INTERNATIONAL STANDARD **ISO/IEEE 11073-40101**

First edition
2022-03

Health informatics — Device interoperability —

Part 40101: Foundational — Cybersecurity — Processes for vulnerability assessment

Informatique de santé — Interopérabilité des dispositifs —

Partie 40101: Fondamentaux — Cybersécurité — Processus pour l'évaluation de la vulnérabilité

Reference number
ISO/IEEE 11073-40101:2022(E)

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted (see www.iso.org/directives).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

ISO/IEEE 11073-40101 was prepared by the IEEE 11073 Standards Committee of the IEEE Engineering in Medicine and Biology Society (as IEEE Std 11073-40101-2020) and drafted in accordance with its editorial rules. It was adopted, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Technical Committee ISO/TC 215, *Health informatics*.

A list of all parts in the ISO/IEEE 11073 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Health informatics—Device interoperability

Part 40101: Foundational—Cybersecurity—Processes for vulnerability assessment

Developed by the

IEEE 11073 Standards Committee
of the
IEEE Engineering in Medicine and Biology Society

Approved 24 September 2020

IEEE SA Standards Board

Abstract: For Personal Health Devices (PHDs) and Point-of-Care Devices (PoCDs), an iterative, systematic, scalable, and auditable approach to identification of cybersecurity vulnerabilities and estimation of risk is defined by this standard. The standard presents one approach to iterative vulnerability assessment that uses the Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) classification scheme and the embedded Common Vulnerability Scoring System (eCVSS). The assessment includes system context, system decomposition, pre-mitigation scoring, mitigation, and post-mitigation scoring and iterates until the remaining vulnerabilities are reduced to an acceptable level of risk.

Keywords: cybersecurity, embedded Common Vulnerability Scoring System, IEEE 11073-40101™, medical device communication, Personal Health Devices, Point-of-Care Devices, STRIDE, vulnerability assessment

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

PDF: ISBN 978-1-5044-7086-5 STD24423
Print: ISBN 978-1-5044-7087-2 STDPD24423

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system. An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; https://www.copyright.com/. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website. Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this standard was submitted to the IEEE SA Standards Board for approval, the Public Health Device Working Group had the following membership:

Daidi Zhong, *Chair*
Michael Kirwan and **Christoph Fischer**, *Vice Chairs*

Karsten Aalders
Charles R. Abbruscato
Nabil Abujbara
Maher Abuzaid
James Agnew
Manfred Aigner
Jorge Alberola
David Aparisi
Lawrence Arne
Diego B. Arquillo
Serafin Arroyo
Muhammad Asim
Kit August
Doug Baird
David Baker
Anindya Bakshi
Abira Balanadarasan
Ananth Balasubramanian
Sunlee Bang
M. Jonathan Barkley
Gilberto Barrón
David Bean
John Bell
Olivia Bellamou-Huet
Rudy Belliardi
Daniel Bernstein
George A. Bertos
Chris Biernacki
Ola Björsne
Thomas Blackadar
Thomas Bluethner
Douglas P. Bogia
Xavier Boniface
Shannon Boucousis
Julius Broma
Lyle G. Bullock, Jr.
Bernard Burg
Chris Burns
Jeremy Byford-Rew
Satya Calloji
Carole C. Carey
Craig Carlson
Santiago Carot-Nemesio
Randy W. Carroll
Seungchul Chae
Peggy Chien
David Chiu
Jinyong Choi
Chia-Chin Chong
Saeed A. Choudhary
Jinhan Chung
John A. Cogan
John T. Collins
Cory Condek
Todd H. Cooper
David Cornejo
Douglas Coup
Nigel Cox
Hans Crommenacker
Tomio Crosley
Allen Curtis
Jesús Daniel Trigo
David Davenport
Russell Davis
Sushil K. Deka
Ciro de la Vega
Pedro de-las-Heras-Quiros
Jim Dello Stritto
Kent Dicks
Hyoungho Do
Jonathan Dougherty
Xiaolian Duan
Sourav Dutta
Jakob Ehrensvard
Fredrik Einberg
Javier Escayola Calvo
Mark Estes
Leonardo Estevez
Bosco T. Fernandes
Morten Flintrup
Joseph W. Forler
Russell Foster
Eric Freudenthal
Matthias Frohner
Ken Fuchs
Jing Gao
Marcus Garbe
John Garguilo
Liang Ge
Rick Geimer
Igor Gejdos
Ferenc Gerbovics
Alan Godfrey
Nicolae Goga
Julian Goldman
Raul Gonzalez Gomez
Chris Gough
Channa Gowda
Charles M. Gropper
Amit Gupta
Jeff Guttmacher
Rasmus Haahr
Christian Habermann
Michael Hagerty
Jerry Hahn
Robert Hall
Shu Han
Nathaniel Hamming
Rickey L. Hampton
Sten Hanke
Aki Harma
Jordan Hartmann
Kai Hassing
Avi Hauser
Wolfgang Heck
Nathaniel Heintzman
Charles Henderson
Jun-Ho Her
Helen B. Hernandez
Timothy L. Hirou
Allen Hobbs
Alex Holland
Arto Holopainen
Kris Holtzclaw
Robert Hoy
Anne Huang
Zhiyong Huang
Ron Huby
David Hughes
Robert D. Hughes
Jiyoung Huh
Hugh Hunter
Philip O. Isaacson
Atsushi Ito
Michael Jaffe
Praduman Jain
Hu Jin
Danny Jochelson
Akiyoshi Kabe
Steve Kahle
Tomio Kamioka
James J. Kang
Kei Kariya
Andy Kaschl
Junzo Kashihara
Colin Kennedy
Ralph Kent
Laurie M. Kermes
Ahmad Kheirandish
Junhyung Kim
Minho Kim
Min-Joon Kim
Taekon Kim
Tetsuya Kimura
Alfred Kloos
Jeongmee Koh

Jean-Marc Koller
John Koon
Patty Krantz
Raymond Krasinski
Alexander Kraus
Ramesh Krishna
Geoffrey Kruse
Falko Kuester
Rafael Lajara
Pierre Landau
Jaechul Lee
JongMuk Lee
Kyong Ho Lee
Rami Lee
Sungkee Lee
Woojae Lee
Qiong Li
Xiangchen Li
Zhuofang Li
Patrick Lichter
Jisoon Lim
Joon-Ho Lim
Xiaoming Liu
Wei-Jung Lo
Charles Lowe
Don Ludolph
Christian Luszick
Bob MacWilliams
Srikkanth Madhurbootheswaran
Miriam L. Makhlouf
Romain Marmot
Sandra Martinez
Miguel Martínez de Espronceda Cámara
Peter Mayhew
Jim McCain
László Meleg
Alexander Mense
Behnaz Minaei
Jinsei Miyazaki
Erik Moll
Darr Moore
Chris Morel
Robert Moskowitz
Carsten Mueglitz
Soundharya Nagasubramanian
Alex Neefus
Trong-Nghia Nguyen-Dobinsky
Michael E. Nidd
Jim Niswander
Hiroaki Niwamoto
Thomas Norgall
Yoshiteru Nozoe
Abraham Ofek
Brett Olive
Begonya Otal
Marco Paleari
Bud Panjwani
Carl Pantiskas
Harry P. Pappas
Hanna Park
Jong-Tae Park
Myungeun Park
Soojun Park
Phillip E. Pash
TongBi Pei
Soren Petersen
James Petisce
Peter Piction
Michael Pliskin
Varshney Prabodh
Jeff Price
Harald Prinzhorn
Harry Qiu
Tanzilur Rahman
Phillip Raymond
Terrie Reed
Barry Reinhold
Brian Reinhold
Melvin I. Reynolds
John G. Rhoads
Jeffrey S. Robbins
Chris Roberts
Stefan Robert
Scott M. Robertson
Timothy Robertson
David Rosales
Bill Saltzstein
Giovanna Sannino
Jose A. Santos-Cadenas
Stefan Sauermann
John Sawyer
Alois Schloegl
Paul S. Schluter
Mark G. Schnell
Richard A. Schrenker
Antonio Scorpiniti
KwangSeok Seo
Riccardo Serafin
Sid Shaw
Frank Shen
Min Shih
Mazen Shihabi
Redmond Shouldice
Sternly K. Simon
Marjorie Skubic
Robert Smith
Ivan Soh
Motoki Sone
Emily Sopensky
Rajagopalan Srinivasan
Nicholas Steblay
Lars Steubesand
John (Ivo) Stivoric
Raymond A. Strickland
Chandrasekaran Subramaniam
Hermanni Suominen
Lee Surprenant
Ravi Swami
Ray Sweidan
Na Tang
Haruyuyki Tatsumi
Isabel Tejero
Tom Thompson
Jonas Tirén
Janet Traub
Gary Tschautscher
Masato Tsuchid
Ken Tubman
Akib Uddin
Sunil Unadkat
Fabio Urbani
Philipp Urbauer
Laura Vanzago
Alpo Värri
Andrei Vasilateanu
Dalimar Velez
Martha Velezis
Rudi Voon
Barry Vornbrock
Isobel Walker
David Wang
Linling Wang
Jerry P. Wang
Yao Wang
Yi Wang
Steve Warren
Fujio Watanabe
Toru Watsuji
David Weissman
Kathleen Wible
Paul Williamson
Jan Wittenber
Jia-Rong Wu
Will Wykeham
Ariton Xhafa
Ricky Yang
Melanie S. Yeung
Qiang Yin
Done-Sik Yoo
Zhi Yu
Jianchao Zeng
Jason Zhang
Jie Zhao
Thomas Zhao
Yuanhong Zhong
Qing Zhou
Miha Zoubek
Szymon Zyskoter

