Adaption of a Secure Software Development Methodology for Secure Engineering Design

SUNE VON SOLMS1 (Member, IEEE), AND LYNN A. FUTCHER2

1Department of Electrical Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa
2School of Information Technology, Nelson Mandela University, Port Elizabeth 6031, South Africa

Corresponding author: Sune Von Solms (svensolms@uj.ac.za)

ABSTRACT With the rapid advancement of technologies in the era of Industry 4.0, the inter-connected nature of operations and systems is introducing a rapidly changing landscape of digitized and connected systems. Cybercrime is considered as possibly the greatest threat to connected systems worldwide, and therefore there exists a large drive in engineering to include cybersecurity in the design, development and maintenance of smart cyber-physical systems. Traditionally, the cybersecurity space was considered the responsibility of Information Technology (IT) professionals, where the greater IT infrastructure was required to keep these engineering systems safe. However, through the evolution of engineering and control systems, the IT infrastructure has started to become more integrated with these systems, improving the efficiency of the systems, but also making them more susceptible to cyber-attacks. These changes mean that securing these systems cannot remain the sole responsibility of the IT professionals, as systems must be designed with cybersecurity in mind. Considering that engineers are designing and developing more integrated systems, there exists a knowledge gap in the field of cybersecurity engineering and engineers’ understanding of their cybersecurity responsibilities. This study aimed to determine the level of security that is currently considered in standard electrical engineering projects in a typical academic environment. This baseline serves as a motivation to develop a practical approach to assist engineering students in considering cybersecurity when developing engineering systems and products.

INDEX TERMS Engineering education, engineering design, security, secure software design.

I. INTRODUCTION

The adoption of Internet of Things (IoT) devices and services in businesses and homes leads to the increase of IoT-centric attacks. Even though these attacks are on the rise, IoT continues to enjoy an even greater surge in popularity and adoption [1] where there is thought to be more than 30 million connected devices worldwide [2]. A substantial number of the attacks stem from poorly built integrated systems and software, where security is not considered during the requirements analysis, design, implementation or testing of the product [3], [4]. The vulnerabilities in IoT devices which include outdated software/firmware, default usernames and passwords, and the inability to run software updates or change usernames and passwords [1], are leveraged to gain initial access to networks of corporate targets which then can be further exploited [5].

Traditionally, the Information Technology (IT) professionals were considered the party responsible for cybersecurity, where the greater IT infrastructure was required to secure engineering systems. However, as engineering and control systems became more integrated with the IT infrastructure, securing these systems cannot remain the sole responsibility of IT professionals, as engineering and control systems must be designed with cybersecurity in mind [6]–[8]. These highly integrated systems require that cybersecurity principles be included in engineering software development, networking, device design and deployment, as well as training and education [9]. Unfortunately, many engineering programs still neglect to address the importance of the inclusion of security into the engineering design. Most engineering design methodologies do not consider the risks related to the embedded software and associated data and tend to think of security as an afterthought where the greater IT infrastructure must keep developed engineering systems safe. Engineers designing products or processes need to learn how to consider and include security within the engineering design process.
Various studies have been done on the integration of security into engineering design. In 2016 the International Council of Systems Engineering (INCOSE) started the processes to incorporate security across the entire systems development lifecycle [7], [8] which would support the practice of including security in engineering [10]. The Secure Systems Development Life Cycle (S-SDLC) has been proposed by many scholars as a method to define security requirements which must be included in system, project or application designs. Although lot of work can be seen in the literature on the incorporation of security practices into engineering design, little has been done to provide engineers with a simple, practical approach of how to do so.

At undergraduate level, engineering students are informed about the cyber-risks associated with poorly built integrated systems and software, but are not taught in a practical manner regarding how to address these risks. This leads to engineers who develop products which can ultimately pose a security threat to the user or greater network where it is employed. This study is therefore conducted to:

1) understand and assess the current level of software security understanding among final year Electrical Engineering students at a South African university. 
2) propose a simple and practical approach to integrate security into the engineering systems development life cycle to support engineering students in the development of secure products and processes.

