Chapter 18

User Experience Models for Threat Monitoring and Security Management in Health Care

By Fabrizio Bertone, Francesco Lubrano, Federico Stirano, Zenjie Li, Barry Norton, Michele Petruzza and Marco Gavelli

DOI: 10.1561/9781680838237.ch18

Published in *Cyber-Physical Threat Intelligence for Critical Infrastructures Security* by John Soldatos, Isabel Praça and Aleksandar Jovanović (eds.). 2021. ISBN 978-1-68083-822-0. E-ISBN 978-1-68083-823-7.

Suggested citation: Fabrizio Bertone, Francesco Lubrano, Federico Stirano, Zenjie Li, Barry Norton, Michele Petruzza and Marco Gavelli. 2021. "User Experience Models for Threat Monitoring and Security Management in Health Care" in *Cyber-Physical Threat Intelligence for Critical Infrastructures Security*. Edited by John Soldatos, Isabel Praça and Aleksandar Jovanović. pp. 391–414. Now Publishers. DOI: 10.1561/9781680838237.ch18.

now
the essence of knowledge

18.1 Introduction

The continuous monitoring of security and safety in hospitals is a complex task that involves many persons covering different roles and interacting with different systems. While the technical backend aspects of threat monitoring and security management tools are essential for the execution of their tasks, an optimal and consistent User Experience is also important for a correct interpretation of the information and a quick reaction to identified issues. Security analytics tools can be complex and require trained personnel with specific skills not always available in such environments. For these reasons, the direct involvement of end users during the design phase of the graphical interfaces is an important step to ensure a better quality of experience and acceptance.

This chapter describes a set of tools used and developed in the context of the SAFECARE project, focusing especially on those with which end users, in particular security operators and emergency managers, interact directly.

The following sections describe two systems related to physical security, used by guards and security operators. The first is deployed in traditional fixed monitoring locations, while the other involves the use of mobile devices and can also be used by medical or other staff.

A different kind of platform described later is a management system used by crisis managers to analyze and get updates on asset availability, in particular when cyber or physical incidents occur.

From another perspective, analyzing physical security in a hospital environment is a challenging task, in particular because it involves data collection activities, which often have to consider hazardous and manifold scenarios, such as forceful intrusion and fire. Due to the critical function of hospitals, it is usually a good practice to use simulated environments to test attack scenarios or run training sessions. Privacy is another concern in this context, narrowing the possibilities to leverage advanced features such as face recognition. Furthermore, the recent COVID-19 pandemic has created extra challenges due to the very limited accessibility of hospitals.

Those difficulties can be significantly mitigated by applying the *virtual hospital* concept, which is introduced in the last part of this chapter. For all these reasons, in SAFECARE, besides the real environments, some virtualization techniques were introduced to simulate both the physical environment and the ICT assets (e.g., networks, firewalls, servers, routers, etc.), realizing the virtual hospital, a realistic representation of a common health-care infrastructure.

Figure 18.1 represents the SAFECARE tools presented in this chapter and the connection among them. The SAFECARE modules that provide advanced analytics features, such as the propagation of potential impact, the cybersecurity tools,

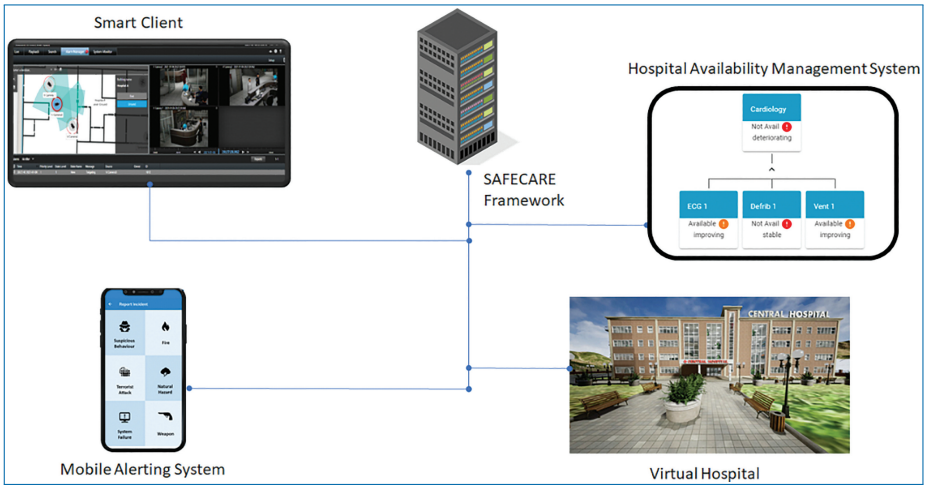

Figure 18.1. Visual representation of the SAFECARE tools presented in this chapter.

and the alerting system are represented by the server image in the center of the figure. These modules and the actual integrated architecture used in the project have been described in detail in (Bertone *et al.*, 2020).

18.2 Video Management System

In a hospital environment, one of the most common monitoring tools is the Video Management System (VMS). Its main role is to allow the live monitoring of security cameras and the management of the video recordings. Additional functions can greatly enhance the usability and the efficacy of this tool, improving the overall capability of identifying potential physical security incidents.

Milestone Systems' VMS, XProtect®, is used in this study. XProtect® is a powerful VMS solution, and the XProtect® Smart Client, as a part of this product, is a graphic application for daily surveillance operations. Using Smart Client, the users can view live and recorded videos, view devices displayed on maps, receive and acknowledge alarms. Thanks to the Milestone Integration Platform Software Development Kit[1] (MIP SDK), one can easily add support for hardware devices, either physical or virtual, and add new custom software features.

18.2.1 Smart Client—Integrated Display of Map, Floorplan, Cameras, and Alarms

Smart Client users can view and access cameras and other devices at multiple locations worldwide, based on geography and building layout, using a feature called Smart Map. It can also show the monitored buildings, including the floorplan, cameras, and monitoring sensors, in an integrated display.

A building can be added to a map that is based on a map server, such as OpenStreetMap or Google Maps. A building is represented by a quadrilateral with freely adjustable vertices to match the physical extent of the building in the real world. The user can add one or more floors to this building and add multiple buildings in different locations as shown in Figure 18.2(a). The user can then zoom out to see all the buildings, and quickly navigate to each location to view video feeds from each camera (Figure 18.2(b)). It is also possible to add links to buildings in other locations, making an easy switch between different locations for multisite hospitals.