The following members of the individual balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Robert Aiello
Johann Amsenga
Bjoern Andersen
Pradeep Balachandran
Demetrio Bucaneg, Jr.
Lyle G. Bullock, Jr.
Craig Carlson
Juan Carreon
Pin Chang
Malcolm Clarke
Christoph Fischer
David Fuschi
Randall Groves
Robert Heile
Werner Hoelzl
Raj Jain
Martin Kasparick
Stuart Kerry
Edmund Kienast
Yongbum Kim
Raymond Krasinski
Javier Luiso
H. Moll
Nick S. A. Nikjoo
Bansi Patel
Dalibor Pokrajac
Beth Pumo
Stefan Schlichting
Thomas Starai
Mark-Rene Uchida
John Vergis
J. Wiley
Yu Yuan
Oren Yuen
Janusz Zalewski
Daidi Zhong

When the IEEE SA Standards Board approved this standard on 24 September 2020, it had the following membership:

Gary Hoffman, *Chair*
Jon Walter Rosdahl, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Ted Burse
Doug Edwards
J. Travis Griffith
Grace Gu
Guido R. Hiertz
Joseph L. Koepfinger*
David J. Law
Howard Li
Dong Liu
Kevin Lu
Paul Nikolich
Damir Novosel
Dorothy Stanley
Mehmet Ulema
Lei Wang
Sha Wei
Philip B. Winston
Daidi Zhong
Jingyi Zhou

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 11073-40101-2020, Health informatics—Device interoperability—Part 40101: Foundational—Cybersecurity—Processes for vulnerability assessment.

Users of Personal Health Devices (PHDs) and Point-of-Care Devices (PoCDs) have implicit expectations on convenience, connectivity, accessibility, and security of data. For example, they expect to connect PHDs/PoCDs to their mobile devices and dashboards, view the data in the cloud, and easily share the information with clinicians or care providers. In some cases, the users themselves are taking action to build connections between PHDs/PoCDs, mobile devices, and the cloud to create the desired system. While many manufacturers are working on solving PHD/PoCD connectivity challenges with proprietary solutions, no standardized approach exists to provide secure plug-and-play interoperability.

The ISO/IEEE 11073 PHDs/PoCDs family of standards, Bluetooth Special Interest Group profiles and services specifications, and the Continua Design Guidelines (PCHAlliance [B7]) were developed to specifically address plug-and-play interoperability of PHDs/PoCDs (e.g., physical activity monitor, physiological monitor, pulse oximeter, sleep apnoea breathing therapy equipment, ventilator, insulin delivery device, infusion pump, continuous glucose monitor). In this context, the following terms have specific meanings:

— *Interoperability* is the ability of client components to communicate and share data with service components in an unambiguous and predictable manner as well as to understand and use the information that is exchanged (PCHAlliance [B7]).
— *Plug and play* are all the user has to do to make a connection—the systems automatically detect, configure, and communicate without any other human interaction (ISO/IEEE 11073-10201 [B5]).[1]

Within the context of *secure* plug-and-play interoperability, cybersecurity is the process and capability of preventing unauthorized access or modification, misuse, denial of use, or the unauthorized use of information that is stored on, accessed from, or transferred to and from a PHD/PoCD. This standard describes the process part of cybersecurity for transport-independent applications and information profiles of PHDs/PoCDs. These profiles define data exchange, data representation, and terminology for communication between agents (e.g., pulse oximeters, sleep apnoea breathing therapy equipment) and connected devices (e.g., health appliances, set top boxes, cell phones, personal computers, monitoring cockpits, critical care dashboards).

For PHDs/PoCDs, this standard defines an iterative, systematic, scalable, and auditable approach to identification of cybersecurity vulnerabilities and estimation of risk. This standard presents one approach to iterative vulnerability assessment that uses the Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) classification scheme and the embedded Common Vulnerability Scoring System (eCVSS). The assessment includes system context, system decomposition, pre-mitigation scoring, mitigation, and post-mitigation scoring and iterates until the remaining vulnerabilities are reduced to an acceptable level of risk.

[1] The numbers in brackets correspond to the numbers of the bibliography in Annex A.

Contents

Health informatics—Device interoperability

Part 40101: Foundational—Cybersecurity—Processes for vulnerability assessment

1. Overview

1.1 General

Many Personal Health Devices (PHDs) and Point-of-Care Devices (PoCDs) provide vital support for people living with chronic disease or experiencing a life-threatening medical event. Cybersecurity attacks on vulnerable devices may lead to the alteration of prescribed therapy (e.g., sleep apnoea breathing therapy, insulin therapy) or to information disclosure that results in insurance or identity fraud or in direct or indirect patient harm. Companies subject to a successful cybersecurity attack may suffer financial harm and a negative reputation.

Manufacturers of regulated PHDs/PoCDs are required to address cybersecurity vulnerabilities through a detailed risk analysis of use cases specific to the device. Of the various approaches to vulnerability assessment, some are not repeatable, scalable, systematic, and auditable. Both manufacturers and regulatory bodies may benefit from a common approach to vulnerability assessment based on threat modeling capable of analyzing PHDs/PoCDs across domains and described in a trusted open consensus standard. Likewise, patients, providers, and payers benefit from consistent and sufficient information provided in PHD/PoCD labeling.

This standard is based on the PHD Cybersecurity Standards Roadmap findings (IEEE white paper [B4]) and presents a repeatable, scalable, systematic, and auditable approach to vulnerability assessment.[2] While a specific approach is provided, any comparable approach is appropriate and will be compatible with the mitigations found in IEEE Std 11073-40102™ [B3]. In Figure 1, this standard is depicted by the top row, and IEEE Std 11073-40102 is depicted by the bottom row.

[2] The numbers in brackets correspond to the numbers of the bibliography in Annex A.

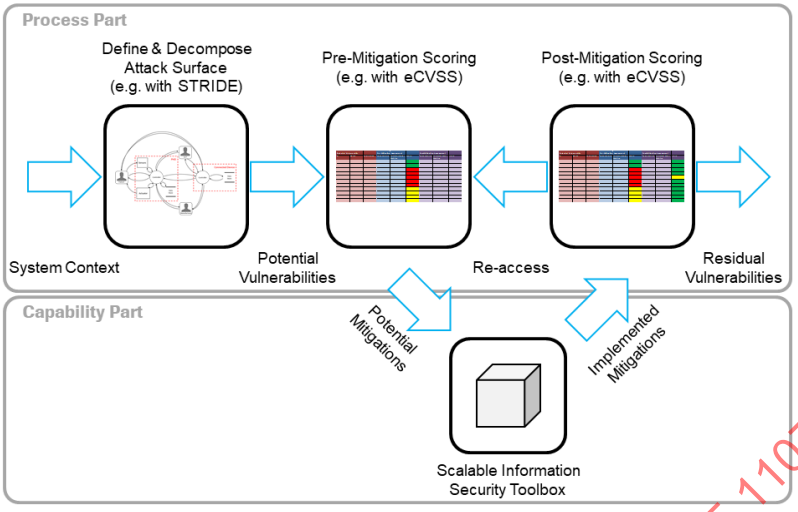

Figure 1—Vulnerability assessment workflow

1.2 Scope

Within the context of secure plug-and-play interoperability, cybersecurity is the process and capability of preventing unauthorized access or modification, misuse, denial of use, or the unauthorized use of information that is stored on, accessed from, or transferred to and from a PHD/PoCD. The process part of cybersecurity is risk analysis of use cases specific to a PHD/PoCD.

For PHDs/PoCDs, this standard defines an iterative, systematic, scalable, and auditable approach to identification of cybersecurity vulnerabilities and estimation of risk. This iterative vulnerability assessment uses the Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) classification scheme and the embedded Common Vulnerability Scoring System (eCVSS). The assessment includes system context, system decomposition, pre-mitigation scoring, mitigation, and post-mitigation scoring and iterates until the remaining vulnerabilities are reduced to an acceptable level of risk.

1.3 Purpose

The purpose of this document is to define a common approach to cybersecurity assessment in PHDs/PoCDs and define an iterative, systematic, scalable, and auditable vulnerability assessment appropriate for use in the design of PHDs/PoCDs.

1.4 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals is *required to*).[3,4]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals is *recommended that*).

[3] The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.
[4] The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is used only in statements of fact.

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals is *permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals is *able to*).

2. Definitions, acronyms, and abbreviations

2.1 Definitions

For the purposes of this document, the terms and definitions provided in the PHD Cybersecurity Standards Roadmap (IEEE white paper [B4]) apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined there.[5]

2.2 Acronyms and abbreviations

CRUD	create, read, update, and delete
CVSS	Common Vulnerability Scoring System
DFD	data flow diagram
eCVSS	embedded Common Vulnerability Scoring System
HCP	Health Care Provider
PHD	Personal Health Device
PoCD	Point-of-Care Device
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privileges
TMT	Threat Modeling Tool
UML	Unified Modeling Language

3. Risk management

Various regulations, standards, and guidelines address the subject of risk and risk management. In some cases, the application of specific standards may be mandated by regulations, contracts, or customer expectations. This standard does not define a specific risk management process as appropriate for all manufacturers because each manufacturer's risk management process needs to comply with the regulations, standards, contracts for the specific disease domain, and the jurisdiction in which the device is marketed.

Instead, this standard presents a repeatable, scalable, systematic, and auditable approach to vulnerability assessment that is adaptable to various mandates and can be used within a PHD/PoCD risk management process when evaluating PHD/PoCD communication. The assessment identifies and prioritizes vulnerabilities based on device use cases and, through iteration, helps to minimize reasonably foreseeable risks associated with PHD/PoCD communication to an acceptable level. It is the responsibility of the manufacturer's management to define the appropriate acceptable level. In the PHD/PoCD domain, there are fitness devices with low information security concerns and disease management devices with higher information security concerns. Therefore, the risk management process is based on the device's intended use cases within a specific domain, which represent a wide variance where high-risk PHDs/PoCDs represent the upper limit. As such, the risk evaluation of a PHD/PoCD with fewer information security concerns may identify only a subset of vulnerabilities.