II. METHODOLOGY
In order to develop a practical approach that engineering students can utilize to develop and incorporate secure software in engineering projects, it is important to understand and assess the current level of software security understanding among final year engineering students. To determine this baseline, a content analysis was conducted on final year capstone project documents of a group of final year engineering students.

The student documents used in this content analysis step is from a subset of electrical engineering students from a single institution. It is recognized that this is not an exhaustive study as documents from other engineering disciplines and other universities were not included. However, this study is simply conducted to provide a broad baseline to determine engineering students’ general understanding of security concepts. In order to add more rigor to the study, a brief survey was conducted to determine engineering students’ understanding of and feelings towards software security in engineering. The results from the content analysis and survey conducted provided sufficient evidence to support the need for a practical approach to support engineering students in the development of secure products and processes.

Based on the results from the content analysis, a simple and practical approach is proposed for engineering students on how to incorporate software security practices within engineering design. The method is adapted from the secure software development methodology (SecSDM) proposed by [11].

The results of the various steps are discussed in the subsequent sections.

III. BASELINE STUDY: ENGINEERING STUDENT SECURITY UNDERSTANDING
A baseline study was conducted on final year capstone project documents of final year Electrical Engineering students from a single institution. These documents are written as a requirement for their capstone project in the final year of study. The purpose of the capstone project is to assess the student’s ability to successfully complete a project of limited engineering scope by following the general project lifecycle. This project aims to prepare students for entry into the engineering industry and similar problems that they will encounter and need to solve with independent research (stated in yearbook).

The collection of the final documents of the students for the content analysis was approved by the faculty, where all documents were analyzed anonymously and confidentially, not influencing the students’ grades in any way. In total, 78 documents were approved for analysis. It must be noted that the current curriculum for the capstone project does not include a specific focus on software security and security in general. As indicated by the yearbook entry, the focus of the capstone project is to demonstrate the ability to successfully complete a project through the full normal project lifecycle. Emphasis is placed on the process of development of a project/product, focusing on hardware and software design as well as integration and testing. As such, the use of secure coding practices and the corresponding security-related terms are not expected to be abundant in these documents.

A. CONTENT ANALYSIS
The content analysis involved the analysis of the final year project documents for words relating to various software security practices. The selected security phrases and terms were obtained from the SecSDM approach [11]. Table 1 presents the security-related terms and phrases that were used for the analysis.

B. RESULTS OF THE CONTENT ANALYSIS
The following results were obtained from the content analysis performed on the 78 documents. The results indicate that students mention concepts and terms which are present in the software security field, such as “information”, “data”, “threat” and “vulnerability”. However, when viewed in the context of security or software itself (phrase search), the word counts are not high, indicating that the terms were not used in the context of software security or security at all.

Table 2 shows the term or phrase, the number of documents the term/phrase were found in and the total number of hits. From the results, it can be seen that context must be considered in this search to determine a true baseline. See the following examples for clarification:

1) The term “software” is present in all 78 documents a total of 1279 times and “security” a total of 613 times in 56 documents. However, when searching for
TABLE 1. Security key terms and phrases.

| Terms          | Phrases / words in proximity # |
|----------------|--------------------------------|
| Authenticat*   | Access control                 |
| Authors*       | Data integrity                 |
| Confidential   | Digital signature              |
| Data           | Information assets             |
| Encipher*      | Routing control                |
| Encrypt*       | Security service               |
| Identification | Software/Security              |
| Information    | Software/threat                |
| Integrity      | Traffic padding                |
| Non-repudiation| User authenticat*              |
| Notar*         | User authors*                 |
| Seccur*        | User identification            |
| Software       | User procedure                 |
| Threat         |                                |
| Vulnerabilit*  |                                |

# The search for words in proximity allows for the location of instances where a student wrote “integrity of data” instead of “data integrity.”
* The use of root words and parts of words allows for the location of all forms of a certain word, for example “seccur*” enabled the capture of the forms “security,” “secure,” “securing” etc.
† British English is adopted by South African engineering students.

