# Attack Modeling and Mitigation Strategies for Risk-Based Analysis of Networked Medical Devices

Bronwyn J. Hodges
Dept of Info. Systems Technology
School of Computing
University of South Alabama
bjh1521@jagmail.southalabama.edu

J. Todd McDonald
Dept of Computer Science
School of Computing
University of South Alabama
jtmcdonald@southalabama.edu

William Bradley Glisson
Cyber Forensics Intelligence Center
Dept of Computer Science
Sam Houston State University
glisson@shsu.edu

Michael Jacobs
USA Simulation Program
Division of Academic Affairs
University of South Alabama
mjacobs@southalabama.edu

Maureen Van Devender
Dept of Info. Systems Technology
School of Computing
University of South Alabama
mvandevender@southalabama.edu

J. Harold Pardue
Dept of Info. Systems Technology
School of Computing
University of South Alabama
hpardue@southalabama.edu

## Abstract

*The escalating integration of network-enabled medical devices raises concerns for both practitioners and academics in terms of introducing new vulnerabilities and attack vectors. This prompts the idea that combining medical device data, security vulnerability enumerations, and attack-modeling data into a single database could enable security analysts to proactively identify potential security weaknesses in medical devices and formulate appropriate mitigation and remediation plans. This study introduces a novel extension to a relational database risk assessment framework by using the open-source tool OVAL to capture device states and compare them to security advisories that warn of threats and vulnerabilities, and where threats and vulnerabilities exist provide mitigation recommendations. The contribution of this research is a proof of concept evaluation that demonstrates the integration of OVAL and CAPEC attack patterns for analysis using a database-driven risk assessment framework.*

## 1. Introduction

The escalating integration of technology into the healthcare sector enables new and innovative ways for medical personnel to interact with patients, capture additional data, and enhance healthcare delivery. It is being argued that technological advancements such as robotic surgeries, implantable cardiac devices, physiological monitors, and Internet of Things (IoT) devices generate additional data that are valuable to medical professionals, ultimately leading to big data analysis opportunities while concurrently inspiring evolution to an overall pervasive healthcare environment [1-3]. In addition to the benefits, technology exposes healthcare to the risks that are inherent to digital settings, which fosters an environment that is conducive to adversarial cyberattacks [4-6]. These concerns are being raised by practitioners [7], the government [8], and academics [6, 9-15] for medical equipment and any devices that communicate with this equipment.

As with any network-enabled device, medical devices are susceptible to cyber threats, including ransomware, data breaches, distributed denial of service attacks (DDoS), insider threats, and many more. Certain threats may allow attackers to gain access to patient data and/or alter the functionality of a device while connected to a patient. A 2019 survey reveals that healthcare executives see medical device security as one of the top five risks they face and that they lack effective strategies to address their concerns [16]. Support for this stance is visible in a recent article describing how a series of widely used infusion pump-linked workstations contain security flaws [17]. This environment provides the impetus for this research and exploration into the use of an open-source vulnerability scanner known as Open Vulnerability Assessment Language (OVAL) [18] to scan devices connected to medical devices and identify community standard vulnerabilities. These vulnerabilities are then linked to attack patterns in MITRE Corporation's Common Attack Pattern Enumeration and Classification (CAPEC) framework [19] to identify how an attacker could leverage these vulnerabilities as well as identify how to mitigate

URI: https://hdl.handle.net/10125/64538
978-0-9981331-3-3

identified threats. The study presented in this paper is part of an MSc Thesis [20] that integrates OVAL and CAPEC into a Threat-Vulnerability-Asset risk framework known as MedDevRisk [11, 12]. The contribution of this research is a proof of concept evaluation that demonstrates the integration of OVAL and CAPEC for analysis using a database-driven risk assessment framework.

The remainder of the paper is organized as follows: Section 2 provides background on key cybersecurity concepts and standards; Section 3 provides an overview of the case-study methodology performed to evaluate integration of new concepts such as attack models, OVAL assessments, and risk using real-world data; Section 4 details results and analysis from the study. Finally, section 5 summarizes the work and provides recommendations for future work.

## 2. Background

Previous academic research focusing on the risk assessment of networked medical devices led to the production of a database-driven risk assessment framework called MedDevRisk [11], illustrated in Figure 1. MedDevRisk utilizes constructs such as STRIDE [21] and Threat Vulnerability Asset (TVA) [22, 23] in schema relationships that connect Common Vulnerability Scoring System (CVSS) [24] risk assessment criteria and Common Vulnerability and Exposures (CVE) [25] incident reports. MedDevRisk schema is implemented in a relational database management system (RDBMS). Through the use of Structured Query Language (SQL), MedDevRisk can produce a risk assessment of medical devices stored in the database targeted at both low-and-high-level managers. This research extends the original data sources and data relationships by integrating OVAL reporting [18] [12] and CAPEC attack patterns [19].

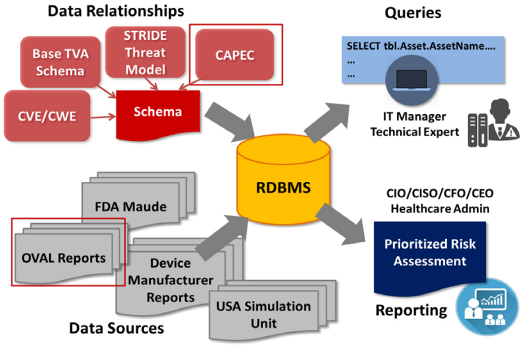


**Figure 1. MedDevRisk Extensions [11]**

STRIDE [26] is a threat model that classifies threats into six attack vectors, which include Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. TVA [22, 23, 27] is another threat framework utilized by MedDevRisk to establish a relationship between the assets stored in the database, identified threats, and vulnerabilities. MedDevRisk uses CVEs [25] to link public vulnerability classifications to assets and uses CVSS [24] to assign a numerical risk value to identified vulnerabilities.