[5] *IEEE Standards Dictionary Online* is available at https://dictionary.ieee.org. An IEEE account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

4. Software of unknown provenance

The development of a PHD/PoCD is similar to the development of any device or system where manufacturers implement within their domain of specialties and otherwise include third-party solutions. If these third-party or off-the-shelf (OTS) solutions are software, they are known as *software of unknown provenance* (SOUP) (FDA [B1]). Since PHD/PoCD manufacturers are held responsible for any harm that occurs from the safety and efficacy of the design and intended use of their devices, they are also responsible for the SOUP within their devices.

One way to manage SOUP within a system is for the manufacturer to verify each version of the SOUP included with the system. Alternatively, the manufacturer could include additional information security controls within the system to protect the system from potential vulnerabilities of the SOUP. This standard for vulnerability assessment addresses the latter as the assessment considers all components within the PHD/PoCD and related systems. The benefit of including SOUP in the vulnerability assessment and mitigating any identified vulnerabilities with specific security controls is that it reduces the burden on the manufacturer when verifying the SOUP.

5. Multi-component system vulnerability assessment

As the connectivity of PHDs/PoCDs increases, multi-component, heterogeneous systems become more widespread. A multi-component system includes multiple connected components from potentially various manufacturers either within a single device or as a system (e.g., system of systems), where at least one component is a PHD/PoCD. Examples of multi-component systems include an automated insulin delivery system, patient monitor, and cloud service that gathers data from PHDs/PoCDs to support consumers, providers, or payers. An automated insulin delivery system connects a continuous glucose monitor and insulin pump to an automated dosing controller. A patient monitor typically includes devices such as a thermometer, blood pressure monitor, and pulse oximeter. Also, it is important to note that as the connectivity increases the intended use is subject to change.

The vulnerability assessment described in this standard is applicable to a multi-component system. Both in a standalone PHD/PoCD or as part of multi-component system, the following assumptions must be made: the environment is hostile, and the PHD/PoCD does not know the inner workings of the connected device(s). Thus, the PHD/PoCD should not trust the other connected device(s) implicitly. Instead, security is the responsibility of the manufacturer of the PHD/PoCD interfaces, which should be well described and without hidden functionality. Also, the security of one component of the system should not depend on the security of another. Each component of the system, by itself, should provide sufficient security to protect against direct attacks or chaining of attacks. Assessing each component individually for vulnerabilities without any inherent trust between the components of the system greatly improves the security of the system as a whole (i.e., zero trust). This assessment should not be omitted even when the multiple components were intended to work together.

6. Threat modeling

6.1 General

Threat modeling is an approach of analyzing the security of a system (e.g., device, application) or a system of systems (e.g., multi-component system) so that vulnerabilities can be identified, enumerated, and prioritized. Threat modeling typically employs a systematic approach to identifying assets most desired by an attacker and related attack vectors. This step leads to the decomposition of the system by investigating each asset and attack vector individually and determining the kind of attacks to which they are vulnerable. From this effort, a list of vulnerabilities is created for the system and ordered in terms of risk, potential to cause harm, or any other criteria deemed appropriate.

6.2 Data flow diagram

One approach to system decomposition is the creation of a data flow diagram (DFD). DFDs are typically used to graphically represent a system, but different representation may also be used [such as a UML diagram (Unified Modeling Language) or an SysML diagram (System Modeling Language)]. In any case, the same basic method is applied: decompose the system into parts and determine the kinds of attack to which the parts may be vulnerable, with related risk and harm. DFDs consist of five elements:

- **Data flows** represent data in motion over system interfaces.
- **Data stores** represent data at rest within the system.
- **Processes** create, read, update, or delete data and are typically applications run within the system.
- **Interactors** are the end points of the system (e.g., end-user) and generally are providers and consumers that are outside the scope of the system.
- **Trust boundaries** represent the borders between trusted and untrusted elements of the DFD.

See Figure 3 (in 8.3.3) for an example of a DFD.

6.3 STRIDE classification scheme

STRIDE is a classification scheme for characterizing identified threats during the development process according to the kinds of exploits that are used by the attacker. The STRIDE acronym is formed from the first letter of each of the following threat categories: **S**poofing, **T**ampering, **R**epudiation, **I**nformation **D**isclosure, **D**enial of Service, and **E**levation of Privilege.

In the context of PHD/PoCD communication and this standard, the PHD/PoCD is modeled based on the use case description and focuses on data flow between processes and external actors. An actor is a person or organization unit playing a coherent set of roles when interacting with the system within a particular use case. The model identifies the threat surfaces to the device, which are used for system decomposition. Annex B provides additional description of STRIDE.

A variant of STRIDE includes lateral movement (LM) as a threat category (Lockheed Martin [B6]). Lateral movement refers to the technique attackers use to move through a network looking for desirable assets. Because the focus of this standard is on the interface to and from the PHD/PoCD, STRIDE-LM is not a vulnerability assessment option. However, when considering a direct-to-cloud interface, lateral movement has an important role.

7. Scoring system

7.1 General

A systematic and repeatable method of quantifying vulnerabilities is typically achieved via a scoring system, which provides a rank and priority to each of the identified vulnerabilities based on specific criteria.

7.2 CVSS

While various scoring systems exist, the Common Vulnerability Scoring System (CVSS) has become a widely accepted method for quantifying the severity of vulnerabilities. The CVSS is an open industry standard for normalized scoring of vulnerabilities across disparate software platforms. Assigning scores to vulnerabilities allows the prioritization to guide which of the vulnerabilities should be mitigated.

The CVSS assessment is composed of metrics for the following three areas of concern:

- **Base metrics:** Represent the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.
- **Temporal metrics:** Represent the characteristics of a vulnerability that change over time but not among user environments.
- **Environmental metrics:** Represent the characteristics of a vulnerability that are relevant and unique to a particular user's environment.

7.3 eCVSS

The original version of CVSS was designed to address software-only systems and create scores after the software systems were in the field. It was designed to allow for the use of the base metrics along with additions or adjustments. In the effort to develop this standard, changes to CVSS were needed to support physical PHDs/PoCDs and create scores during design to guide the development of systems. As a result, the embedded Common Vulnerability Scoring System (eCVSS) was created as a slightly modified branch of CVSS 2.0 (FIRST [B2]). This creation is not an effort of the Forum of Incident Response and Security Teams (FIRST), but is instead proposed by the working group that developed this standard.

The eCVSS modifications to CVSS are as follows:

- The Temporal Group was effectively removed by forcing the three metrics to a neutral value because the scoring in eCVSS is conducted during design.
- The three "Requirement" metrics in the Environmental Metric Group (i.e., Confidential, Integrity, Availability) were recognized to be system wide. These metrics are set only once for the system and then applied to all the identified vulnerabilities.
- The Target Distribution metric was removed as it refers to distribution of the system, and the scoring in eCVSS is conducted during design. Instead, a new Awareness metric in the Environmental Metric Group has replaced the Target Distribution metric.

A complete description of eCVSS along with scoring equations is provided in Annex C.

eCVSS has no explicit metric that an attack would occur and be successful, e.g., likelihood. Instead, if an attack can be successful, it will be successful. Therefore, mitigations that reduce the probability of a successful attack often result in the same post-mitigation risk score. In some cases it could be argued that the threat could not possibly happen, and such a conclusion would result in a score of zero. In other cases, such as when using authorization that could possibly be sidestepped, the Awareness metric usually would have to reach at least a score of **User** to be considered. Some vulnerability assessment methodologies may consider the likelihood of an event to be reduced after mitigation, e.g., by reducing the window of opportunity of the attack.

Within eCVSS, various metrics require the analysis to select a value of **None**, **Partial**, or **Complete**. For users who employ this approach within their organization, it is important to note the difference between **Partial** and **Complete**. Briefly, **Partial** means that the attacker can successfully attack the system, but not with the desired granularity. As example, the attack may be to gather data from the system, but cannot specify which data is received. On the other hand, **Complete** means the attacker has full control of the attack.

A vulnerability that may include direct or indirect harm to an end-user is identified with eCVSS in two ways. First, each vulnerability has a yes/no entry to identify the vulnerabilities that impact the safety efficacy of the system. This step allows filtering of the analysis to include these vulnerabilities in the risk management process as they may be subject to non-cybersecurity mitigations. Second, the Collateral Damage Potential metric considers the patient along with business and legal damages. The weighting of the Collateral Damage Potential is significant in that even seemingly basic vulnerabilities with direct end-user harm will receive a higher risk score.

The assessment rates each vulnerability as low risk, moderate risk, or high risk. The assessment includes risk threshold levels to distinguish vulnerabilities. Users who wish to follow this approach should evaluate and assign appropriate risk threshold levels for the system under assessment.

8. Process for vulnerability assessment

8.1 Iterative vulnerability assessment

According to this standard, the processes for vulnerability assessment of a system shall be iterative and shall include at least the following steps:

1) **System context:** Represented by a use case description.
2) **System decomposition:** Modeling of the system using a DFD that is analyzed using STRIDE to generate a list of vulnerabilities.
3) **Scoring (pre-mitigation):** Quantifying each identified vulnerability using eCVSS.
4) **Mitigation:** Mitigating vulnerabilities with unacceptable risk using information security controls.
5) **Scoring (post-mitigation):** Quantifying the mitigated vulnerabilities using eCVSS to determine if the risk was reduced to an acceptable level.
6) **Iteration:** Repeat from step 4 until all vulnerabilities have been reduced to an acceptable level of risk.