Inside a building on the map, cameras and monitoring sensors such as fire sensors and temperature sensors can be added with the exact locations, as represented in

1. https://doc.developer.milestonesys.com/html/index.html

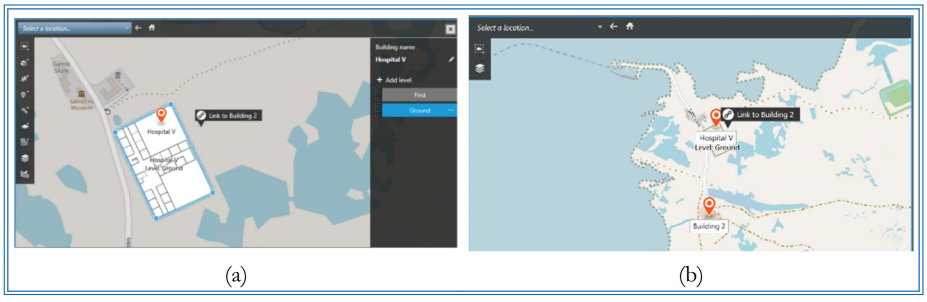

Figure 18.2. Building representation on OpenStreetMap in Smart Client. (a) Adding a building; (b) Showing all buildings by zooming out.

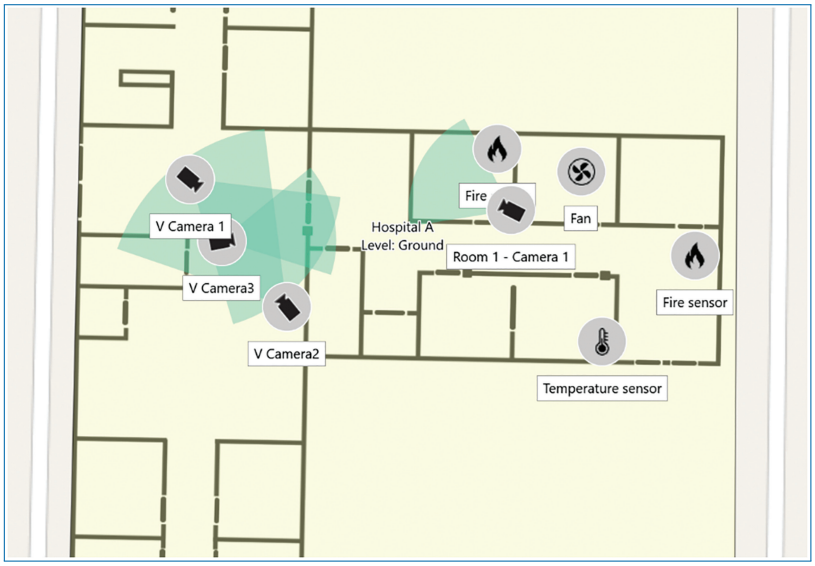

Figure 18.3. Cameras and monitoring sensors are added to the building on the map.

Figure 18.3. For cameras, the orientation and field of view can also be precisely represented.

The Alarm Manager page on Smart Client allows the user to visualize the map, display related videos and a list of alarms when a camera, a sensor, or some rule or advanced analytics functionality based on a combination of these, triggers an alarm. In this case, connected cameras and sensors will be highlighted on the map, and allow the user to view the relevant part of their recorded video feed or other output.

To demonstrate these features, two different scenarios have been performed in the virtual hospital.

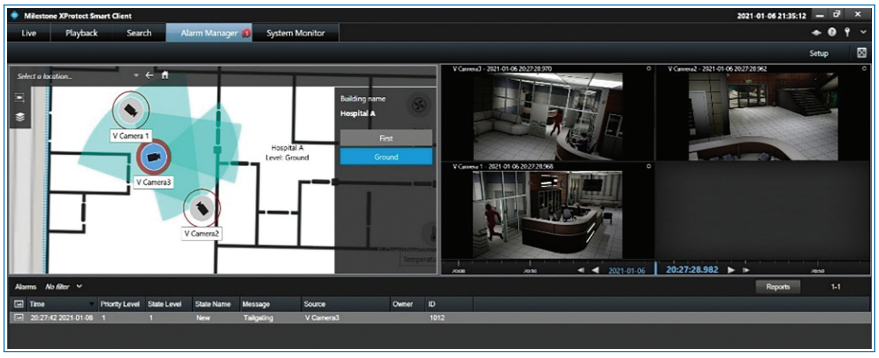

Figure 18.4. A tailgating scene is detected, and an alarm is shown in Smart Client.

Tailgating Example

Tailgating is a behavior in which an unauthorized person follows an authorized one when passing an access-controlled gate. To automatically detect this particular behavior, a specific video analytics plug-in has to be installed on the VMS software.

Indeed, video analytics plug-ins are often deployed with VMS software to implement advanced features, such as automatic detection and alerting. In Milestone XProtect®, the Video Processing System (VPS)[2] is the framework for integrating third-party video analytics.

To demonstrate the video analytics plug-in installed in XProtect®, we have simulated a tailgating scene in the virtual hospital, where an unauthorized malicious person follows a staff member that is crossing an access-controlled gate. The video analytics plugin detects that two persons have passed the door based on the live video of V Camera 3, while the door has only opened once according to the door access control. Thus, a tailgating behavior is detected, based on the output from the video analytics and access control system, and an alarm is triggered. V Camera 3 and two other connected cameras in the same room, V Camera 1 and V Camera 2, are highlighted on the map in Smart Client as seen in Figure 18.4. Simultaneously, the recorded videos of the tailgating scene are shown in Smart Client. The alarm is also shown on a list at the bottom of the window so the operator can take further action.

Fire Example

We have simulated a fire scene in the virtual hospital room, which contains both a fire sensor and a camera. When a fire sensor triggers a fire alarm, the sensor itself and the connected camera in the same room will be highlighted on the map in the

2. https://doc.developer.milestonesys.com/html/gettingstarted/intro_vps_toolkit.html

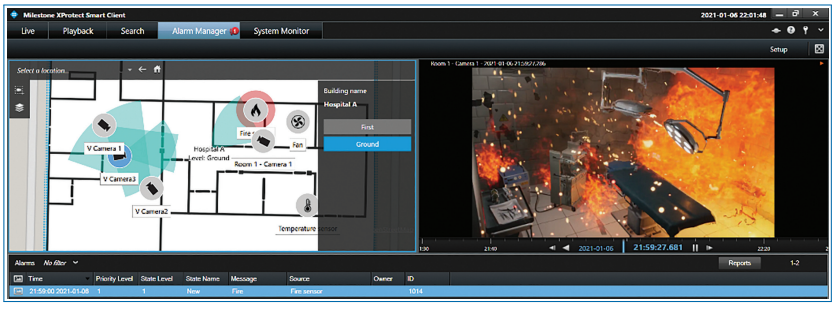

Figure 18.5. A fire is detected by the fire sensor, and an alarm is shown in the Smart Client Alarm Manager.

Smart Client Alarm Manager. Simultaneously, the recorded videos of the fire scene are shown in Smart Client. Both events can be seen in Figure 18.5, respectively on the left and right sides. The alarm is also shown on a list so the operator can react (at the bottom of the same image).