### 2.1 OVAL

OVAL [12] is an open international community standard for the assessment and reporting of the machine state of a computer. It includes an XML language for expressing machine state and reference implementations and repositories of information written in the OVAL XML language. OVAL provides for the representation of system configuration states for testing and evaluation. Figure 2 provides an overview of the OVAL process [18], illustrating how OVAL definitions are paired against specific system characteristics to generate analysis and results.

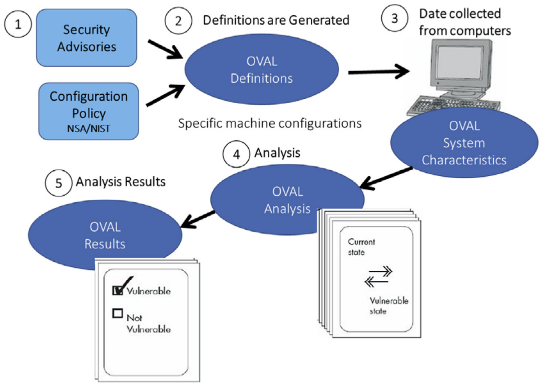

1
Security Advisories
2 Definitions are Generated
3 Date collected from computers
OVAL Definitions
Configuration Policy NSA/NIST
Specific machine configurations
OVAL System Characteristics
4 Analysis
5 Analysis Results
OVAL Analysis
OVAL Results
Vulnerable
Not Vulnerable
Current state
Vulnerable state

**Figure 2. OVAL Process [18]**

The OVAL language has three core schema types: definitions, system characteristics, and results. Definition schemas define systems in two areas: security advisories that warn of threats and vulnerabilities, and government agency best practice policies for system security. The definition schemas are structured to specify the configuration information that is to be collected from an individual system in order to compare it to a definition. The OVAL process is a comparison of an OVAL definition to the system characteristics which yields an OVAL result that follows the results schema format and identifies whether the system is vulnerable or not vulnerable. OVAL identifies system vulnerabilities through the application of Common Vulnerability Enumerations (CVEs) [25] and

Common Platform Enumerations (CPE) [28], which is now a NIST standard.

## 2.2 CAPEC

The Common Attack Pattern Enumeration and Classification (CAPEC) [19] attack framework was created by MITRE Corporation to identify an adversarial viewpoint of a system weakness. An attacker's viewpoint is perceived by identifying the skill level needed to pull off an attack, attack patterns used to gain access to weaknesses, and the attack steps that are taken to exploit weaknesses. CAPEC also identifies mitigation strategies for these attack patterns. CAPEC not only provides an attacker's viewpoint of a weakness but also includes security tactics to combat an attack.

CAPEC delineates attack patterns into two high-level abstractions, which are domains of attack and mechanisms of attack [19]. For example, CAPEC documents the classic buffer overflow exploit as pattern 100 (Overflow Buffers), which represents both a software-based attack (its domain) as well as an example of manipulating data structures (its mechanism). As with a typical pattern, CAPEC pattern 100 provides: 1) a description, 2) likelihood of attack (high), 3) typical severity (very high), 4) relationships to other CAPEC items, 5) an execution flow detailing exploration, experimentation, and exploitation, 6) prerequisites for the attack (for example: 'targeted software inadequately performs bounds-checking'), 7) skills required, 8) resources required, 9) indicators, 10) consequence and 11) mitigations [19]. Each CAPEC pattern is further tied to one or more related CWE weaknesses [29], which provides an even greater amount of documented examples and mitigations.

# 3. Methodology

This research explores the use of OVAL to identify device-specific and community classified vulnerabilities in networked medical devices. CAPEC is used to provide an adversarial viewpoint of identified vulnerabilities along with community provided mitigation techniques to combat an attack. Expanding the functionality of MedDevRisk to incorporate OVAL and CAPEC, this research conducts a case study as defined by Oates [30] on devices used in an academic medical setting. The following tasks were completed to achieve this integration:

1. **Device Data Collection:** OVAL was executed on four devices provided by the Human Simulation Unit at the University of South Alabama
2. **Framework Data Collection:** XML files containing CVE and CAPEC data were used to import framework data into MedDevRisk
3. **Schema Expansion:** Modified a peer-reviewed MedDevRisk schema to incorporate OVAL and CAPEC as well as eliminate normalization issues
4. **Data Entry:** Python scripts were used to import gathered data to reduce errors introduced by human interaction with MedDevRisk
5. **Query Configuration:** Queries were created to highlight new framework functionalities
6. **Result Reports:** Queries were executed and results analyzed

## 3.1 Device Data Collection

Before modifying the peer-reviewed database schema for MedDevRisk [12], a solid understanding of the data produced by OVAL is needed. This knowledge was gained through the help of a technical specialist employed in the Simulation Unit at the University of South Alabama. This specialist executed OVAL on four Apple Mac OS devices connected to medical equipment. This collection process corresponds to steps 1-5 of the OVAL process described in Figure 2. A tool called OVAL Interpreter (version 5.10.1.7) was installed on each case-study computer in the Simulation Unit and used to conduct a comparison between the OVAL Definitions and System Characteristics files. The operating systems on the case study machines included Apple macOS X Mavericks versions 10.9.2 and 10.9.5 and macOS Sierra version 10.12. Four OVAL results files were produced from this comparison. The following data were provided in each Result file: 'OVAL ID', 'Result', 'Class', 'Reference ID' and 'Title'.