The steps are depicted in Figure 1 (in 1.1) and discussed in 8.2 through 8.6.

8.2 System context

8.2.1 Use case description

System context is step 1 in the iterative vulnerability assessment. The system context is provided in a use case description, which includes an introduction to, and system context of, the PHD/PoCD. The use case description contains the name, a brief description, a block diagram, a list of intended actors, a list of assets, and a mapping between intended actors and assets along with any relevant references. See Annex E for an example use case description.

8.2.2 Actors

The PHDs/PoCDs could have up to fifteen actor roles from seven categories (see Figure 2).

a) **Manufacturer:** Trained individuals or group working for an enterprise (e.g., legally responsible distributing company, supplier) handling the complete, or participating in a part of the, systems' life cycle from conception to development to maintenance to end of life.
   1) R&D Engineer: Plans, designs, specifies, develops, and/or tests the system components as part of research and development (R&D).
   2) Supplier: Supply chain employee who builds out of raw materials and/or sub-components the system components in a factory.
   3) Technical Supporter: Helps customers with complaints about system components and creates bridge to R&D (e.g., service and investigation).
   4) Seller: Demonstrates the system to individuals (e.g., patient, payer) who decide, or have influence about, whether the system is used.
   5) Trainer: Provides training about the system to the Health Care Provider (HCP) (who will train the end-user) or directly to the end-user.

b) **Operator:** Trained individuals or group working for an enterprise (e.g., clinic, medical practice, diabetes center) supporting the work of the HCP by managing the equipment.
   1) IT Network Professional: Responsible for information technology (IT), i.e., the enterprise network and computing facilities.
   2) IT Security Professional: Responsible for securing the enterprise network and computing facilities.
   3) Biomedical Engineer: Responsible for configuring, testing, and maintaining the system components owned by, or used within, the perimeter of the enterprise.

c) **Business User:** Trained individuals or group working for an enterprise (e.g., health insurance company, governmental organization, health care supply store) participating in the supply chain.
   1) Payer: Pays for the system components and services. Depending on the local rules, this actor decides, or is part of the decision on, whether the system is used.
   2) Distributor: Brings the system components from the manufacturer to the end-user. Depending on the local rules, this actor decides, or is part of the decision on, whether the system is used.

d) **Health Care Provider (HCP):** Trained individuals or group working for an enterprise (e.g., clinic, medical practice, diabetes center) interacting with the patient and operating the system components for demonstration purposes or to adapt the therapy.
   1) Counselor: Typically looks after patients on a short-cycle basis and helps them to manage their disease via training and consulting (e.g., diabetes nurse educator). Depending on the local rules, this actor decides, or is part of the decision on, whether the system is used.
   2) Nurse: Supports the physician and helps patients during their interactions within the HCP perimeter. Depending on the local rules, this actor decides, or is part of the decision on, whether the system is used.
   3) Physician: Diagnoses disease and sets up initial therapy. Authorizes systems that are subject to prescription. After therapy is running, typically looks after patients on a long-cycle basis to adapt the therapy. Depending on the local rules, this actor decides, or is part of the decision on, whether the system is used.

e) **End-User:** Individuals who typically use the system.
   1) Patient: Uses the system components as part of overall therapy regimen (e.g., person with diabetes). Patients should generally have elementary knowledge about the disease and knowledge about personal hygiene. Depending on the local rules, this actor decides, or is part of the decision on, whether the system is used.
   2) Caregiver: Cares for patients who are sick or disabled. Children, elderly persons, or handicapped patients may need assistance provided by a reliable caregiver (e.g., parents). These caregivers should have knowledge equivalent to the level required for patients.

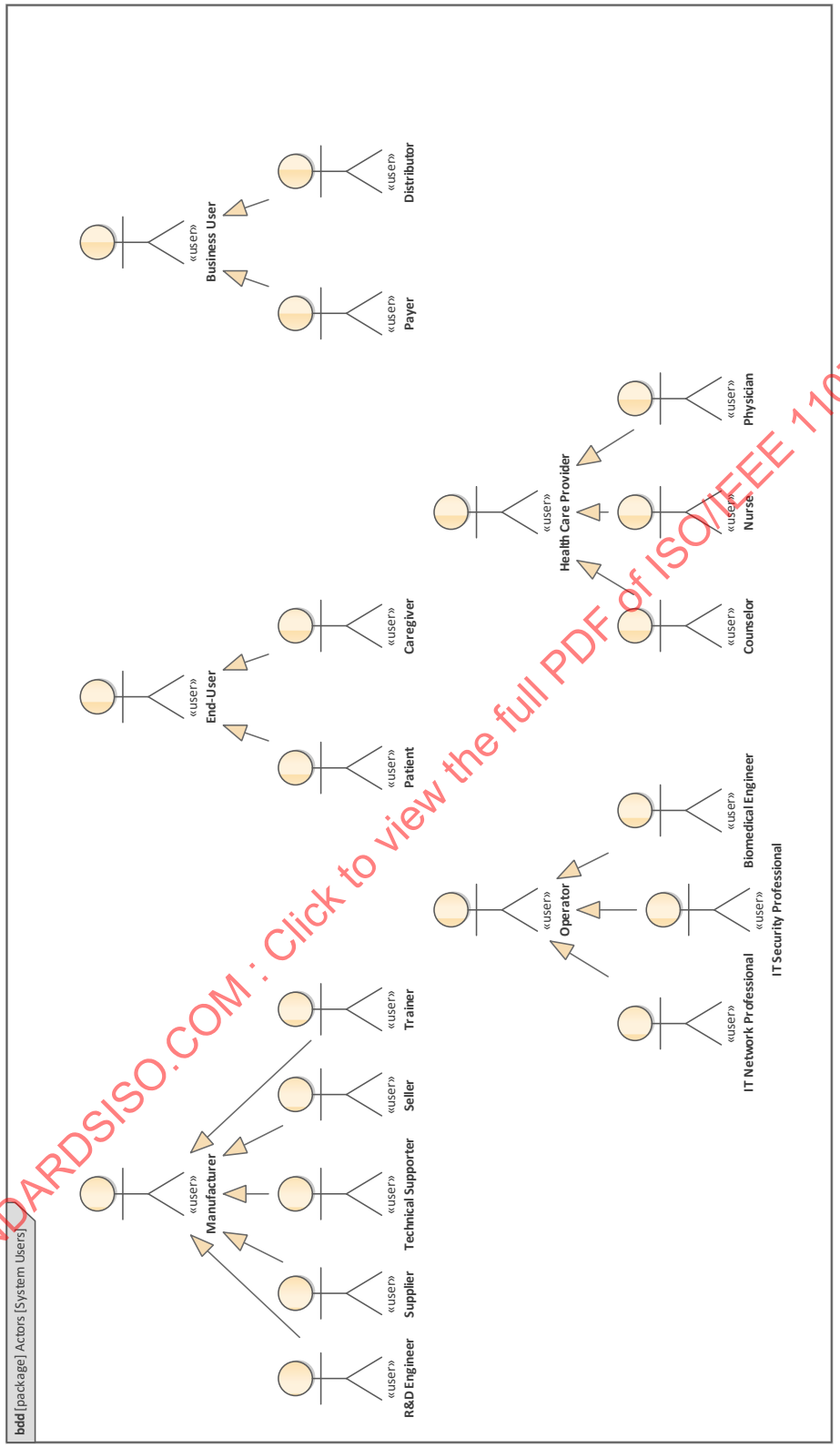

Figure 2—Actors

8.2.3 Assets

For the intended actors, the PHD/PoCD controls, stores, and transmits various assets (i.e., the data asset inventory). However, these assets may potentially interest an attacker. The assets of interest are listed in Table 1.

Table 1—Assets

Name	Description
Credentials	Login credentials [e.g., usernames, passwords, tokens, personal identification numbers (PINs), wireless secure codes].
Therapy data	Therapy relevant data (e.g., treatment settings, measurements).
Device data	Non-therapy relevant data (e.g., language selection).
Logs	History of actions executed by the PHD/PoCD that provides support to the manufacturer, HCP, etc.
Annunciation	Information from a PHD/PoCD telling the user that the PHD/PoCD needs attention (e.g., status, reminder, error, warning, maintenance).
Device control	Control over the device functionalities.
Firmware/software application	The application running on or in the device.
Intellectual property	The intellectual property within the device.
Protected health information	Any information about health status, provision of health care, or payment for health care that can be linked to a specific individual.

8.2.4 Mapping actors to assets

The mapping between the actors and assets uses the four basic actions that can be conducted on an asset: create, read, update, and delete, also known as CRUD. The mapping describes intended actions that a specific actor would typically perform. For example, the manufacturer may “create” device information (e.g., serial number, regional configurations) while the HCP or end-user may “read” the device information.

8.3 System decomposition

8.3.1 General

System decomposition is step 2 in the iterative vulnerability assessment. The data flows of the PHD/PoCD are modeled, and threat surfaces are identified. The model of the data flow then undergoes a system decomposition to create a list of vulnerabilities.

8.3.2 Trust boundaries

As part of modeling the device type, boundaries are used to define areas of inherent trust. A trust boundary is depicted as a dotted box and, when used in a model, indicates that both sides of the boundary do not trust each other. As such, when a data flow crosses a trust boundary, it becomes an untrusted data flow and generates applicable vulnerabilities. The PHD/PoCD and connected device use trust boundaries to show contained processes, actuators, sensors, data flows, and data stores. As mentioned previously, the assessment should not omit elements within a trust boundary even when the multiple components were intended to work together.