18.3 Mobile Alerting System

The purpose of the Mobile Alerting System is to enable both the security and medical personnel of the hospital to collaborate with the help of pre-existing security infrastructure by taking advantage of the pervasive presence of portable terminals like smartphones and tablets. This integration aims to improve the response of the hospital to cyber-physical threats by:

- improving the reaction time of operators and workers;
- enriching the communication with the operators by providing updated information on threats and *response plans* with contextual information (e.g., location inside the hospital, timestamp, affected assets);
- enabling the staff to report specific security threats (e.g., system failures, natural hazards, suspicious behaviors, etc.).

This section describes the user interface of the mobile app and the implemented functionalities.

18.3.1 Report Incident

The *Report Incident* functionality enables all the users of the application to report possible threats that are happening in the hospital.

Figure 18.6 shows how the interface for incident reporting looks like. The first screen (a) shows the top-level threat classes in which the complete list is subdivided.

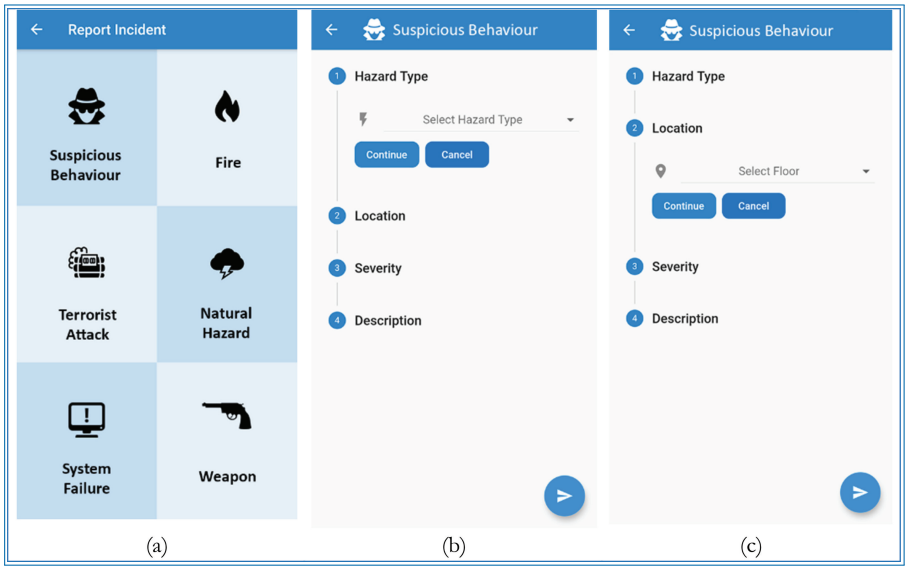

Figure 18.6. Incident reporting.

Once a class of incidents is selected, a new screen will be shown to allow filling the fields required to complete the report (Figure 18.6(b) and (c)). Depending on the chosen incident class, the type of information that needs to be filled in can be different.

18.3.2 Alert Evaluation

The alert evaluation functionality is available only to the security operators stationed around the hospital and enables them to receive information (e.g., videos from cameras, location inside the hospital, assets involved) on possible threats that need to be verified in person.

To access the Alert Evaluation screen, the user can either tap on the notification displayed upon receiving a new alert (Figure 18.7(a) and (b)) or by navigating to the Alerts page and selecting one of the alerts of the list (Figure 18.7(c)).

Inside the Alert Details Screen (Figure 18.8) is visualized the information (severity of the alert, component which generated it, etc.) regarding the alert and the security events that generated it. By clicking on a single event inside the alert, detailed information (textual or video) is displayed.

An Alert can be in one of two possible states as can be seen in previous Figure 18.7(c):

- Not Assigned: no security personnel have yet given the availability to verify the event detailed inside the alert. In this case, the security guard can become

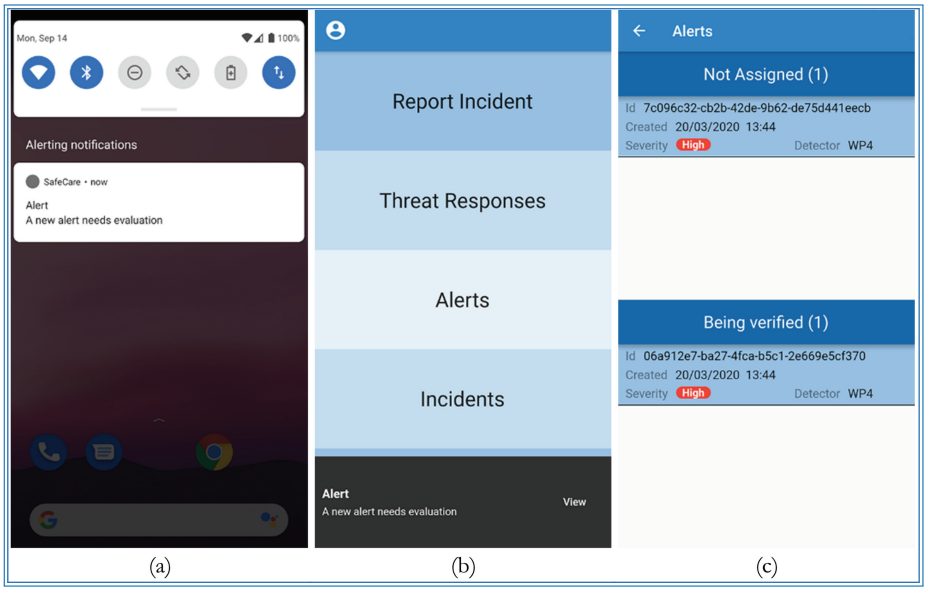

Figure 18.7. Alert evaluation.

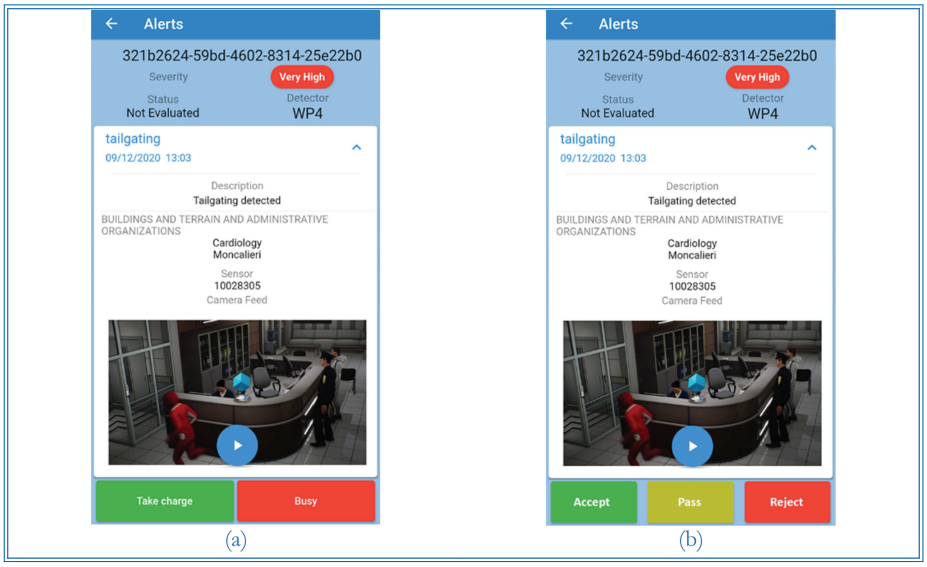

Figure 18.8. Alert details.

in charge of verifying the alert by selecting the "Take charge" button on the bottom left side of the alert screen (Figure 18.8(a)).