“software” in proximity to the word “security” it is only present in 5 documents a total of 19 times.
2) The terms “information”, “data”, “threat” and “identification” have a high hit count and are present in many documents, however, where they are used in the context of software security, the count is significantly lower.

These results indicate that students understand issues such as security, integrity and identification, but not always in the context of software. Basic software security issues were mentioned in a limited number of documents, with “user authentication” the most popular topic. Complex security terms and phrases were not found in many documents, with some terms not found in any of the documents. Complex security terminology such as “access control”, “digital signature”, “traffic padding”, “routing control”, “user procedure” and “security service” were not found in any of the documents.

From the content analysis it is clear that software security was not a central theme or design consideration in these engineering projects. This was expected, as security was not a requirement. To triangulate these results, further insight was sought into the final year students understanding of and feeling towards software security in engineering through a short survey given to a small group of students.

C. RESULTS OF STUDENT SURVEY

After the completion of the content analysis, a small group of students who completed their capstone project were requested to complete a survey. Due to the submission dates of the final documents at the end of the academic year, most students were already on holiday when the surveys were conducted and were not available. In total, a small group of 7 students were requested to complete the survey and 5 students’ results could be used. Two students’ results were discarded as they utilized online data sources to assist them in answering the questions, therefore not accurately reflecting their understanding of the security terms and implementation. The survey had 25 questions for participants to answer, which included open and closed responses. The survey was adapted from the SecSDM survey [11] and structured as follows:

1) General information briefly describing the student’s engineering project.

2) Main questions relating to initial risk assessment, design and implementation of software security controls in the engineering project.

3) General opinions where students described how they perceive security within engineering projects and the need for engineers to understand software security.

The results obtained from the survey indicated that the students understood the importance of software and the securing thereof, as well as the concept of identification, authentication and access control services within their project. They also recognized that personal information and access to products need to be protected.

However, all indicated that they did not document these elements as it was not seen as a key element of their project. The student survey also confirmed that there existed many security-related terms which students are not familiar with. Students indicated that they were not sure what terms such as confidentiality, integrity, non-repudiation, traffic padding, routing control and notarization meant. The only instances where students implemented security mechanisms in their project is when they utilized software where they are already built in, such as web portals and databases. Access control was implemented by students where users had to log into a database but was limited to a basic username and password implementation.

Students stated that because they are not trained in security, many engineers do not understand cyber security that well. A subset of the students stated that software engineers and IT professionals who are authorities in the field must identify the security risks and cover the security processes. Other students indicated that cybersecurity training must be included in engineering education so that engineers can design and develop products to security specifications.

The student survey confirmed the results of the content analysis where it can be argued that students generally understand the importance of security but are not trained in it and therefore do not feel responsible for it. The lack of knowledge in this field may also explain why students state that they have implemented some security features but left it undocumented as it was deemed to be an important feature.

It can therefore be motivated that basic education relating to software security services and mechanisms in engineering projects should be provided to engineers. If students are practically guided to clearly identify the services required in their project/product as well as the mechanisms which must be implemented, students might be more comfortable to take responsibility for and implement security features.
TABLE 2. Content analysis results.

| Term         | # Document hits | # Hits |
|--------------|-----------------|--------|
| Data         | 78              | 3246   |
| Information  | 78              | 1321   |
| Software     | 74              | 1279   |
| Secure*      | 56              | 613    |
| Identification| 42              | 127    |
| Authenticat* | 15              | 61     |
| Authors*     | 8               | 39     |
| Threat       | 23              | 38     |

| Phrase       | # Document hits | # Hits |
|--------------|-----------------|--------|
| Data/integrity| 1               | 4      |
| Information/assets| 0           | 0      |
| Software/threat| 1              | 4      |
| Software/Security | 5             | 19     |
| User/identification| 4          | 11     |
| User/authenticat*| 10             | 57     |
| User/authors* | 3               | 4      |