8.3.3 Threat model

Threat modeling of a PHD/PoCD is realized by making a DFD, which identifies the external actors, system processes, interactors, and internal storage and the data flow interfaces between those elements. External

actors are typically the human users of the system (e.g., patient, HCP). System processes are controllers of the system (e.g., delivery controller of an insulin delivery device that determines how much insulin should be delivered, and when, and then controls the pump drive). Interactors are the sensor and actuators of the system, (e.g., pump drive of an insulin delivery device). Internal storage is the data storage within the device (e.g., stored therapy settings, observations). Figure 3 depicts a harmonized nomenclature and layout as a generic DFD for a PHD/PoCD. Note that trust boundaries are depicted with a dotted line box.

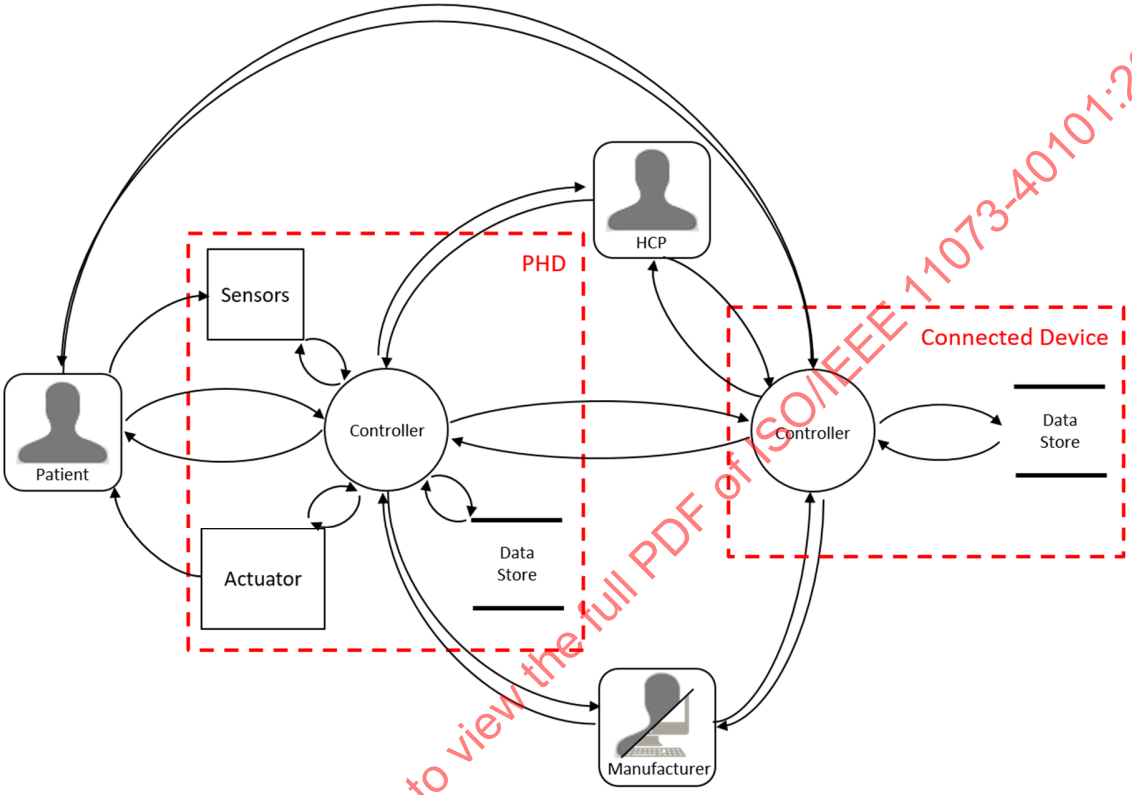

Figure 3—PHD/PoCD generic threat model

8.3.4 Vulnerability list

Once the device type is modeled and the threat surfaces have been identified, it is possible to decompose the system using a specific framework to generate a list of vulnerabilities. STRIDE has been chosen for this standard.

Each threat category of STRIDE generates vulnerabilities based on the elements of the DFD. See Table B.2 for a mapping of DFD elements to STRIDE threat categories. Each of these STRIDE threat categories can be further broken down into vulnerability types, as listed in Table B.3. While most of these vulnerability types are applicable to this standard, the following vulnerability types were defined as out of the scope of this standard:

- Elevation of Privilege: Cross Site Request Forgery—not applicable to PHD/PoCD use cases
- Spoofing: Spoofing of Source Data Store—internal interfaces are not currently considered
- Spoofing: Spoofing of Destination Data Store—internal interfaces are not currently considered
- Information Disclosure: Weak Access Control for a Resource—internal interfaces are not currently considered
- Denial of Service: Data Flow is Potentially Interrupted, when the data flow is internal to the device—internal interfaces are not currently considered

8.4 Scoring

8.4.1 General

Scoring is step 3 and step 5 in the iterative vulnerability assessment. eCVSS is the selected scoring system to quantify identified vulnerabilities, both as part of the pre-mitigation assessment and post-mitigation assessment (see Annex C for additional details). For pre-mitigation assessment, scoring is used to identify the vulnerability risk level. These quantified pre-mitigation vulnerabilities are categorized into low risk, moderate risk, or high risk based on their assigned score. For post-mitigation assessment, scoring is used to determine the reduction of risk for the mitigated vulnerabilities.

8.4.2 eCVSS metric guidelines

The guidelines described in Table 2 assist in clarifying interpretation of the eCVSS metric definitions for setting the metric values.

Table 2—eCVSS metric guidelines

Category	Guideline
General	For pre-mitigation assessment, assume no security controls have been implemented.
	When in doubt, assess the worst-case scenario.
Impacts Safety Efficacy	Identifies threats/vulnerabilities that impact the safety of the patient.
	This metric does not have a direct effect on the assessment scoring, but is input to other metrics (e.g., Collateral Damage Potential).
Access Vector	Do not make any assumptions about the transport the device typically uses.
Access Complexity	Score higher if — It requires insider knowledge of the device design. — It depends on a narrow time window. — Vulnerable configuration is very rarely seen in practice. — It requires specific actions or information before a successful attack can be launched.
Authentication	For pre-mitigation analysis, should be set to **None**, unless the device type's use case requires authentication.
Confidentiality Impact	**Complete** means the attacker can access any data. For example, set to **Complete** when spoofing/impersonating device/user making read requests or remote code execution for read functionality.
	Partial means the attacker can access data, but it has no control of the type of data it can access. For example, set to **Partial** when tampering data in transit or spoofing/impersonating device/user receiving create/update/delete (CUD) requests.
	None means the attacker cannot access any data. For example, set to **None** when denial of service (e.g., blocking data flow) or spoofing/impersonating device/user sending data.
Integrity Impact	**Complete** means the attacker can modify any data. For example, set to **Complete** when spoofing/impersonating device/user sending data requests or remote code execution CUD functionality.
	Partial means the attacker can modify data, but it has no control of the type of data it can modify. For example, set to **Partial** when tampering data in transit or repudiation (e.g., modifying log data).
	None means the attacker cannot modify any data. For example, set to **None** when denial of service (e.g., blocking data flow) or spoofing/impersonating device/user receiving data.

Table 2—eCVSS metric guidelines *(continued)*

Category	Guideline
Availability Impact	**Complete** means the attacker can shut down or stop principal functionality of the targeted device. For example, set to **Complete** when spoofing/impersonating device/user controlling device or denial of service (e.g., crashing or stopping the device).
	Partial means the attacker can interrupt or reduce performance of the targeted device. For example, set to **Partial** when denial of service (e.g., blocking data flow) or tampering data in transit (i.e., denial of service).
	None means the attacker cannot affect the availability of the system. For example, set to **None** when spoofing/impersonating device/user receiving data or information disclosure.
Collateral Damage Potential	When in doubt, use the Suggested Collateral Damage Value approach (see 8.4.3).
	Set value to greater than **None** when — Impacts Safety Efficacy set to **Yes**. — Impacts to Confidentiality (e.g., potential legal damage). — Impacts to Integrity (e.g., potential business damage). — Impacts to Availability (e.g., potential patient damage).
Awareness	Set to **User** when — Access Vector is **Local**, and the device is attached to the user (e.g., reading or modifying data from device display). — Device is attached to user, and the device stops or crashes.
	Set to **Automatic** when the connected device can no longer communicate with the device under attack.
	Set to **Complete** when denial of service related to user interaction.

8.4.3 Suggested collateral damage

The effort to develop this standard detected that the eCVSS environmental metric Collateral Damage Potential may be subjective and the metric is difficult to repeat. To improve this situation, a Suggested Collateral Damage Value has been included in the assessment. This suggested value is determined by considering, for each vulnerability, the potential for business damage, legal damage, and patient damage. The Suggested Collateral Damage Value is only a suggestion, and the assessment may set any value for the Collateral Damage Potential metric. Table 3 provides a definition for each type of damage value.

Table 3—Suggestion collateral damage value definitions

Type of damage potential	Value	Value description
Business, legal, or patient	**None**	There is no damage related to business, legal, or the patient.
	Low	There is potential for minor level of damage related to business, legal, or the patient. For example, attack only affects a single instance of the product, some patients/physicians lose confidence in the product, minor patient harm exists.
	Medium	There is potential for major level of damage related to business, legal, or the patient. For example, regulatory forces recall, attack may affect one or more batches of the product but not all products, moderate patient harm (excluding life-threatening harm) exists, public loses confidence in the product.
	High	There is potential for catastrophic level of damage related to business, legal, or the patient. For example, regulatory forces stop of sale, attack may affect all instances of the product, legal action threatens the business viability, severe patient harm (including death) exists, public loses confidence in the brand/company.