- Being verified: a security guard is already tasked with the verification but is yet to answer.

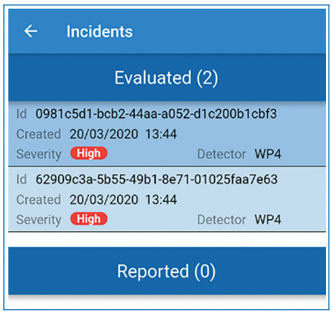

Figure 18.9. Incident history.

The security guard that took charge of verifying the alert can give feedback on it with the three new buttons that appear inside the alert (Figure 18.8(b)):

- Accept: the content of the alert is confirmed, and the alert is to be promoted into an incident.
- Reject: the content of the alert is a false alarm, or it does not constitute an emergency. In this case, the alert is rejected, and it does not become an incident.
- Pass: the security guard has encountered some problems and can't verify the alert. In this case, the alert is returned to the "not assigned" state, and a new notification is sent to the mobile users to restart the verification process.

18.3.3 Incidents History

The "Incidents" screen contains alerts promoted to incidents by the mobile app users ("Evaluated" list) and incidents reported directly by the mobile app users ("Reported" list), both lists can be seen in Figure 18.9. This screen is accessible only by the security operators. By clicking on one of the incidents the user can see more detailed information about the incident.

18.3.4 Impact Evaluation

The "Impacts" screen contains the list of impact messages received (Figure 18.10(a)). By selecting an impact, the "Impact Details" screen will open, showing all the information about the impacted assets with their related information (e.g., type, location, asset Id, etc.) in table form (Figure 18.10(b)). By selecting the "Incident Id" inside the impact details, the "Incident Details" screen of the corresponding incident will open.

Figure 18.10. Impact evaluation.

Figure 18.11. Threat response.

18.3.5 Threat Response

The "Threat Response" screen contains the *threat responses*: messages containing actions to be taken in case of emergency, depending on the role of the receiver (e.g., in case of a fire alarm hospital workers may be tasked with evacuating patients from an area while the security personnel may be tasked with the fire extinction). The message may also contain additional information like videos.

A *threat response* can be visualized by clicking on the view button inside the notification as can be seen on the bottom of Figure 18.11(a) or by selecting it inside the "Threat Responses" screen (Figure 18.11(b)).

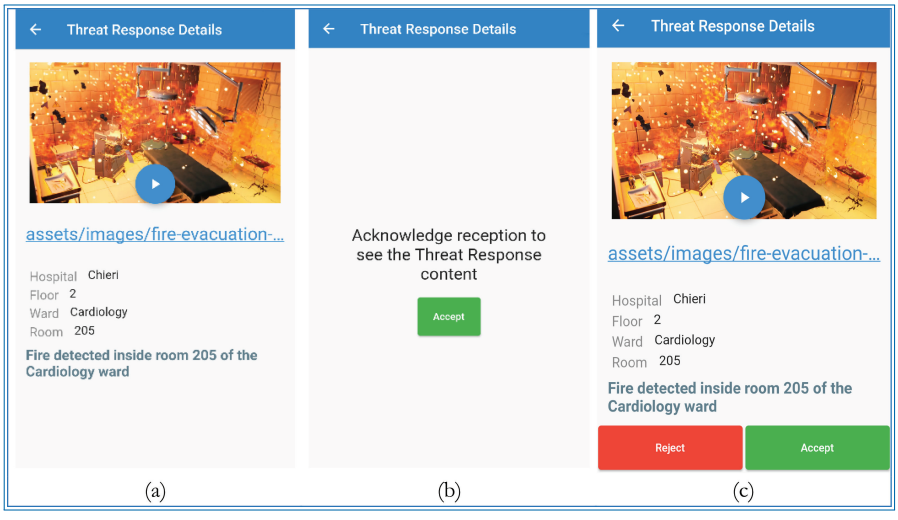

Figure 18.12. Threat response details.

The *threat response* message can be differentiated into 3 types of priority depending on the feedback required from the user when received:

- Notification: no confirmation is requested upon reception (Figure 18.12(a)).
- Acknowledgment: the user must confirm the reception of the notification (Figure 18.12(b)).
- Confirmation: the user must give feedback by accepting or rejecting the content of the threat response (Figure 18.12(c)).

Inside the page displaying the *threat response* are contained the details and the additional information attached (textual or video).

18.4 Hospital Availability Management System

During a crisis situation in a hospital due to physical and/or cyberattacks, as occurred during the Wannacry malware outbreak in 2017 (Ghafur *et al.*, 2019), the security managers need to understand which assets have been affected, possibly including potential cascading effects on other assets. During the recent COVID-19 pandemic, an increase of both cyber (Muthuppalaniappan and Stevenson, 2020) (e.g., malware attacks) and physical (Devi, 2020; World Health Organization, 2020) (e.g., theft) incidents have been observed (SAFECARE, 2020), worsening an already critical situation of scarce medical resources.

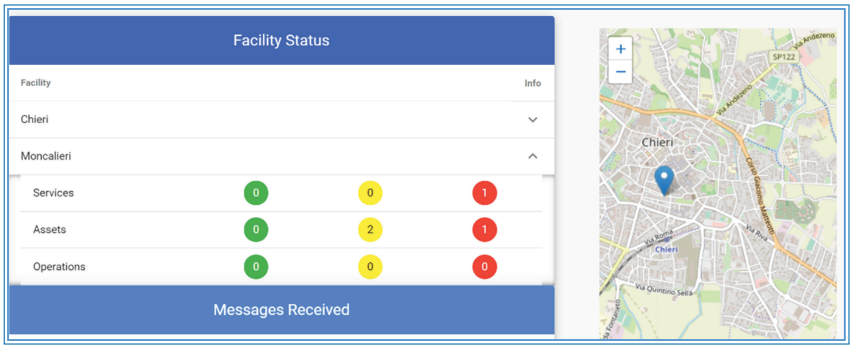

Figure 18.13. HAMS home page.