TABLE 3. Phrases SDLC engineering activities [15], [16].

| Phase | Engineering Activities |
|-------|------------------------|
| 0: Concept Exploration Phase | Investigate new opportunities, Explore technology readiness, Evaluate pre-concept match with users’ needs |
| 1: Requirements & Conceptualisation / Initiation Phase | Identify stakeholders’ needs, Evaluate alternate concepts, Recommend possible solutions, Define system requirements & specifications (functional, performance, operational, support etc.) |
| 2: Design & Development / Acquisition Phase | Develop detailed planning, Design system architecture & allocate systems requirements to subsystems and components, Develop detailed system design (logical & physical), Performs system synthesis analysis to assure system integration |
| 3: Production & Implementation / Production Phase | Produce system components & construct system, Code/configure system functional components, Conduct code walkthrough/unit tests, Perform system integration, Inspect and test (systems acceptance test) |
| 4: Utilisation & Support / Operations / Maintenance Phase | Operate system to satisfy users’ needs, Provide sustained systems capability, Perform corrective maintenance |
| 5: Retirement / Disposition / Disposal Phase | Store, archive or dispose of system |

FIGURE 1. Generic systems engineering development lifecycle (adapted from [14]).

IV. SECURITY ACTIVITIES IN ENGINEERING DESIGN

A. SYSTEMS DEVELOPMENT LIFE CYCLE

There exist many engineering design processes, where one of the widely used is the Systems Development Life Cycle (SDLC). There exist many variations of the SDLC, which are tailored toward certain designs and implementations [12, 13a, 14a]. Generally, these steps can also be shown in a more linear manner. See the general V-diagram of the SDLC and the linear translation as shown in Figure 1. Based on the 1220-IEEE Standard for Application and Management of the Systems Engineering Process (2005), the following are basic activities associated with each phase of the linear SDLC as shown in Figure 1 [15], [16].

Note that various names are provided for each phase, due to the fact that various standards use different names for each phase. Table 3 captures a summary of various versions within one table.

B. SECURITY CONSIDERATIONS FOR ENGINEERING DESIGN

There exist much literature on the development and use of a Security Systems Development Life Cycle (S-SDLC), but there currently exists no standard for the S-SDLC.

Table 4 indicates how some experts consider the integration and implementation of security within the SDLC [16] according to activities listed in SP 800-160 Vol. 1 and the Information Assurance Technical Framework (IATF) [17], [18]. It must be noted that the listed activities are not exhaustive and that many others exist. The order and phase in which the activities are executed may also differ for other frameworks and processes. The summary of security considerations captured in Table 4 do, however, provide enough clarity on the types of activities proposed for each phase of the SDLC.

C. SECURE SOFTWARE DEVELOPMENT METHODOLOGY (SecSDM)

Although there exists many frameworks and suggestions on where and how to incorporate security into engineering design (as shown in Table 4), the complexity lies in the practicality of implementation. Many documents discuss what should be done, but clear and practical guidelines are not provided.

Reference [11] developed the SecSDM methodology together with a corresponding software tool as they found...
TABLE 4. Security considerations in the SDLC.