8.4.4 System-wide metrics

One of the CVSS modifications that eCVSS introduced is the recognition that the “Requirement” metrics in the Environmental Metric Group (i.e., Confidential, Integrity, and Availability) are system wide. As such, these metrics are set only once for the system, and then each identified vulnerability is scored using these values. The values of these metrics are set based on the use cases and service provided by the PHDs/PoCDs.

8.4.5 Risk level thresholds

To determine the level of risk for each scored vulnerability, thresholds can be defined. The definition of the thresholds should consider the device types, use cases, and primary function and be assigned by domain experts to help ensure the thresholds are appropriate. These thresholds distinguish low-, moderate-, and high-risk vulnerabilities.

8.5 Mitigation

Mitigation is step 4 in the iterative vulnerability assessment. Mitigation of a cybersecurity vulnerability is achieved by introducing an information security control into the system designed specifically to strengthen the system against an attack using that vulnerability. As described in 8.4, each mitigated vulnerability is re-assessed and re-scored to determine the new risk level of the vulnerability. Often the vulnerabilities are not removed but reduced to an acceptable level of risk.

The capability part of cybersecurity is described in IEEE Std 11073-40102 [B3]. As such, this standard does not go into detail about which selected mitigations address common PHD/PoCD cybersecurity vulnerabilities or recommend scenarios when such mitigations should be applied.

8.6 Iteration

Iteration of the vulnerability assessment is step 6. Based on the post-mitigation scores, all vulnerabilities that still have unacceptable risk should receive additional information security controls to reduce the risk to an acceptable level. Note that additional mitigations applied to a specific PHD/PoCD interface may also reduce other identified vulnerabilities on that interface and their post-mitigation score should reflect the improved mitigation.

Once all of the identified vulnerabilities are reduced to an acceptable level of risk, the iterative vulnerability assessment is complete.

Annex A

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

[B1] FDA, "Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software."[6]

[B2] FIRST, "A Complete Guide to the Common Vulnerability Scoring System," June 2007.[7]

[B3] IEEE Std 11073-40102™, Health informatics—Device interoperability—Part 40102: Foundational—Cybersecurity—Capabilities for Mitigation.[8]

[B4] IEEE white paper, "PHD Cybersecurity Standards Roadmap," Apr. 2019. (https://standards.ieee.org/industry-connections/personal-health-device-cybersecurity-whitepaper.html)

[B5] ISO/IEEE 11073-10201, Health informatics—Device interoperability—Part 10201: Domain information model.[9]

[B6] Lockheed Martin, "A Threat-Driven Approach to Cyber Security," 2019.[10]

[B7] PCHAlliance, "Continua Design Guidelines," Dec. 2017.[11]

[6] This FDA document is available from the U.S. Food and Drug Administration (https://www.fda.gov/medical-devices/digital-health-center-excellence/cybersecurity).
[7] FIRST documents are available from the Forum of Incident Response and Security Teams (https://www.first.org).
[8] IEEE publications are available from The Institute of Electrical and Electronics Engineers, Inc. (https://www.ieee.org/).
[9] ISO/IEEE publications are available from the International Organization for Standardization (https://www.iso.org/), The Institute of Electrical and Electronic Engineers, Inc. (https://www.ieee.org/), and the American National Standards Institute (http://www.ansi.org/).
[10] Lockheed Martin white papers are available from the Lockheed Martin Corporation (https://www.lockheedmartin.com).
[11] PCHAlliance documents are available from the Personal Connected Health Alliance (https://www.pchalliance.org).

Annex B

(informative)

STRIDE

STRIDE is a threat classification model developed by Microsoft® for assessing computer information security threats. Microsoft also provides modeling software known as the Threat Modeling Tool (TMT), which can generate a list of vulnerabilities from a threat model. The vulnerability list can be exported from Microsoft TMT and imported to Microsoft Excel® (see Annex D) for further scoring.[12] STRIDE is a mnemonic for information security threats in the six categories listed in Table B.1.

Table B.1—STRIDE threat categories and security properties

Threat category	Description	Security property
Spoofing	Spoofing is a key risk for systems that have many users but provide a single execution context at the application and database level. An example of identity spoofing is illegally accessing and then using another user's authentication information, e.g., username and password.	Authentication
Tampering	Data tampering involves the malicious modification of data. Examples include unauthorized changes made to persistent data, e.g., that held in a database, and the alteration of data as it flows between two computers over an open network, e.g., the Internet.	Integrity
Repudiation	Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise. For example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations.	Non-repudiation
Information Disclosure	Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it. Examples include the ability of users to read a file to which they were not granted access or the ability of an intruder to read data in transit between two computers.	Confidentiality
Denial of Service	Denial of service attacks deny service to valid users. An example is making a web server temporarily unavailable or unusable. Protection against certain types of denial of service threats improves system availability and reliability.	Availability
Elevation of Privileges	Elevation of privilege threats involve an unprivileged user that gains privileged access to a system and thereby has sufficient access to compromise or destroy the entire system. Examples include situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself. Such situations are considered dangerous.	Authorization

The main use of STRIDE is to decompose a system threat model into a threat list by analyzing each component for susceptibility to the threats. The resulting threat list can be used as input into vulnerability assessment (see Clause 8), where the vulnerability of each threat is scored to determine its risk and then vulnerabilities of unacceptable risk are appropriately mitigated by adding information security controls to the system. The resulting system with additional information security controls becomes input for the vulnerability assessment, and the scoring and mitigation process is repeated. This iterative approach is continued until the remaining vulnerabilities are determined to be of an acceptable level of risk. The rationale for this approach is that mitigation of all high-risk vulnerabilities identified by each threat for each component of the system produces a secure system.

[12] The information in this annex is given for the convenience of users of this standard and does not constitute an endorsement by IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

Table B.2 lists each element of the DFD and shows the STRIDE threat category to which it is susceptible.

Table B.2—DFD elements and STRIDE threat category

Element	Spoofing	Tampering	Repudiation	Information Disclosure	Denial of Service	Elevation of Privilege
Data flows		X		X	X	
Data stores		X		X	X	
Processes	X	X	X	X	X	X
Interactions	X		X			

Table B.3 lists the possible threat types for each STRIDE threat category.

Table B.3—Threat types by STRIDE threat category

STRIDE threat category	Threat type	Description
Spoofing	Spoofing of Source External Actor	An external actor may be spoofed by an attacker and this action may lead to unauthorized access to a process or data store. Consider using a standard authentication mechanism to identify the external actor.
	Spoofing of Destination External Actor	An external actor may be spoofed by an attacker, and this action may allow data to be sent to the attacker's target instead of to the external actor. Consider using a standard authentication mechanism to identify the external actor.
	Spoofing of Source Process	A process may be spoofed by an attacker, and this action may lead to unauthorized access to another process or data store. Consider using a standard authentication mechanism to identify the source process.
	Spoofing of Destination Process	A process may be spoofed by an attacker, and this action may lead to information disclosure by an external actor, another process, or data store. Consider using a standard authentication mechanism to identify the destination process.
	Spoofing of Source Data Store	A data store may be spoofed by an attacker, and this action may allow incorrect data to be delivered to the requesting process or external actor. Consider using a standard authentication mechanism to identify the source data store.
	Spoofing of Destination Data Store	A data store may be spoofed by an attacker, and this action may allow data to be written to the attacker's target instead of to the data store. Consider using a standard authentication mechanism to identify the destination data store.
Tampering	Potential Lack of Input Validation for a Process	Data flowing across interface may be tampered with by an attacker. This action may lead to a denial of service attack against a process or data store, an elevation of privilege attack against a process or data store, or an information disclosure by a process or data store. Failure to verify that input is as expected is a root cause of many exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.
Repudiation	Potential Data Repudiation by Process	A process claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
	External Actor Potentially Denies Receiving Data	An external actor claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Table B.3—Threat types by STRIDE threat category *(continued)*

STRIDE threat category	Threat type	Description
Information Disclosure	Data Flow Sniffing	Data flowing across interface may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.
	Weak Access Control for a Resource	Improper data protection of a data store can allow an attacker to read information not intended for disclosure. Review authorization settings.
Denial of Service	Data Flow Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.
	Potential Process Crash or Stop	A process crashes, halts, stops, or runs slowly; all cases would violate an availability metric.
	Potential Excessive Resource Consumption for a Process or a Data Store	Does a process or a data store take explicit steps to control resource consumption? Resource consumption attacks can be hard to address, and at times it makes sense to let the operating system do the job. Be careful that resource requests do not deadlock and that they do timeout.
Elevation of Privilege	Cross Site Request Forgery	Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The user browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g., by session cookie, integrated authentication, Internet Protocol whitelisting. The attack can be carried out in many ways, e.g., by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can be resolved on the server side only by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) that is known only to the legitimate web site and the browser and is protected in transit through Secure Sockets Layer/ Transport Layer Security (SSL/TLS).
	Elevation by Changing the Execution Flow in a Process	An attacker may pass data into a process to change the flow of program execution within the process to the attacker's choosing.
	Elevation Using Impersonation	A process or external actor may be able to impersonate the context of another process or external actor to gain additional privilege.
	A Process May be Subject to Elevation of Privilege Using Remote Code Execution	A process may be able to remotely execute code for another process.

The Open Web Application Security Project® (OWASP®) has assigned appropriate primary mitigation techniques to address the vulnerabilities threatened by each STRIDE category. Table B.4 shows this list of suggested primary mitigation techniques.