The 'OVAL ID' specifies the specific OVAL definition that was tested. The 'Result' data specifies if the system being evaluated was compliant or incompliant with a certain 'OVAL ID'. 'Class' data categorizes the specific definition into the categories of vulnerability, inventory, miscellaneous, patch, or compliance. This research only utilizes definitions that are in the vulnerability category. 'Reference ID' identifies a specific CVE or CPE ID connected to a specific 'OVAL ID'. 'Title' provides a description of the 'OVAL ID'.

## 3.2 Framework Data Collection

MITRE, a Federally Funded Research Center, provides XML files containing data relevant to CVE and CAPEC definitions. These files were used in a Python script that extrapolated the XML file data collected from case-study machines and imported the data into corresponding tables in MedDevRisk. The data gathered includes: CAPEC ID, CAPEC Name, Description, Attack Steps, Attack Techniques, Mitigation Strategies, CWE ID, CVE ID, CVE Description, Impact Score, Attack Vector, Attack Complexity, Confidentiality, Availability, Integrity and CWE ID.

The OVAL results gathered from the case study machines were linked to CVE records and CAPEC records. Each OVAL record contains a reference to a CVE, which helps us gain more information on the vulnerability from the National Vulnerability Database. Since CVE records contain references to CWEs, OVAL results can be tied to specific CAPEC records. Connecting these three frameworks allows for a holistic understanding of a vulnerability found in a medical device for both upper management and device security specialists. Figure 3 shows two new tables (tblOval and tblCAPEC) that were required to support integration into the MedDevRisk relational database and provide connecting points using CVE and CWE identifiers: required schema expansion of MedDevRisk is discussed next.

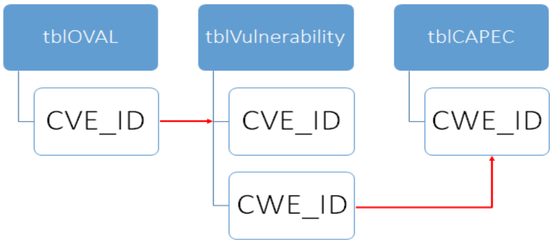


Figure 3. Key Schema Interactions

### 3.3 Schema Expansion

This research expands the original MedDevRisk framework by Seale et al. [11]. The author's database schema provides a peer-reviewed starting point for potential integration of new concepts such as attack patterns (CAPEC), real-time vulnerability assessment (OVAL), and continued use of standard frameworks such as CVE and CWE. Figure 4 shows the original schema, and Figure 5 shows the adapted version produced to support this case study. The original tables were either maintained (seen as brown in Figure 5) or modified (seen as green in Figure 5), and new tables and relationships were required (seen as yellow in Figure 5). Maintained tables with original intent include Cause, Control, ControlType, Threat

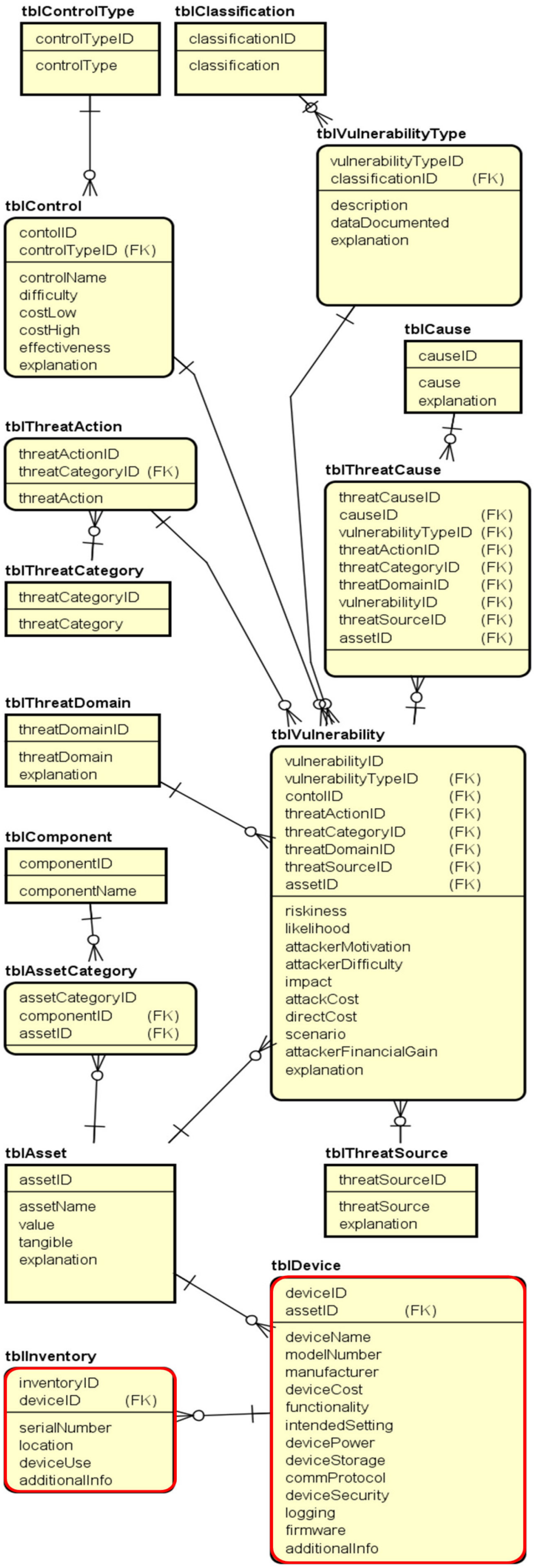


Figure 4. Original MedDevRisk Schema [15]