| Phase                              | Security Activities: SP 800-160 | Security Activities: IATF |
|------------------------------------|---------------------------------|---------------------------|
| 0: Concept Exploration Phase       | Survey & understand the policies, standards, and guidelines. | Discover information protection needs |
|                                    | Define security classification & protection level. | Ascertain the system purpose |
| 1: Requirements & Conceptualisation / Initiation Phase | Identify information assets | Identify information asset needs protection. |
|                                    | Conduct preliminary risk assessment. | Define System Security Requirements based on the protection needs. |
| 2: Design & Development / Acquisition Phase | Conduct risk assessment. | Design System Security Architecture |
|                                    | Define security requirements & select security controls (categories & types). | Design system architecture to meet on security requirements. |
|                                    | Perform cost/benefit analysis (CBA). | Develop Detailed Security Design based on security architecture |
|                                    | Security planning (based on risks & CBA) | Design security functions & features for the system |
|                                    | Practice Information Systems Security Engineering (ISSE) process to develop security controls | |
|                                    | Develop security test & evaluation plan for verification & validation of security controls | |
| 3: Production & Implementation / Implementation / Production Phase | Implement security controls in accordance with system security plan (SSP) | Implement System Security |
|                                    | Perform Security Certification & Accreditation of target system. | Implement designed security functions and features into the system. |
| 4: Utilisation & Support / Operations / Maintenance Phase | Configuration management & perform change control | Assess Security Effectiveness & assess effectiveness of ISSE activities |
|                                    | Continuous monitoring | |
|                                    | Perform periodic security assessment | |
| 5: Retirement / Disposition / Disposal Phase | Preserve information, archive and store electronic information | |
|                                    | Sanitize media. | |
|                                    | Dispose hardware. | |

FIGURE 2. SecSDM mapped to the software development life cycle (adapted from [11]).

that many IT undergraduate programs neglect to address the importance of integrating information security into the software development lifecycle. They developed the SecSDM as a practical approach for the integration of security into software design. The software tool was designed to be practical and to provide a clear step-by-step approach to identify the required security controls and how to practically implement the corresponding security mechanisms to satisfy the requirements [11]. Figure 2 shows how the SecSDM activities proposed by [11] are mapped to the Software Development Life Cycle.

The activities related to SecSDM is summarized in Table V.

V. MAPPING OF SECSDM TO SDL FOR ENGINEERS

It can be argued from the baseline study conducted in Section III that the same case can be made for engineering students than what was made for IT students – that undergraduate engineering programs neglect to address the importance of integrating information security into engineering design [11].

The Software Development Life Cycle and the Systems Development Life Cycle have many overlaps in common. Therefore, the adaptation of the SecSDM to the field of engineering can be motivated to be a good fit to provide engineering students with practical guidelines of how to identify the required security controls and how to practically implement the corresponding security mechanisms in the engineering product/system/process. The security activities mapping from the NIST SP 800-64 and the Information...
Assurance Technical Framework (IATF) frameworks can provide further guidance of where certain security activities from the SecSDM will fit into the SDLC. The adaptation of the SecSDM to the field of engineering to form E-SecSDM (Engineering SecSDM), is shown in Figure 3.

It can be seen from Figure 3 that the E-SecSDM activities remained identical to the original SecSDM activities. However, the names of the phases have been updated to reflect that of the typical engineering SDLC. It can also be seen that one additional phase is added to the process (retirement). Table 6 provides more detail on the E-SecSDM, describing each phase in parallel with the Key Engineering Activities relating to each phase. The Key Engineering Activities as well as the Key Security Engineering activities are adapted from Tables 3, 4 and 5.

To clarify the Key Security Engineering Activities shown in Table 6, each phase is briefly discussed using a relevant example within engineering. Comments are provided on how the engineering environment can differ from a typical software development environment as there might exist very specific security needs in this regard.

A. PHASE 0: CONCEPT EXPLORATION

In the exploration and conceptualization phase, the engineer is mainly concerned with the exploration of technology readiness, the pre-evaluation of the users’ needs and risk analysis. Alongside the evaluation of the users’ needs, the engineer must now determine the risks associated with this project. The conducted risk analysis can therefore be expanded to identify the information assets and the threats and vulnerabilities associated with each. The security requirements to mitigate the identified risks must then be determined. SecSDM suggests that information security requirements are stated in terms defined by ISO/IEC TR 13335-3, which includes confidentiality, integrity, availability, accountability, authenticity and reliability of information [19], [20].