Table B.4—STRIDE threat category and primary mitigation techniques

Threat category	Primary mitigation techniques
Spoofing	Authentication Protect secrets and secret data Do not store secrets
Tampering	Authorization MACs Digital signatures Input validation and sanitization Physical tamper resistant Physical tamper evidence
Repudiation	Digital signatures Audit trails
Information Disclosure	Authorization Privacy-enhanced protocols Encryption Protect secrets and secret data Do not store secrets
Denial of Service	Authentication Authorization Filtering Throttling Quality of service
Elevation of Privilege	Authorization Run with least privilege End-user signalization

Annex C

(informative)

embedded Common Vulnerability Scoring System

C.1 Overview

The original version of CVSS was designed to address software-only systems and create scores after the software systems were in the field. In the effort to develop this standard, changes to CVSS were needed to support physical PHDs/PoCDs and create scores during design to guide the development of systems.

As a result, the embedded Common Vulnerability Scoring System (eCVSS) was created as a slightly modified branch of CVSS 2.0 (FIRST [B2]). This creation is not an effort of the Forum of Incident Response and Security Teams (FIRST), but is instead proposed by the working group that developed this standard.

Changes to CVSS are as follows:

— The Temporal Group was effectively removed by forcing the three metrics to a neutral value because the scoring in eCVSS is conducted during design.
— The three "Requirement" metrics in the Environmental Metric Group (i.e., Confidential, Integrity, Availability) were recognized to be system wide; hence the System-wide Metric Group was created. These metrics are set only once for the system and then applied to all the identified vulnerabilities.
— The Target Distribution metric was removed as it refers to distribution of the system, and the scoring in eCVSS is conducted during design. Instead, a new Awareness metric in the Environmenal Metric Group has replaced the Target Distribution metric.

Based on these changes, eCVSS is composed of three metric groups: System-wide, Base, and Environmental. Each metric group is composed of a set of metrics as shown in Figure C.1.

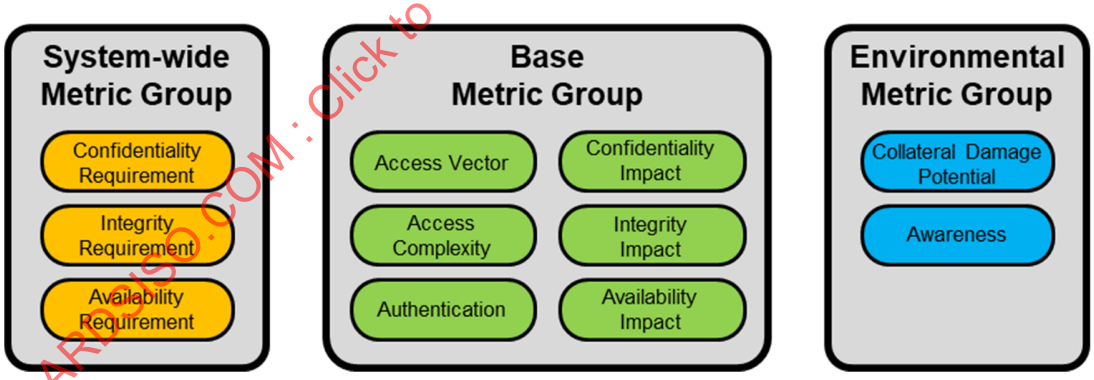

Figure C.1— eCVSS metric groups

Using the 11 separate metrics from the three metric groups of eCVSS, a vulnerability can be scored from 0 to 10 with granularity of tenths of units, where a score of 0 means *no issue* and 10 means *high-level concern*. The eCVSS metric groups are described in Table C.1.

Table C.1—eCVSS metric group description

Metric group	Description
System-wide	Represents the system requirements for a confidentiality, integrity, and availability (CIA) triad (see IEEE Std 11073-40102 [B3]) that are set once for the product and then applied to all vulnerabilities.
Base	Represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.
Environmental	Represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment.

C.1.1 System-wide metrics

The System-wide Metric Group captures the product requirements for a confidentiality, integrity, and availability (CIA) triad (see IEEE Std 11073-40102 [B3]). These metrics are set once for the product and then applied to all identified vulnerabilities.

Table C.2 provides a definition for each System-wide metric as well as the possible values and numeric representation.

Table C.2—Descriptions of System-wide metrics and values

System-wide metric	Metric description	Metric value	Value description	Numeric
Confidentiality Requirement (CR)	Enables the analyst to customize the score depending on the importance of the affected target device to the organization, measured in terms of confidentiality, integrity, and availability.	Undefined	N/A	1.000
		Low (L)	Loss of [confidentiality | integrity | availability] is likely to have only a limited adverse effect on the organization or users of the device.	0.500
Integrity Requirement (IR)		**Medium** (A)	Loss of [confidentiality | integrity | availability] is likely to have a serious adverse effect on the organization or users of the device.	1.000
Availability Requirement (AR)		**High** (H)	Loss of [confidentiality | integrity | availability] is likely to have a catastrophic adverse effect on the organization or users of the device.	1.510

C.1.2 Base metrics

The Base Metric Group captures the characteristics of a vulnerability that are constant with time and across user environments. The Access Vector, Access Complexity, and Authentication metrics capture how the vulnerability is accessed and whether extra conditions are required to exploit it. The Confidentiality Impact, Integrity Impact, and Availability Impact metrics measure how a vulnerability, if exploited, will directly affect an asset.

Table C.3 provides a definition for each base metric as well as the possible values and numeric representation.

Table C.3—Descriptions of base metrics and values

Base metric	Metric description	Metric value	Value description	Numeric
Access Vector (AV)	How the vulnerability is exploited. The more remote the attacker can be to attack a system, the greater the score.	Undefined	N/A	0.000
		Local (L)	Attacker requires physical access to the device.	0.395
		Adjacent (A)	Attacker requires access to a broadcast or very short-range communications.	0.646
		Network (N)	Attacker requires access to WAN or Internet.	1.000
Access Complexity (AC)	The complexity of the attack required to exploit the vulnerability once an attacker has gained access to the system. The lower the required complexity, the higher the vulnerability score.	Undefined	N/A	0.000
		Low (L)	Specialized access conditions or extenuating circumstances do not exist.	0.710
		Medium (M)	The access conditions are somewhat specialized.	0.610
		High (H)	Specialized access conditions exist.	0.350
Authentication (Au)	The strength of the authentication process used to exploit the vulnerability.	Undefined	N/A	0.000
		None (N)	Authentication is not required to access and exploit the vulnerability.	0.704
		Single (S)	Authentication is easily defeated or uses a weak method for vetting. Examples include — Storing or transmitting of credentials in plain text — Fixed (i.e., hard coded) credentials — Automatic trust based on device type	0.560
		Multiple (N)	Authentication employs industry's best practice for vetting the authenticity of the user or device. Examples include: — Storing of hashed credentials only — Multiple levels of authentication — Enforced unique credentials	0.450

Table C.3—Descriptions of base metrics and values *(continued)*

Base metric	Metric description	Metric value	Value description	Numeric
Confidentiality Impact (C)	The impact to confidentiality of a successfully exploited vulnerability.	Undefined	N/A	0.000
		None (N)	There is no impact to the confidentiality of the system.	0.000
		Partial (P)	There is considerable information disclosure. Access to some system files is possible; however, the attacker does not have control over what is obtained, or the scope of the loss is constrained.	0.275
		Complete (C)	There is total information disclosure, allowing all system files to be revealed.	0.660
Integrity Impact (I)	The impact to integrity of a successfully exploited vulnerability.	Undefined	N/A	0.000
		None (N)	There is no impact to the integrity of the system.	0.000
		Partial (P)	Modification of some system files or information is possible; however, the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.	0.275
		Complete (C)	There is a total compromise of system integrity.	0.660
Availability Impact (A)	The impact to availability of a successfully exploited vulnerability.	Undefined	N/A	0.000
		None (N)	There is no impact to the availability of the system.	0.000
		Partial (P)	There is reduced performance or interruptions in resource availability.	0.275
		Complete (C)	There is a total shutdown of the target system, rendering the system's principal functionality non-operational.	0.660

C.1.3 Environmental metrics

The Environmental Metric Group captures the characteristics of a vulnerability that are associated with a user's environment. Since environmental metrics are optional, they each include a metric value that has no effect on the score over time. This value is used when the user feels the particular metric does not apply and wishes to "skip over" it.

Table C.4 provides a definition for each environmental metric as well as the possible values and numeric representation.

Table C.4—Descriptions of environmental metrics and values

Environmental metric	Metric description	Metric value	Value description	Numeric
Collateral Damage Potential (AV)	The potential for loss of life or physical assets through damage or theft of property or equipment. This metric may also measure economic loss of productivity or revenue. The greater the damage potential, the higher the vulnerability score.	Undefined	N/A	0.000
		None (N)	There is no potential for loss of life, physical assets, productivity, or revenue.	0.000
		Low (L)	A successful exploit of this vulnerability may result in slight physical damage, property damage, loss of revenue, or productivity.	0.100
		Low-Medium (LM)	A successful exploit of this vulnerability may result in moderate physical damage, property damage, loss of revenue, or productivity.	0.300
		Medium-High (MH)	A successful exploit of this vulnerability may result in significant physical damage, property damage, loss of revenue, or productivity.	0.400
		High (H)	A successful exploit of this vulnerability may result on catastrophic physical damage, property damage, loss of revenue, or productivity.	0.500
Awareness (Aw)	The ability of a vulnerability exploit to be detected by the system or its user. It is meant as an environment-specific indicator to lower the scoring as a result of an exploit being detected.	Undefined	N/A	1.000
		None (N)	Exploit cannot be detected by the user or the device.	0.000
		User (U)	Exploit is detectable by the user, e.g., the device case has obvious alterations, tampering is evident.	0.510
		Automatic (A)	Exploit is detectable by the device (either software or hardware).	0.680
		Complete (C)	Exploit is detectable by the user and the device.	0.840

C.1.3.1 Suggested collateral damage value

The effort to develop this standard detected that the eCVSS environmental metric Collateral Damage Potential may be subjective and the metric is difficult to repeat. To improve this situation, a Suggested Collateral Damage Value has been included in the assessment. This suggested value is determined by considering, for each vulnerability, the potential for business damage, legal damage, and patient damage. The Suggested Collateral Damage Value is only a suggestion, and the assessment may set any value for the Collateral Damage Potential metric. Table C.5 provides a definition for each type of damage value. See also the equations defined in C.2.