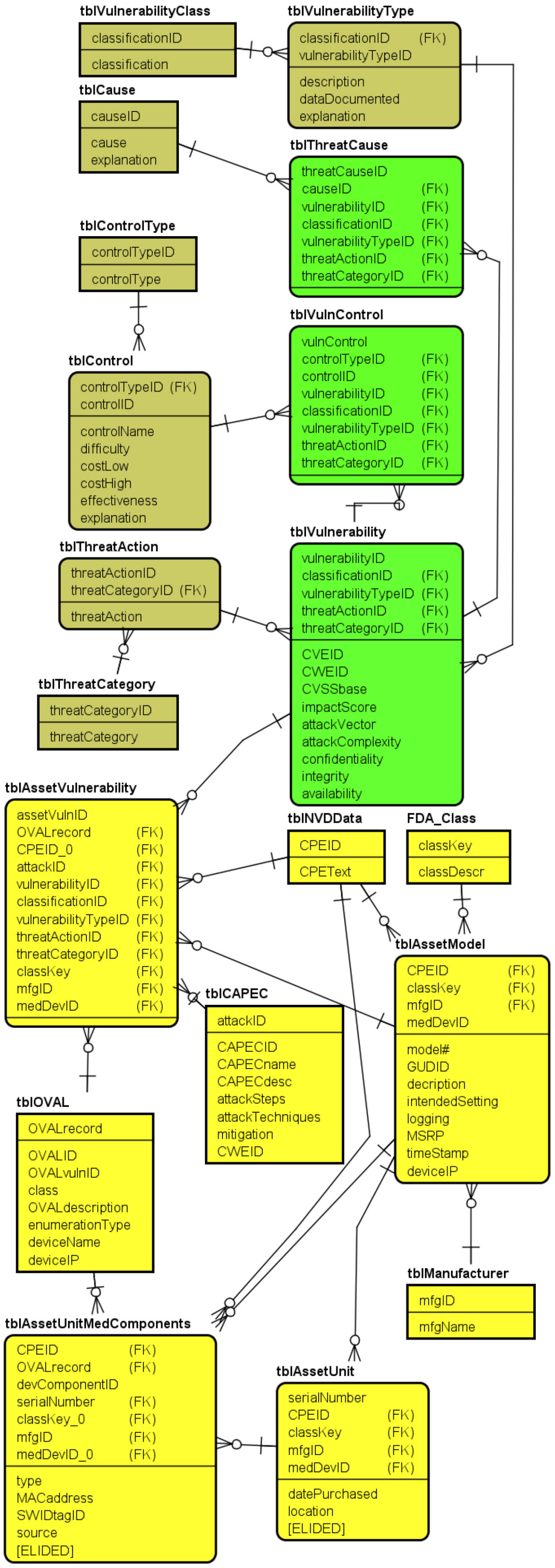


**Figure 5. Revised MedDevRisk Schema**

Action, ThreatCategory, VulnerabilityType, and VulnerabilityClass. The Control table is used to identify potential mitigation strategies for identified vulnerabilities that either do not have a corresponding CVE or CWE, which means these vulnerabilities do not have CAPEC values to identify community standard mitigation strategies. Table ControlType identifies what type of control the suggested mitigation is such as whether the control revolves around ‘Software Security’ or ‘Training and Awareness’. The ThreatAction table identifies whether a vulnerability is a threat to disclosing or manipulating health information through a specific attack, for example, a man-in-the-middle attack or a SQL injection attack. Table ThreatCategory groups specific threat actions together based on whether the threat will disclose information or manipulate information. Finally, table VulnerabilityClass classifies vulnerabilities by identifying to what technical area the vulnerability relates. For example, if a vulnerability is identified in a software application, the VulnerabilityClass would categorize that vulnerability in the ‘Software’ group.

Schema expansion during the case study also led to repurposing the Vulnerability table from an intersection table into a central combiner of data related to identified vulnerabilities and any related attributes. A new AssetVulnerability table now links assets, threats, and controls (mitigations). The Vulnerability table provides the storage location for the CVE links (found in the National Vulnerability Database [31]), the impact score (from community derived CVSS calculations and the version of CVSS used to compute it), associated the CWE identifier (MITRE’s type classification), and the relationship of the vulnerability to the traditional Confidentiality, Integrity, and Availability (CIA) security triad.

The implementation of this case study required the development and addition of new tables to the MedDevRisk schema. Table AssetModel contains information from both tblDevice and tblInventory from the original schema such as device name, model number, intended settings, and logging. The table AssetModel provides general information about an asset with a primary key that is automatically populated and incremented with each new record.

The AssetUnit table contains data about an individual serialized device. The data in this table are the individual instantiations of models in tblAssetModel. It contains information such as device serial number, purchase date, and location. The primary key for this table is a composite key of the primary key from tblAssetModel and the device’s serial number, which forms a one-to-many relationship. The AssetUnitMedComponents table contains the technical data related to an asset such as the MAC address(es), operating system, firmware, etc. and has a composite key of the a component ID that increments when a new component is added and keys from tblAssetUnit where there is one-to-many relationship (one unit, many components). The Manufacturer table contains the company name that created an asset along with an automated primary key that increments with the addition of a new record.