When working with patient data and health records, one of the most important requirements remains confidentiality. However, with the design of a fitness tracking device, in addition to the recording, sharing and storing of health records, additional requirements must be considered. Previously, the patient and the healthcare professional provided data for the health record, where the device itself now automatically collects continuous data using electronic sensors (acceleration, heart rate, etc.). Automated data collection through sensors allows for cheating or the logging of incorrect or misrepresented data, which can lead to incorrect or unreliable health records. This means that reliability of information is an additional security requirement which must be considered in the design. Additionally, the continuous logging of the user’s location may lead to the ability to track the user without their knowledge. Confidentiality of live data, not simply records, must therefore also be considered.

B. PHASE 1: REQUIREMENTS & CONCEPTUALIZATION

Based on the pre-evaluation of needs and technology readiness, as well as the risk assessment, the engineer must recommend possible solutions and define full system requirements and product specifications. The risks identified in the previous phase will inform the detailed requirements and product specifications. The security requirements can be therefore used to inform which security services must be selected to mitigate the identified risks. [11] states that the identified security services must be carried out independent from implementation, supporting the process of engineering design where the detailed requirements and specifications are developed before any design or development takes place. SecSDM suggests that five basic security services, which includes identification and authentication, authorization/access control, confidentiality, integrity and nonrepudiation/non-denial be considered [19], [21].

The securing of health records might constitute the requirement of confidentiality and integrity services. When considering the addition of a wearable tracking device, the engineer might require the addition of more strict access control/authorization security services as the patient’s health records are now not only updated by the health practitioner, but also a fitness device and smartphone application. These security controls must ensure that the data produced by the tracking device is accurate and that it cannot be tampered with, requiring integrity of data.

C. PHASE 2: DESIGN & DEVELOPMENT

The main aim of the design and development phase is for engineers to design system architecture and allocate systems requirements to subsystems and components. The security services identified in the former stage must be defined and mapped to specific security mechanisms, which indicates how the services will be implemented. SecSDM recommends that the ISO 7498-2 standard’s security mechanisms be used, namely encryption, digital signatures, access control, data integrity, authentication exchange, traffic padding, routing control and notarization [19], [21].

For example, the requirement of health records to remain confidential will require the utilization of encryption as the security mechanism. In addition to the need for encryption, the fitness device requires the implementation of strict access control as well as data integrity mechanisms.

D. PHASE 3: PRODUCTION & IMPLEMENTATION

The implementation phase of the engineering design process involves the construction of subsystems, system integration...
and testing (hardware and software). When including the SecSDM activities in the SDLC, the engineer must implement the identified security mechanisms when software implementation is conducted. As engineers may not be skilled in secure coding standards or knowledgeable in best practices, the E-SecDSM should preferably identify possible security tools and components which can be utilized. The engineer should also be advised on coding standards and best practices to use when a specific security control is implemented.

When a fitness tracker is developed, the engineer should be guided to utilize software tools and components which include these security mechanisms to ensure that the data logged and recorded by the device is true and not tampered with. For testing and verification, the engineer would be advised to include security experts in this phase to test the implementation of the security controls to see if all subsystems adhere to coding standards and best practices and that it can withstand an attack.

E. PHASE 4: UTILIZATION & SUPPORT

During the utilization and support phase of the SDLC, the engineer must ensure that the product operates according to the users’ needs. When secure software practices are added to the utilization and support phase, the engineer must consider not only the functional operation of the software embedded on the device, but also the security aspects associated with it. Software/firmware must be continually evaluated to ensure that the system is secure as intended. It is also imperative to ensure that the user is not only educated on the functional use of the product, but also on how to use the product in a secure way.

If a user is provided with a wearable fitness tracker which connects to his/her smartphone and integrates to a medical database, all software functions must be kept up to date. Updating of the medical database might be the responsibility of a professional but ensuring an updated tracking device and smartphone would be the responsibility of the user.