In this critical context, operators need to understand in a timely and reliable way which services and how many resources (e.g., staff and beds) are still available.

Therefore, having a fast communication of detected incidents and subsequent processing of availability are key points to provide relevant information as soon as possible, giving emergency managers and medical operators the possibility to take more accurate decisions.

In the SAFECARE project, a system called *Hospital Availability Management System* (HAMS) (Lubrano *et al.*, 2021; Stirano *et al.*, 2021) has the role of managing and monitoring the availability of assets and provide updated status and availability information, in particular after cyber and/or physical incidents. The integration of cyber and physical security aspects into a unique system is a pillar of the SAFECARE project. The HAMS leverages this integration to provide updated information inside a unique user interface, designed with the support of the end users. This interface represents an innovative way to manage the hospital status, spread the alerts about incidents, and analyze the potential impacts on assets and health services.

The HAMS home page, depicted in Figure 18.13, provides general information about the overall status of the hospital and shows the location of the facilities and buildings. The two tables in this view summarize the current situation in the hospital, through a simple color-coded list. The first table provides the status of the facilities or services:

- Green: the facility operates normally;
- Yellow: the facility has been involved in an incident but is still running;
- Red: the facility has been involved in a severe incident that heavily affects the availability of services, assets, or operations.

The second table contains the number of messages received by the HAMS for the three categories considered (incidents, impacts, and response reports).

Incident messages include both physical and cyber incident messages and report a set of security alerts, validated by human operators. Impact messages contain lists of assets potentially impacted by the incident and are provided by an internal module of SAFECARE called Impact propagation & Decision Support Model (Atigui *et al.*, 2020). Finally, response reports contain information about the relevant users that received security alerts and how they replied to these alerts. This aspect plays a key role during emergency management: the possibility to verify the list of recipients that have been alerted, who acknowledge and who not, is a powerful mean to better manage the emergency and can have significant and positive impacts towards a more efficient management of communications, awareness, and effectiveness of the actions done by security officers and other relevant users. As introduced in Section 18.3.5, SAFECARE includes in its framework an automatic alerting system that triggers predefined reaction plans (threat responses) according to the severity and the type of incident, and the nature of impacted assets.

Finally, the HAMS integrates a real-time notification system that alerts users in case of reception of new *incident*, *impact*, or *response report* messages.

18.4.1 Table View

The Table views (Figures 18.14 and 18.15) provide a snapshot of the current availability status of the different services, assets, and operations through a tabular view.

The Department Table, shown in Figure 18.14, provides information on the availability status of the departments (services) in the selected facility. The availability status is represented by the fields:

- Availability: a Boolean variable (i.e., *available* or *not available*);
- Status: a parameter that can assume values *green*, *yellow,* or *red;*
- Stability: a parameter that can assume the values of *stable*, *deteriorating,* or *improving*.

Figure 18.14. HAMS department availability table view.

Figure 18.15. HAMS operations availability table view.

Each department row contains also the information related to the number of beds and staff members available. Department rows can be expanded to show the associated assets (e.g., the medical devices), providing the status information for each one of them.

Both department and asset status information can be manually modified by the authorized users, after clicking on the pencil icon in the "Actions" column.

The Operation Table (Figure 18.15) is very similar to the Department Table. It shows the representation of the status availability of each operation and gives as well the possibility to manually change the status by clicking on the pencil icon.

18.4.2 Tree View

The HAMS offers a second kind of view to visualize the hierarchical structure of a facility. Leveraging on a tree representation, this view provides a clear picture of the availability status of the different assets with an intuitive representation of the hierarchy among them (Figure 18.16).

There are two main branches in the tree. The first one lists the medical services, corresponding to the departments in the Dashboard View, with their status. Recursively, each service lists the depending medical devices. On the other hand, the second branch lists the operations. In this case, there are no other associated assets, as HAMS focuses mainly on medical assets, while each operation is seen as a complete system and not split into single assets.

Each element in the tree is represented by a card showing the availability status, with a colored icon and text to inform the user about the availability status.

18.4.3 Incidents List

The HAMS interface provides a dedicated view to show details on the received *incident* messages and the corresponding *impact* and *response* messages.

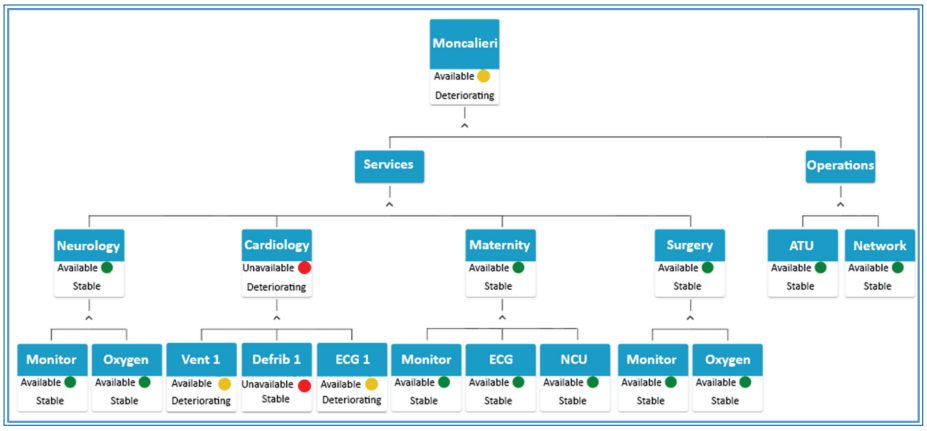

Figure 18.16. HAMS Tree view.

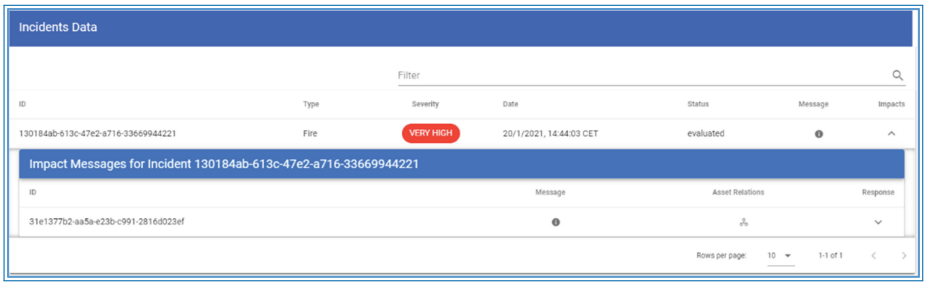

Figure 18.17. HAMS incidents list.