To support the analysis and integration of OVAL data, the OVAL table was created. Attributes such as device name and device IP connect records to certain assets and were needed to determine what records relate to what assets due to the potential of more than one asset being vulnerable to the same OVAL ID. Table AssetModel contains an attribute for device IP to connect it to the OVAL table. DeviceIP would have been a good attribute to use for a primary key in tblOVAL; however, device IP addresses could be considered sensitive information, which means the use of this data should be restricted. This led to the creation of an OVALrecord attribute that is automatically populated and incremented with the addition of a new record. The OVALID is the record identifier established by the OVAL community, which means one OVAL ID can be tied to many devices, and many devices can be connected to one OVAL ID. Due to a possible many-to-many relationship between assets and OVAL records, the OVAL ID is not used as the primary key. The class attribute identifies the type of OVAL ID definition.

For purposes of the case study, all OVALrecord identifiers have a class of 'Vulnerability'; however, OVAL can identify other classes such as inventory, miscellaneous, and patch. The OVALdescription provides information on the vulnerability tow which the device could be susceptible. The Attribute enumerationType is the test result of an OVAL scan, so the data in this column will be either 'true', 'false', or 'undetermined'. Finally, OVALvulnID is the CVEID that is connected to that specific OVAL definition. This record is how the OVAL table connects to Vulnerability table.

To integrate attack models, the CAPEC table was created with supporting attributes. The Attribute attackID is the primary key that is automatically assigned and incremented when a new record is added to the table. An automatic primary key was necessary because some identified vulnerabilities (CVEs) have more than one corresponding CWE. One CWE can be connected to multiple CAPEC records, so the attackID identifies one CAPEC record that is connected to a specific CWE. This removes potential many to many relationships between tblVulnerability and tblCAPEC. The CAPECID attribute is the identifier created by MITRE that allows a user to find that specific CAPEC's information in their online database. CAPECdesc delineates the attack that could be conducted when a device has a specific weakness. Attribute attackSteps describe the actions an adversary would perform when exploiting an asset's weakness while attackTechniques identifies how an attacker gains information that enables him/her to complete an attack such as how they obtain user credentials to gain access to a system. The mitigation attribute provides data about how cybersecurity specialists can prevent a certain attack from occurring.

### 3.4 Data Entry

Python scripts were used to automatically import the data gathered from OVAL, CVE, and CAPEC definitions and to execute the case study execution on target computers. Python version 3.7 was used to extrapolate data from the XML files provided by the OVAL community. Three scripts were created to transfer the gathered data. One script gathered the OVAL data listed in section 3.1, and two other scripts were created to gather the data listed in section 3.2.

### 3.5 Query Configuration

After importing data into a database derived from the expanded MedDevRisk schema (seen in Figure 5), existing queries from prior publications [11, 12] were utilized and adapted based on the extended schema. New queries were created to support extended reporting capabilities based on the addition of CAPEC and OVAL related data.

## 4. Case Study Analysis

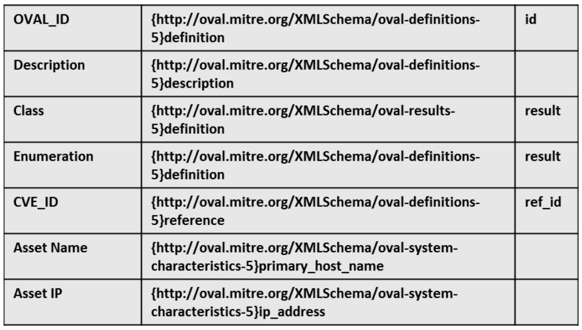
Data from the original MedDevRisk framework were combined with results of the case study methodology outlined in Section 3. New data included CAPEC information and OVAL support definitions, as well as enhanced CVE and CWE data from current National Vulnerability Data feeds. An SQLServer database was used to implement the expanded MedDevRisk schema (seen in Figure 5) as well as to store appropriate data and results.

The OVAL data collection aspect of the case study ran on four Mac devices, running various versions of the macOS operating system, that are part of the University of South Alabama Simulation Unit. The OVAL Intrepreter software was executed on each case study devices to generate device specfic

OVAL system characteristics files. The resulting data was collected, and scripts were executed to populate appropriate tables in the MedDevRisk database with OVAL analysis information. Analysis of the case study data covers the following four aspects: OVAL data results, enhanced CWE and CVE data results, and CAPEC integration results, as well as reporting capabilities, developed and executed as part of the case study.