F. PHASE 5: RETIREMENT

The SecSDM did not have a specific phase dedicated to the retirement of a project. In engineering, however, the retirement of a product leaves behind an artifact which may potentially contain private or sensitive information. The retirement of a fitness tracking device, for example, would require all data stored in the memory of the artifact itself to be erased. If a corresponding application was used on a smartphone, the privileges of that application must be revoked, and the application deleted. As with the previous phase, these actions must be executed by the user, which required education from the developers’ side informing the user of the best practices for disposal/retirement.

VI. CONCLUSION

This paper motivated the need for the integration of secure software practices into engineering design. Various proposals on the integration of security into the Systems Engineering
Development Lifecycle have been proposed in the literature, but limited work has been done on providing a practical approach on how to do so.

The adaptation of the SecSDM into the Systems Engineering Development Lifecycle can provide a practical approach for engineers to consider the active integration of security throughout the design of a product. The E-SecSDM can provide clear milestones for each phase of the SDL to assist engineers who may not be familiar with the importance of the inclusion of software security practices into engineering design.

The first proposed step to include software security practices into engineering design is to introduce final year engineering students conducting their capstone project to the integration of security into the Systems Engineering Development Lifecycle using E-SecSDM. Therefore, future work will include the creation of step-by-step practical guidelines based on the proposed E-SecSDM and a case study where engineering students will adopt these guidelines throughout an engineering design process.

However, it is known that many engineering programs still neglect to address the importance of the inclusion of security into the engineering design. Therefore, introducing engineering students to security and closing the knowledge gap related to cybersecurity in engineering have to be addressed at an early stage when students first learn software development. Another proposal is that the E-SecSDM guidelines must be integrated in earlier software design and programming modules in order to ensure that future engineers learn how to consider and include security within the engineering design process.

REFERENCES

[1] J. Wallen. (2019). Five nightmarish attacks that show the risks of IoT security. Accessed: Aug. 2, 2019. [Online]. Available: https://www.zdnet.com/article/5-nightmarish-attacks-that-show-the-risks-of-iot-security/

[2] A. Nordrum. (2016). Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated. Accessed: Feb. 2, 2020. [Online]. Available: https://spectrum.ieee.org/tech-talk/computer/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[3] J. Walden and C. E. Frank. “Secure software engineering teaching modules,” in Proc. 3rd Annu. Conf. Inf. Secur. Curriculum Develop., 2006, pp. 19–23.

[4] W. Conklin and G. Dietrich, “Secure software engineering: A new paradigm,” in Proc. 40th Annu. Hawaii Int. Conf. Syst. Sci. (HCIS), 2007, pp. 1–6.

[5] J. Davis. (2019). Microsoft Warns Hackers Targeting IoT Devices to Breach Networks. Accessed: Feb. 2, 2020. [Online]. Available: https://health.security.com/news/microsoft-warns-hackers-targeting-iot-devices-to-breach-networks

[6] A. Kiel. “What we know about ‘Industry 4.0’ so far” in Proc. Int. Assoc. Manage. Technol. (IAMOT), 2017, pp. 1–22.

[7] S. Morgan. (2005). IBM’s CEO On Hackers: ‘Cyber Crime Is The Greatest Threat To Every Com-Pany In The World’. Accessed: Feb. 2, 2020. [Online]. Available: https://www.forbes.com/sites/stevemorgan/2015/11/24/ibms-ceo-on-hackers-cyber-crime-is-the-greatest-threat-to-every-company-in-the-world/?sh=05337380

[8] E. Tamura. (2006). Hewlett Packard Enterprise Leads Transformation of Cyber Defense with ‘Build it In’ and ‘Stop it Now’. Accessed: Sep. 1, 2019. [Online]. Available: http://www8.hp.com/us/en/hp-news/press-release.html?id=2184147#.WiU5S6uyUI

[9] S. Von Solms and A. Marnewick. “Identifying security requirements body of knowledge for the security systems engineer,” in Towards a Cybersecurity Society: Education and Training (IFIP Advances in Information & Communication Technology), New York, NY, USA: Springer, 2019.