Incidents are represented through a table, sorted by incident date in descending order. Each incident row shows the incident category, the severity, the date, the status, and it is possible to view the full message by clicking on the icon in the message column.

By expanding the incident row, the impact messages related to the selected incident are visualized (Figure 18.17). In the impact row, it is possible to visualize the full message and the impact graph built using the relations among the hospital assets.

The impact row can be expanded to show the information coming from the response messages.

18.4.4 Impact Graph Visualization

The HAMS can provide a visualization of the impact graph including the involved assets, based on the output of other analysis modules of SAFECARE. The graph displays the relations among the different assets of a hospital, highlighting the assets

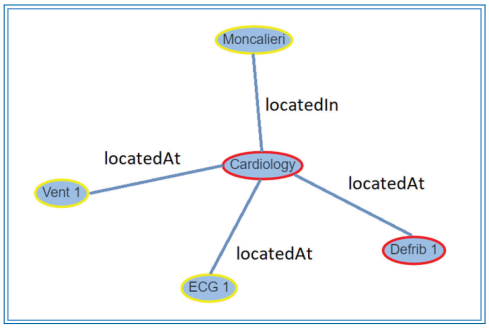

Figure 18.18. Impact graph of assets affected by an incident.

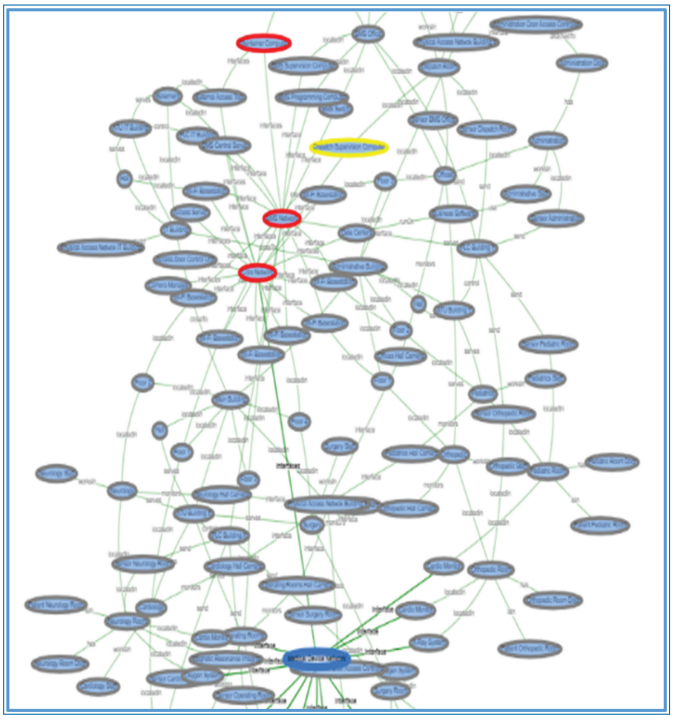

Figure 18.19. Complete graph of assets relations.

that are affected by an impact following an incident. Assets in the graph are colored according to the values contained in the impact message (Figure 18.18).

It is also possible to view the complete graph containing all the assets registered in the hospital together with all their relations and logical links (Figure 18.19).

This view can be useful to understand visually the relations between all the assets, however, given the high number of them registered in a hospital, its representation can become quite dense and overwhelming. Filtering functions have been implemented to reduce the number of elements shown and better navigate the graph.

18.5 Virtual Hospital

A *virtual hospital* is a 3D digital model that can be used to provide an overview (demonstration) of a large-scale security monitoring system, intuitive security training, and enables the reproduction of complex attack scenarios from multiple views. The SAFECARE project makes use of this technology to simulate threat scenarios and to test the tools developed in the field of physical security. The virtual hospital enables an initial validation of the integration of different systems such as cameras, access control systems, fire sensors, etc.

Virtual cameras installed in a virtual hospital are integrated with a VMS by fulfilling the same programmatic interfaces as real cameras, via a driver matching those of the many thousands of camera drivers that ship with XProtect®. It is thereby made possible to simulate different scenarios within the virtual hospital, including simulating actions involving human characters and physical assets, and to have a user interaction via the Smart Client as if there were real camera and sensor feeds within a physical hospital.

This introduction briefly lists the general procedures used for constructing a virtual hospital that will be described more in detail in the following sections.

The first action required to realize a virtual hospital is designing a 3D building starting from a 2D floor plan. This step can be skipped if a 3D model of a virtual hospital is already available, and a real-world mapping is not desired. The 3D building is then imported into a 3D physics simulation engine; in this work we utilize the Unreal Engine, which has its origin in gaming but is widely used in simulation and machine learning environments due to its extensibility and photo-realistic rendering capabilities. Following this, it is possible to add characters and furniture to finish the room set-up and add surveillance cameras and other security items.

Finally, the user can set up event handlers in the game engine to manually trigger actions from outside the virtual world. Having constructed the virtual hospital environment, and installed cameras and other devices, the model can then be connected to the VMS using appropriate drivers. The complete workflow is illustrated in Figure 18.20.

18.5.1 3D Building Design

The process of building a virtual hospital, as mentioned in the introduction, can start from already available templates that represent realistic buildings. But if the use case requires to have a faithful representation of a real hospital in the virtualized world, a software solution is needed to build a 3D model starting from the real 2D floor plan, manually or automatically. This process is fairly straightforward, and there exist several supporting tools that are widely used by architects

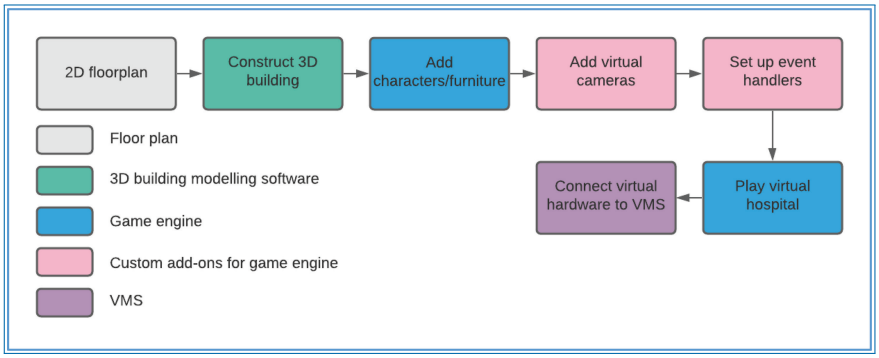

Figure 18.20. Steps for constructing a virtual hospital with camera monitoring.