### 4.1 OVAL Data Results

After execution of the OVAL analysis, Python scripts parsed XML files and mapped data tags to the OVAL table in the MedDevRisk database (as seen in Figure 6). As a result, 684 records were imported into MedDevRisk: 640 records were derived from CVE relationships, and 44 were derived from CPE relationships. Each target machine had 167 vulnerability records classified as unknown results, three classified false, and one vulnerability record classified true. As Figure 7 illustrates, the relational correlation to other tables from the OVAL records allowed identification of vulnerabilities based on specific applications installed on the four target machines. The results were as follows: 552 vulnerabilities related to Adobe applications, with 138 being distinct; 120 vulnerabilities were related to Microsoft applications, with 30 being distinct; and four vulnerabilities related to a combination of Adobe and Microsoft products, with one being distinct. Query execution allowed for the identification of eight vulnerabilities related to Apple security patches or file protocols, with two being distinct.



| | | |
|---|---|---|
| OVAL_ID | {http://oval.mitre.org/XMLSchema/oval-definitions-5}definition | id |
| Description | {http://oval.mitre.org/XMLSchema/oval-definitions-5}description | |
| Class | {http://oval.mitre.org/XMLSchema/oval-results-5}definition | result |
| Enumeration | {http://oval.mitre.org/XMLSchema/oval-definitions-5}definition | result |
| CVE_ID | {http://oval.mitre.org/XMLSchema/oval-definitions-5}reference | ref_id |
| Asset Name | {http://oval.mitre.org/XMLSchema/oval-system-characteristics-5}primary_host_name | |
| Asset IP | {http://oval.mitre.org/XMLSchema/oval-system-characteristics-5}ip_address | |

**Figure 6. OVAL Integration Tags**

The schema adaptation is designed to allow for the generation of OVAL reports continuously. It is also designed so that machines can be retested regularly, and either existing records in the schema are updated (for example, a required patch is applied that mitigates a known vulnerability), or new records are inserted based on new vulnerabilities identified in system configurations. One of the vulnerability results related to a weakness in a filing protocol on Mac operating systems 10.6.x through 10.6.4. This resulted in an ‘unknown’ identification. This should have returned a false designation since OVAL was only tested on machines with operating system versions 10.9.2, 10.9.5 and 10.12. Overall, the case study illustrated successful integration of OVAL data into the TVA relational model of MedDevRisk, as well as verifying that relational queries can be developed to link real-time/continuous threat monitoring with traditional threat/vulnerability mappings.

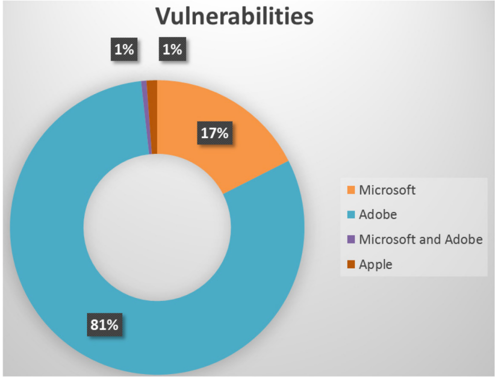


**Figure 7. OVAL Vulnerability Identification**

### 4.2 Enhanced CVE and CWE Data

To provide a current threat and vulnerability assessment, the case study also included data integration of National Vulnerability Database data from 2006 to 2018. This incorporates not only CVEs identified in the original MedDevRisk research [11] but also those CVEs that were identified through OVAL data integration of this case study. The data refresh resulted in 101,011 records imported into the extended MedDevRisk database. Of those records, 75,317 contain CWEs, 640 records connected OVAL and NVD entries (with 160 being distinct), and 544 records connected OVAL, CVE, and CWEs (with 136 being distinct).

### 4.3 CAPEC Integration

To integrate open source CAPEC data into MedDevRisk, Python scripts were created and executed to parse XML-based CAPEC formats into the extended schema of the case study. As a result, 1,206 records were imported. Of these records, 957 records had corresponding CWEs, and 249 did not; 933 records had a corresponding mitigation strategy, and 273 did not; 632 records had execution steps, and 574 did not, and 366 records had associated attack techniques, and 840 did not. Figure 8 summarizes the data. Import scripts were created that would continuously refresh data already in the database and automatically create new or update existing CVE records when CAPEC related data was encountered. CAPEC data also allows OVAL information to be linked through various queries to risk assessment, vulnerability, and mitigation reports.

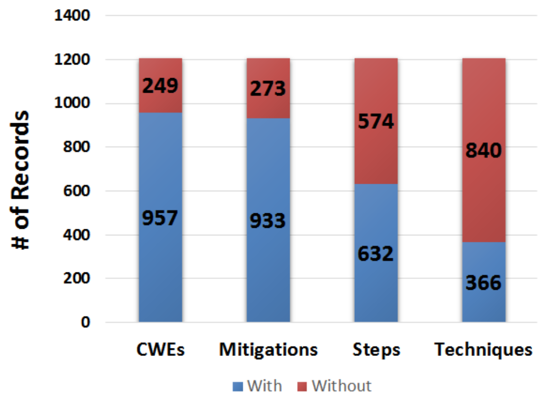


**Figure 8: CAPEC Integration/Data Quality**

As part of the case study, 544 imported OVAL records were identified with corresponding CWEs already in the database (corresponding to 136 distinct records). In terms of relating CAPEC to OVAL, the case study execution revealed that 6,556 records connect CAPEC to OVAL vulnerabilities total (with 153 being distinct). Of these, 6,052 records have associated mitigations (with 113 being distinct), and 504 records do not have mitigations (with 40 being distinct); 5,004 records had corresponding execution steps (with 72 distinct), and 1,552 records did not have associated execution steps (with 81 being distinct); 1,276 records had related attack techniques (with 30 being distinct), and 5,280 records did not have attack techniques (with 123 distinct attack techniques identified). Figure 9 summarizes the relational mappings identified through the case study.