Table C.5—Suggested collateral damage value definitions

Type of damage potential	Value	Value description
Business, Legal, or Patient	**None**	There is no damage related to business, legal, or patient.
	Low	There is potential for a minor level of damage related to business, legal, or patient. For example, the attack affects only a single instance of the product, some patients/physicians lose confidence in the product, or the patient experiences minor harm.
	Medium	There is potential for a major level of damage related to business, legal, or patient. For example, a regulatory body forces recall, the attack may affect one or more batches of the product but not all products, the patient experiences moderate harm (excluding life-threatening harm), or the public loses confidence in the product.
	High	There is potential for a catastrophic level of damage related to business, legal, or patient. For example, a regulatory body forces stop of sale, the attack may affect all instances of the product, legal action ensues that threatens the business viability, the patient experiences severe harm (including death), or the public loses confidence in the brand/company.

C.1.4 Impact safety efficacy

Since the eCVSS approach is conducted against PHDs/PoCDs, impacts to patient safety must also be considered. Any vulnerability that impacts safety is marked and should be addressed as part of patient risk management. While safety and security may be coupled and could be commonly mitigated, this standard is not intended to address impacts to safety. However, note that the value of the eCVSS environmental metric Collateral Damage Potential considers patient harm.

C.2 Scoring equations in pseudo code

C.2.1 eCVSS

```
If Base Section completed Then
  Impact = 10.41 * (1 - (1 - ConfidentialityImpact) * (1 - IntegrityImpact) *
           (1 - AvailabilityImpact))
    If Impact == 0 then
      fImpact = 0
    else
      fImpact = 1.176

  ExploitabilityScore = 20 * AccessComplexity * Authentication * AccessVector
  BaseScore = Round_To_Tenths (((0.6 * Impact) + (0.4 * ExploitabilityScore) -
1.5) *
              fImpact)

If Environment Section completed Then
  AdjustedImpact = Round_To_Tenths(Minimum (10, 10.41 * (1 - (1 -
ConfidentialityImpact *
                  ConfidentialityRequirement) * (1 - IntegrityImpact *
                  IntegrityRequirement) * (1 - AvailabilityImpact *
                  AvailabilityRequirement))))

  AdjustedBaseScore = Round_To_Tenths(((0.6 * AdjustedImpact) + (0.4 *
                      ExploitabilityScore) - 1.5) * fImpact)
  AdjustedTemporalScore = Round_To_Tenths(AdjustedBaseScore)
  EnvironmentalScore = Round_To_Tenths((AdjustedTemporalScore + (10 -
                       AdjustedTemporalScore) * CollateralDamagePotential) *
Awareness)
```

```
If EnvironmentalScore == 0 Then
  OverallScore = BaseScore
Else
  OverallScore = EnvironmentalScore
```

C.2.2 Suggested Collateral Damage

The Suggested Collateral Damage value using the following equations:

```
BDP = Business Damage Potential
LDP = Legal Damage Potential
PDP = Patient Damage Potential
SCDV = Suggested Collateral Damage Value

If (no values entered in BDP, LDP, and PDP) {
  SCDV = ""
} Else If (all damage potential values == none) {
  SCDV = "None"
} Else If (any damage potential value == high) {
  SCDV = "High"
} Else If (2 of the damage potential values == medium) {
  SCDV = "Medium-High"
} Else If (only 1 damage potential value == medium AND any other damage
potential value == low) {
   SCDV = "Low-Medium"
} Else If (any damage potential value == low) {
  SCDV = "Low"
} Else {
  SCDV = "None"
}
```

C.3 Test vectors

The values of the eCVSS metrics are typically described in a compressed format called a *vector*. A vector comprises a metric–value pair separated by a full colon. Both the metric and its value are represented by a letter code (see Table C.3 and Table C.4). Following is an example of a vector:

ISE:Y AV:L AC:M Au:S C:C I:P A:N CDP:MH Aw:U

This vector is interpreted as follows:

- Impact Safety Efficacy (ISE) = **Yes** (Y)
- Access Vector (AV) = **Local** (L)
- Access Complexity (AC) = **Medium** (M)
- Authentication (Au) = **Single** (S)
- Confidentiality Impact (C) = **Complete** (C)
- Integrity Impact (I) = **Partial** (P)
- Availability Impact (A) = **None** (N)
- Collateral Damage Potential (CDP) = **Medium-High** (MH)
- Awareness (Aw) = **User** (U)

See E.4 for both pre- and post-mitigation vectors along with the resulting scores that could be used for validation.

Annex D

(informative)

Microsoft TMT2Excel Macro

The Visual Basic for Applications (VBA) macro in this annex can be used to import the vulnerability list exported from the Microsoft TMT into Microsoft Excel for further scoring.[13]

```
Public Sub MicrosoftTMT2ExcelMacro()
Dim xmlFilePath As Variant
Dim prefixID As Variant
Dim iRet As Integer
Dim oList As ListObject
Dim i As Integer
Dim RowCount As Integer
Dim sdltmtWorksheet As Worksheet
Dim vulAssPreWorksheet As Worksheet
Dim key As Variant
Dim value As Variant
iRet = MsgBox("Would you like to import the data from the Microsoft Threat
Modeling
Tool?" & Chr(10) & Chr(10) & "Note that this will overwrite the 'Vulnerability
Assessment (pre)' sheet, 'Potential Vulnerability' section.", vbOKCancel,
"Notify User")

If iRet = vbCancel Then Exit Sub

'Get the Device Type ID Prefix from the user
prefixID = InputBox("Enter device type ID prefix (e.g. Insulin Pump = 'IP')")
'Let the user select the correct TM4/TM7 file to load
xmlFilePath = Application.GetOpenFilename _
(FileFilter:="TMT 2016,*.TM7,TMT 2014,*.TM4", _
Title:="Open SDL Threat Modeling Tool File", MultiSelect:=False)
If TypeName(xmlFilePath) = "Boolean" Then Exit Sub

'Add the SDLTMT sheet
Application.DisplayAlerts = False
If sheetExists("SDLTMT") Then
Worksheets("SDLTMT").Visible = xlSheetVisible
Worksheets("SDLTMT").Delete
End If
Set sdltmtWorksheet = Worksheets.Add(After:=Worksheets(Worksheets.Count))
sdltmtWorksheet.Name = "SDLTMT"
Set sdltmtWorksheet = Worksheets("SDLTMT")
sdltmtWorksheet.Activate

'Load the XML file into the table
ActiveWorkbook.XmlImport    URL:=xmlFilePath,    ImportMap:=Nothing,
Overwrite:=True, Destination:=Range("$A$1")

Worksheets("SDLTMT").ListObjects(Worksheets("SDLTMT").ListObjects.Count
).Name = "XMLtable"
Application.DisplayAlerts = True
```

[13] The information in this annex is given for the convenience of users of this standard and does not constitute an endorsement by IEEE of these products. Equivalent products may be used if they can be shown to lead to the same results.

```
'Clear up the xml table by deleting all the graphic related entries
Application.ScreenUpdating = False

Set oList = Worksheets("SDLTMT").ListObjects("XMLtable")
RowCount = oList.DataBodyRange.Rows.Count

For i = RowCount To 1 Step -1
Debug.Print oList.DataBodyRange.Cells(i, 2)
If oList.DataBodyRange.Cells(i, 2) = "DRAWINGSURFACE" Then
oList.ListRows(i).Delete
End If
Next

'Kill the first 68 columns that are blank
For i = 1 To 68
oList.ListColumns(1).Delete
Next

'populate the Vulnerability Assessment (Pre) sheet
Set vulAssPreWorksheet = Worksheets("Vulnerability Assessment (pre)")

RowCount = oList.DataBodyRange.Rows.Count
colID = 5
colKey = 10
colValue = 11
currentID = oList.DataBodyRange.Cells(1, colID)
currentVulAssWorksheetRow = 9
currentPrefixIDCount = 1
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 5) = currentID
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 1) = prefixID &
currentPrefixIDCount
For i = 1 To RowCount Step 1
Debug.Print oList.DataBodyRange.Cells(i, colID)
tempID = oList.DataBodyRange.Cells(i, colID)

If IsEmpty(tempID) Then
'end of threat list
Exit For
ElseIf tempID <> currentID Then
'move to the next vulnerability assessment (pre) row
currentID = tempID
currentVulAssWorksheetRow = currentVulAssWorksheetRow + 1
currentPrefixIDCount = currentPrefixIDCount + 1
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 5) = currentID
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 1) = prefixID &
currentPrefixIDCount
End If

Debug.Print oList.DataBodyRange.Cells(i, colKey)
key = oList.DataBodyRange.Cells(i, colKey)
Debug.Print oList.DataBodyRange.Cells(i, colValue)
value = oList.DataBodyRange.Cells(i, colValue)

If key = "Title" Then
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 2) = value
ElseIf key = "UserThreatCategory" Then
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 3) = value
ElseIf key = "UserThreatShortDescription" Then
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 4) = value
ElseIf key = "UserThreatDescription" Then
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow, 4) =
vulAssPreWorksheet.Cells(currentVulAssWorksheetRow,  4)  &  vbNewLine  &
vbNewLine & value
```

```
End If
Next

vulAssPreWorksheet.Activate
sdltmtWorksheet.Visible = xlSheetVeryHidden
Application.ScreenUpdating = True

End Sub
```