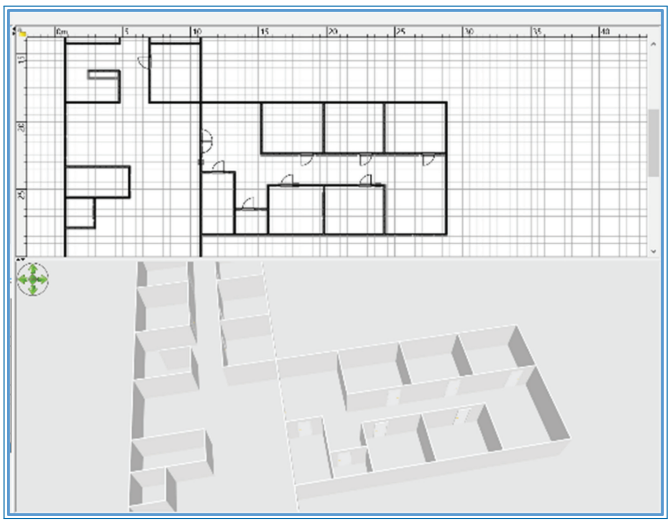

Figure 18.21. Example of constructing a 3D floor plan from 2D using the free and open-source software SweetHome.

and home designers. Figure 18.21 illustrates an example of 3D building construction using a free and open-source application, SweetHome.[3]

After importing the 2D floor plan as an image file, the user can start drawing walls and drag doors in correspondence with the 2D images (Figure 18.21 upper side) while a 3D view is updated synchronously (Figure 18.21 lower side). 3D building construction applications typically also support furniture and other interior room design. Upon completion, the 3D building can be exported to industrial standard formats, such as *.obj* and *.fbx* files.

3. http://www.sweethome3d.com/

18.5.2 3D Game Engine

A 3D game engine is a powerful tool to create realistic-looking scenes with faithful interactions according to a physics simulation. In the application area of health care, the typical requirements for such an engine are:

- It should be compatible with a limited budget.
- It should render high-fidelity scenes and videos with relatively little effort.
- It should not be demanding in artistic design.
- It should not be demanding in programming skills.

The Unreal® engine[4] has been chosen in our work because:

- It is free for internal or free projects based on the Unreal® Engine End User License Agreement for Creators.[5]
- It provides high-fidelity visuals straight out of the box.
- Myriads of free and paid assets are available, and the user does not need to create many virtual objects from scratch.
- Thanks to its visual scripting tool, BluePrint, coding is often unnecessary for ordinary users, while advanced users can still leverage the power of C++ whenever needed.
- Unreal® supports a wide range of operating systems (but the recommended OS is Windows 10 64-bit).

Unreal® Engine version 4.25 has been tested in the current work.

18.5.3 VMS Integrations

Components have been developed to integrate an executable 3D building simulation, in the Unreal Engine, and XProtect® (see Figure 18.22).

The *virtual camera Blueprint* in Unreal® captures the scene as images. The *virtual camera* module (dynamically loadable library, or DLL) acts as a proxy between the virtual camera in the game engine and the VMS. It reads the images and makes them available for the Milestone device driver via HTTP (the Web protocol).

Finally, the virtual camera becomes available in XProtect after installing the device driver.

4. https://www.unrealengine.com
5. https://www.unrealengine.com/en-US/eula/creators

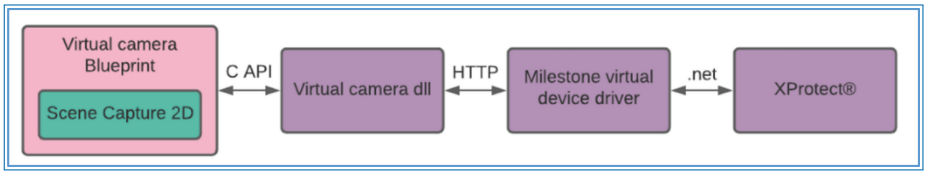

Figure 18.22. Communication of virtual camera and XProtect.

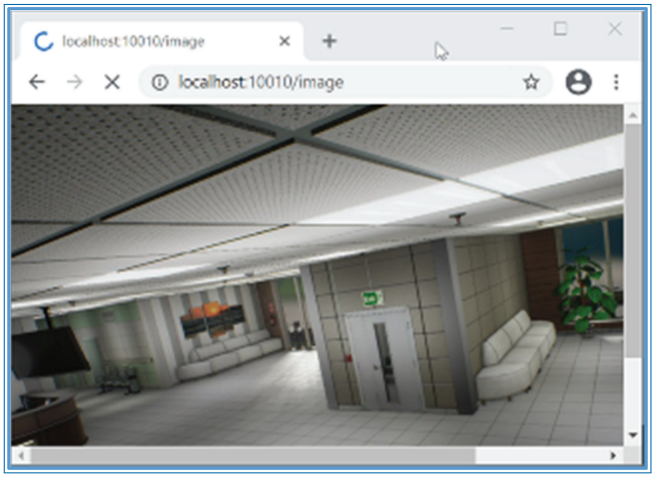

Figure 18.23. Camera view from an internet browser. Google Chrome 87 64-bit is used in this example.

Video capturing of the virtual camera Blueprint implemented in this study is based on the *Scene Capture 2D* component in Unreal®.[6] It is widely used for representing mirrors, mini-maps, teleporters, and security cameras in games. For each frame, the *Scene Capture 2D* captures the scene from its view and stores it as an image in the jpeg format, then pushed to a queue. The *virtual camera DLL* communicates with the virtual world via a standard Windows *DLL* interface.

As an alternative to the VMS, one can directly access the video from a standard modern web browser (see Figure 18.23).

The virtual device driver has been developed with the *Milestone driver framework*[7] that enables devices to be integrated with the VMS. The virtual device driver reads the images at a frame rate set by the user and pushes them to the XProtect Recording Server. The virtual camera appears in XProtect® as an ordinary physical camera.

6. https://docs.unrealengine.com/en-US/API/Runtime/Engine/Components/USceneCaptureComponent2D/index.html

7. https://doc.developer.milestonesys.com/html/gettingstarted/intro_driverframework.html

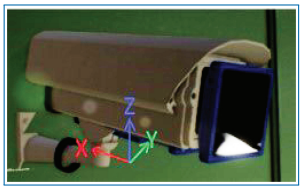

Figure 18.24. Virtual pan-tilt-zoom camera. Tilt: rotate around Y; pan: rotate around Z; zoom: changing field of view.