[10] R. Dove, J. Bayuk, B. Wilson, and K. Kepchar. INCOSE System Security Engineering Working Group Charter. Accessed: Apr. 21, 2020. [Online]. Available: https://www.incose.org/docs/default-source/wgcharters/systems-security-engineering.pdf?sfvrsn=cc6eb26_8

[11] L. Fletcher and R. von Solms, “SecSDM: A usable tool to support IT undergraduate students in secure software development,” in Proc. 6th Int. Symp. Hum. Aspects Inf. Secur. Assurance (HAISA), 2012, pp. 86–96.

[12] G. S. Parnell, P. J. Driscoll, and D. Henderson, Decision Making in Systems Engineering and Management. Hoboken, NJ, USA: Wiley, 2011, p. 497.

[13] S. Evans, D. Heinbuch, E. Kyle, J. Piorkowski, and J. Wallner, “Risk-based systems security engineering: Stopping attacks with intention,” IEEE Secur. Privacy Mag., vol. 2, no. 6, pp. 59–62, Nov. 2004.

[14] D. D. Walden, G. J. Roedder, K. J. Forsberg, R. D. Hamelin, and T. M. Shortell, INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Hoboken, NJ, USA: Wiley, 2015.

[15] IEEE Standard for Application and Management of the Systems Engineering Process, Standard 1220-2005, 2005.

[16] A. Ouyang, “CISSP common body of knowledge review: Software development security domain version: 5.9.” CISSP, San Francisco, CA, USA, Tech. Rep., 2012.

[17] Systems Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, document SP 800-160 1, NIST, 2016.

[18] National Security Agency/Central Security Service Information Assurance Technical Framework (IATF), document Release 3.1., 2002.

[19] L. Fletcher, L. and R. von Solms, “SecSDM: A model for integrating security into the software development life cycle,” Towards a Cybersecurity Society: Education and Training (IFIP Advances in Information & Communication Technology). New York, NY, USA: Springer, 2007.

[20] Information Technology—Guidelines for the Management of IT Security, Part 3—Techniques for the Management of IT Security, Standards ISO/IEC TR 13335-3, 1998.

[21] Information Processing Systems—Open System Interconnection—Basic Reference Model—Part 2: Security Architecture, Standards ISO 7498-2, 1989.

SUNE VON SOLMS (Member, IEEE) is currently an Associate Professor with the Faculty of Engineering and the Built Environment (FEBE), University of Johannesburg. She is also a Registered Professional Engineer with the Engineering Council of South Africa and an NRF Rated Researcher. She conducts research into the social and human aspects of cybersecurity, cybersecurity education and awareness, and the impact of technology in society. She is actively involved in cybersecurity education research, which includes the development of curricula as well as cybersecurity-related skills and competency development of engineers. She also collaborates with industry on a range of other cybersecurity education and awareness projects, including measuring the impact and effectiveness of cybersecurity training.

LYNN A. FUTCHER received the Ph.D. degree in information technology. She is currently an Associate Professor with the Faculty of Engineering, the Built Environment and Technology, School of IT, Nelson Mandela University, Port Elizabeth, South Africa. Her primary areas of research include information and cyber security education and secure software development, and as such are a Key Member of the Nelson Mandela University’s Centre for Research in Information and Cyber Security (CRICS). She also has a keen interest in IT Project Management, Human–Computer Interaction, and User Experience and Usable Security. She has been actively involved in the IFIP Working Group 11.8 (Information Security Education), since 2005, taking on several committee positions, including the Vice-Chair and a Secretary. As current Chair of this working group, she aims to promote Information Security Education, Training and Awareness in academia, government and industry by encouraging collaboration, and engagement amongst these key stakeholders.