## 4.4 Query and Reporting Capability

The case study resulted in the development of new queries, views, and generated reports through the MedDevRisk framework. As Table 1 summarizes, four distinct capabilities were developed: 1) STRIDE (STR) and TVA-based summary analysis; 2) OVAL-based vulnerability reporting; 3) mitigation reporting and CVE identification, and 4) adversarial viewpoint reporting. In MedDevRisk, STRIDE values are determined by Threat Actions (supported by tblThreatAction), and they are linked to CVE threat descriptions. In this case study, four threat actions were used with the CVEs produced by OVAL: 1) Disclose Health Information by Application-Layer; 2) Manipulate Health Information by Application-Layer; 3) Disclose Health Information by Backdoor Methods, and 4) Manipulate Health Information by Backdoor Methods.

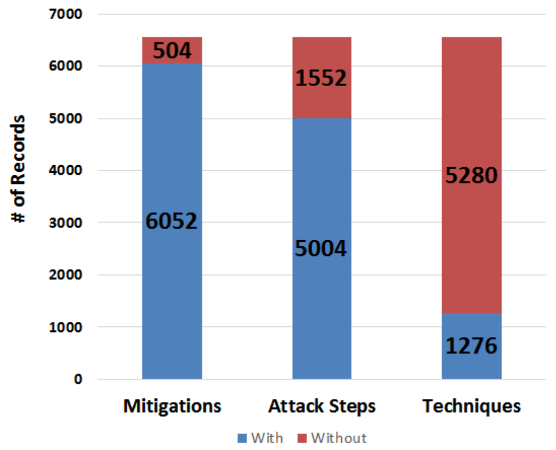


**Figure 9: CAPEC/OVAL Data Relationships**

**Table 1: Extended MedDevRisk Reporting**

| | STR | TVA | OVL | MIT | ADV |
|---|---|---|---|---|---|
| Device Name | X | X | X | X | X |
| Medical Device ID | | X | X | | |
| Asset Model Descr | | X | | | |
| Model # | | X | | | |
| CVSS Base Score | X | | X | | |
| Vulnerability Descr | X | X | | | |
| Vulnerability Class | | X | | | |
| Impact Score | | X | X | X | X |
| Attack Complexity | X | X | X | | |
| Attack Vector | X | X | X | | |
| CIA | X | | X | | |
| STRIDE Action | X | | X | | |
| STRIDE Motivation | X | | X | | |
| Threat Action | X | X | X | | |
| Threat Source | X | X | | | |
| Mitigation | | | X | X | X |
| CVE | | | | X | X |
| Device Name | | X | X | X | X |
| Attack ID | | | | X | |
| CAPEC ID | | | | | X |
| CAPEC Name | | | | | X |
| CAPEC Descr | | | | | X |
| Attack Steps | | | | | X |
| Attack Techniques | | | | | X |

From an adversarial view, the case study demonstrated a proof of concept for potential devices in medical environments that have known associated

CVEs. That basic link provides the ability to link the device to a risk score (CVSS), the attack steps of an adversary, the attack techniques used, and possible mitigations. This information forms the basis for actionable steps for IT personnel and CIOs who manage critical assets, including medical devices. Because PCs and computer workstations often provide soft targets for attackers, the reporting also allows security decision-makers to take a more holistic view of all assets that are part of healthcare environments.

## 5. Conclusions and Future Work

Several key goals framed the case study and methodology of this research. These goals included 1) expanding the current CVE data stored in MedDevRisk through the creation of a new entity in the relationship model; 2) implementing an attacker's point of view by integrating CAPEC attack patterns; 3) performing real-world system evaluation scans on medical devices and using OVAL to gather data from produced reports; 4) integrating OVAL data by creating tables to support relevant OVAL attributes; and 5) executing a risk assessment case-study using real-world data. These goals were accomplished through proof-of-concept development and a case study implemented in the Simulation Unit at the University of South Alabama.

The key contributions resulting from the case study are summarized in Figure 10. By adding OVAL and CAPEC values and expanding on CVE and CWE data, the MedDevRisk framework now provides support to cybersecurity specialists with community standard vulnerability data, mitigation strategies, and adversarial tactics for vulnerabilities medical devices could face. The reports created to evaluate this data provide upper management with information that can aid security specialists in creating attack models and mitigation strategies for networked medical devices.

The case study illustrates the real-world issues associated with integrating threat and vulnerability assessment tools into operational medical settings, which is normally complicated by a lack of physical access to machines in a healthcare domain. In part, the issue of access led to a smaller number of machines chosen for the proof of concept. The case study also illustrated an inability to connect CVEs to CAPEC records automatically (due to CWEs not being assigned in all cases) and also the lack of fully defined data in the open-source data (summarized in Figures 8 and 9). It is recognized that missing framework information causes some of the mitigation strategies to lack validity. While this research successfully created attack models and mitigation strategies for vulnerabilities discovered on networked medical devices or associated connected devices over a network, MedDevRisk can still be enhanced.

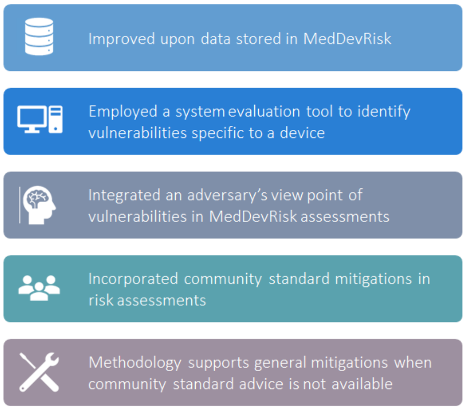

Improved upon data stored in MedDevRisk
Employed a system evaluation tool to identify vulnerabilities specific to a device
Integrated an adversary's view point of vulnerabilities in MedDevRisk assessments
Incorporated community standard mitigations in risk assessments
Methodology supports general mitigations when community standard advice is not available

**Figure 10: Key Contributions of Case Study**

An interesting area of future work identified by the study is the need to create OVAL definitions specifically for medical devices. One main issue that was found during the evaluation of previous and new vulnerability data is that only one used CVE was connected to a networked medical device. In order to accomplish this goal, researchers need physical access to devices and the skill set to be able to correctly evaluate the devices and identify any hardware or software vulnerabilities. Identification of vulnerabilities for specific medical devices enhances the data produced in risk reports as well as strengthening the healthcare cybersecurity posture.