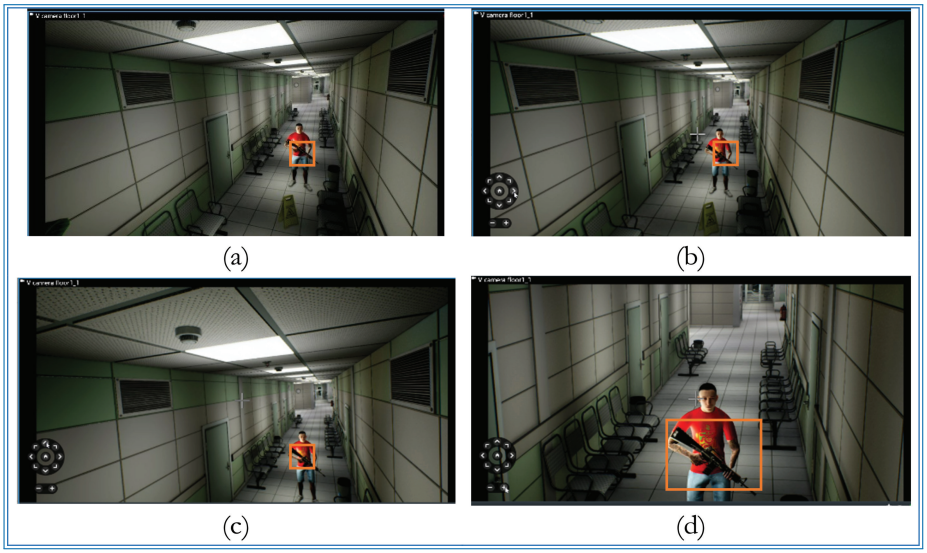

Figure 18.25. Weapon detected by a virtual PTZ camera shown in Smart Client. (a) Camera home view; (b) Camera view panned; (c) Camera view tilted; (d) Camera view zoomed in. Credit for the character: mixamo.com. Credit for the rifle: Mateusz Woliński on sketchfab.com; license: Attribution 4.0 International.[8]

By translating the pan, tilt, and zoom commands from the device driver into rotations and field of view changes, we can simulate real pan-tilt-zoom (PTZ) cameras (see Figure 18.24).

The operator can then use the graphical controls to tilt, pan, and zoom the view in the virtualized hospital scene, as shown in Figure 18.25.

For the weapon detection example shown in Figure 18.25, we have installed a weapon detection plug-in in Milestone XProtect® based on the Video Processing System (VPS) framework. When an armed person appears in the camera view, the weapon will be highlighted with a rectangle and shown in Smart Client, the graphic application of XProtect®.

8. https://creativecommons.org/licenses/by/4.0/

Figure 18.26. Crowding scene shown in Smart Client. Credit for the characters: mixamo.com.

As a further use case example, it is hard to study physical monitoring in a hospital; for example to detect crowding due to privacy or other restrictions, especially during a pandemic. The example below shows a scene where the waiting area is overcrowded, and our VPS-based video analytics plug-in has detected this abnormal phenomenon in XProtect® (Figure 18.26).

Besides virtual cameras, other virtual world controls are often needed to effect a two-way interaction between the VMS and the simulation; e.g., triggering a door opening or a human character action. Those controls can be integrated with similar procedures, with other proxy components responsible for forwarding the commands coming from the operators to the virtual world.

Milestone freely provides the integration components mentioned above.

18.6 Conclusions

The SAFECARE project designed and implemented a complex framework for the integrated security of hospitals. Besides the work "hidden" on the server side, important work was also done researching multiple and optimal user interfaces to maximize the operators' efficacy. The interfaces were specifically designed following end users' directions and feedback, in order to implement the desired functionalities security experts were asking for. The result is a complete suite of software and related user interfaces that allow the management of security in a hospital, providing advanced functionalities to better understand and in case mitigate the impacts of an incident in such critical infrastructure.

The concept of *virtual hospital*, explained in this chapter, is a methodology that can be used to overcome the strict regulations that limit the possibility of freely testing threats and incidents scenarios inside a real hospital. The brief introduction provided in this chapter can be useful to start implementing a virtual hospital (or other kinds of environments) independently.

Acknowledgements

The SAFECARE project has received funding from the European Union's H2020 research and innovation program under Grant Agreement No. 787002.

The authors would like to acknowledge the contributions of Mathias Jes Normann, Gergely Tuskó, and Jakov Rabinovits, working within the SAFECARE project, John Madsen for assisting with XProtect integration, and especially Dennis Schou Jørgensen for work on virtual environments for simulation.

References

Atigui, F., Hamdi, F., Lammari, N., & Cherfi, S. S. (2020). Vulnerability and Incident Propagation in Cyber-physical Systems. In J. Soldatos, J. Philpot, & G. Giunta (Eds.), *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures* (pp. 193–205). Now Publishers. https://doi.org/10.1561/9781680836875.ch11

Bertone, F., Lubrano, F., Gavelli, M., Terzo, O., Biasin, E., Kamenjasevic, E., Demailly, S. D., Lancelin, D., Andernello, S., Tresso, F., Viarengo, L., & Suciu, G. (2020). Integrated Cyber-Physical Security Approach for Healthcare Sector. In J. Soldatos, J. Philpot, & G. Giunta (Eds.), *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures* (pp. 179–192). Now Publishers. https://doi.org/10.1561/9781680836875.ch10

Devi, S. (2020). COVID-19 exacerbates violence against health workers. *Lancet (London, England)*, *396*(10252), 658. https://doi.org/10.1016/S0140-6736(20)31858-4

Ghafur, S., Kristensen, S., Honeyford, K., Martin, G., Darzi, A., & Aylin, P. (2019). A retrospective impact analysis of the WannaCry cyberattack on the NHS. *Npj Digital Medicine*, *2*(1), 1–7. https://doi.org/10.1038/s41746-019-0161-6

Lubrano, F., Stirano, F., Varavallo, G., Bertone, F., & Terzo, O. (2021). Hams: an integrated hospital management system to improve information exchange.

In *Advances in Intelligent Systems and Computing: Vol. 1194 AISC*. Springer International Publishing. https://doi.org/10.1007/978-3-030-50454-0_32

Muthuppalaniappan, M., & Stevenson, K. (2020). Healthcare cyber-attacks and the COVID-19 pandemic: an urgent threat to global health. *International Journal for Quality in Health Care*, *00*(September), 1–4. https://doi.org/10.1093/intqhc/mzaa117

SAFECARE. (2020). *Security Incidents in Healthcare Infrastructure during COVID-19 Crisis*. https://www.safecare-project.eu/?p=588

Stirano, F., Lubrano, F., Vitali, G., Bertone, F., Varavallo, G., & Petrucci, P. (2021). Cross-Domain Security Asset Management for Healthcare. In H. Abie, S. Ranise, L. Verderame, E. Cambiaso, R. Ugarelli, G. Giunta, I. Praça, & F. Battisti (Eds.), *Cyber-Physical Security for Critical Infrastructures Protection* (pp. 139–154). Springer International Publishing. https://doi.org/10.1007/978-3-030-69781-5_10

World Health Organization. (2020). *Attacks on health care in the context of COVID-19*. https://www.who.int/news-room/feature-stories/detail/attacks-on-health-care-in-the-context-of-covid-19