-Center/Documents/Glossary/glossary.pdf
18. ISO 27031:2011: Information technology -Security techniques- Guidelines for information and communication technology readiness for business continuity (2011)
19. Kaur, J., Tonejc, J., Wendzel, S., Meier, M.: Securing BACnet's pitfalls. In: ICT Systems Security and Privacy Protection - 30th IFIP TC 11 International Conference, SEC 2015, Hamburg, Germany, 26–28 May 2015, Proceedings, pp. 616–629 (2015). https://doi.org/10.1007/978-3-319-18467-8_41
20. Kavallieratos, G., Katsikas, S.K., Gkioulos, V.: Towards a cyber-physical range. In: Proceedings of the 5th on Cyber-Physical System Security Workshop, pp. 25–34. ACM (2019)
21. Kim, J., Kim, K., Jang, M.: Cyber-physical battlefield platform for large-scale cybersecurity exercises. In: 11th International Conference on Cyber Conflict, CyCon 2019, Tallinn, Estonia, 28–31 May 2019, pp. 1–19 (2019). https://doi.org/10.23919/CYCON.2019.8756901
22. Lee, E.A.: Cyber physical systems: design challenges. In: 11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008), 5–7 May 2008, Orlando, Florida, USA, pp. 363–369. IEEE Computer Society (2008). https://doi.org/10.1109/ISORC.2008.25
23. Liu, R., Vellaithurai, C., Biswas, S.S., Gamage, T.T., Srivastava, A.K.: Analyzing the cyber-physical impact of cyber events on the power grid. IEEE Trans. Smart Grid **6**(5), 2444–2453 (2015). https://doi.org/10.1109/TSG.2015.2432013
24. Metropolitan.fi: DDoS attack halts heating in Finland amidst winter. https://metropolitan.fi/entry/ddos-attack-halts-heating-in-finland-amidst-winter. Accessed 22 Oct 2019
25. Pan, Z., Hariri, S., Pacheco, J.: Context aware intrusion detection for building automation systems. Comput. Secur. **85**, 181–201 (2019). https://doi.org/10.1016/j.cose.2019.04.011
26. Peacock, M., Johnstone, M.N., Valli, C.: Security issues with BACnet value handling. In: Proceedings of the 3rd International Conference on Information Systems Security and Privacy, ICISSP 2017, Porto, Portugal, 19–21 February 2017, pp. 546–552 (2017). https://doi.org/10.5220/0006263405460552
27. Poudel, S., Ni, Z., Malla, N.: Real-time cyber physical system testbed for power system security and control. Int. J. Electr. Power Energy Syst. **90**, 124–133 (2017)
28. Rea, M.: The IESNA Lighting Handbook: Reference & Application. Illuminating Engineering Society of North America, New York (2000)
29. Ronen, E., Shamir, A.: Extended functionality attacks on IoT devices: the case of smart lights. In: IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, 21–24 March 2016, pp. 3–12 (2016). https://doi.org/10.1109/EuroSP.2016.13
30. Sanders, M.: ACSM's Health/Fitness Facilities Standards and Guidelines. Human Kinetics, Champaign (2019)
31. Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B.: Mitre ATT&CK^{TM}: design and philosophy. Technical report (2018)

32. Tippenhauer, N.O.: Design and realization of testbeds for security research in the industrial internet of things. In: Alcaraz, C. (ed.) Security and Privacy Trends in the Industrial Internet of Things. ASTSA, pp. 287–310. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12330-7_14
33. Urbina, D.I., et al.: Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016, pp. 1092–1105 (2016). https://doi.org/10.1145/2976749.2978388
34. Wargocki, P.: Improving indoor air quality improves the performance of office work and school work (2008)
35. World Health Organization and others: Hospital safety index: Guide for evaluators (2015)