## 6. References

[1] Pramanik, P.K.D., B.K. Upadhyaya, S. Pal, and T. Pal, *Internet of things, smart sensors, and pervasive systems: Enabling connected and pervasive healthcare*, in *Healthcare Data Analytics and Management*. 2019, Elsevier. p. 1-58.

[2] Pramanik, P.K.D., S. Pal, and M. Mukhopadhyay, *Healthcare Big Data: A Comprehensive Overview*, in *Intelligent Systems for Healthcare Management and Delivery*. 2019, IGI Global. p. 72-100.

[3] Leung, L.W.S., A. Roudsari, A. Kuo, and K.L. Courtney, *System Dynamics in Remote Monitoring Service for Cardiovascular Implantable Electronic Devices.* Studies in health technology and informatics, 2019. **257**: p. 277-282.

[4] Van Devender, M.S., W.B. Gisson, M. Campbell, and M.A. Finan. *Identifying Opportunities to Compromise Medical Environments*. in *Americas Conference on Information Systems (AMCIS)*. 2016.

[5] Van Devender, M.S., W.B. Glisson, R. Benton, and G. Grispos, *Understanding De-identification of Healthcare Big Data.* 2017.

[6] Andel, T., M. Campbell, W. Glisson, M. Jacobs, J. Mayr, and J. McDonald, *Compromising a Medical Mannequin.* 2015.

[7] American Medical Association. *Physician cybersecurity*. https://www.ama-assn.org/. Date of Last Access: 06/15/2019.

[8] US Department of Health Human Services. *Report on Improving Cybersecurity in the Healthcare Industry*. 2017. https://www.phe.gov/Preparedness/planning/CyberTF/Documents/report2017.pdf. Date of Last Access: 06/15/2019.

[9] Grispos, G., W.B. Glisson, and P. Cooper, *A Bleeding Digital Heart: Identifying Residual Data Generation from Smartphone Applications Interacting with Medical Devices.* arXiv preprint arXiv:1901.03724, 2019.

[10] Grispos, G., W.B. Glisson, and K.-K.R. Choo. *Medical cyber-physical systems development: a forensics-driven approach*. in *Proceedings of the Second IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*. 2017.

[11] Seale, K., J. McDonald, W. Glisson, H. Pardue, and M. Jacobs. *MedDevRisk: Risk Analysis Methodology for Networked Medical Devices*. in *Proceedings of the 51st Hawaii International Conference on System Sciences*. 2018.

[12] Seale, K.A., *Integrating relational data frameworks into risk assessment of networked medical devices*. 2017: University of South Alabama.

[13] Luckett, P., J.T. McDonald, and W.B. Glisson, *Attack-graph threat modeling assessment of ambulatory medical devices.* arXiv preprint arXiv:1709.05026, 2017.

[14] Holdsworth, J., W.B. Glisson, and K.-K.R. Choo, *Medical device vulnerability mitigation effort gap analysis taxonomy.* Smart Health, 2017.

[15] Alexander, B., S. Haseeb, and A. Baranchuk, *Are implanted electronic devices hackable?* Trends in Cardiovascular Medicine, 2018.

[16] Eddy, N. *Healthcare executives lack action plan to combat cybersecurity threats*. 2019. https://www.healthcareitnews.com/news/healthcare-executives-lack-action-plan-combat-cybersecurity-threats. Date of Last Access: 06/15/2019.

[17] Eddy, N. *Infusion pump-linked workstations contain critical security flaw*. 2019. https://www.healthcareitnews.com/news/infusion-pump-linked-workstations-contain-critical-security-flaw. Date of Last Access: 06/15/2019.

[18] OVAL. *OVAL*. https://oval.cisecurity.org/. Date of Last Access: 06/15/2019.

[19] MITRE Corporation. *CAPEC-Common Attack Pattern Enumeration Classification*. 2019. https://capec.mitre.org/. Date of Last Access: 06/15/2019.

[20] Hodges, B.J., *Attack Modeling and Mitigation Strategies for Risk Based Analysis of Networked Medical Devices*. 2019, University of South Alabama.

[21] Hollasch, L.W. and M. Stroshane. *Designing with Security Threat Models*. 2017. https://docs.microsoft.com/en-us/windows-hardware/drivers/ifs/designing-with-security-threat-models. Date of Last Access: 06/15/2019.

[22] Michelman, E.H. and L.J. Hoffman, *SECURATE: a security evaluation and analysis system using fuzzy metrics*. 1977: Electronics Research Laboratory, College of Engineering, University of California.

[23] Whitman, M.E., *Enemy at the gate: threats to information security.* Communications of the ACM, 2003. **46**(8): p. 91.

[24] Forum of Incident Response and Security Teams (FIRST). *Common Vulnerability Scoring System SIG*. https://www.first.org/cvss/. Date of Last Access: 06/15/2019.

[25] Common Vulnerabilities and Exposures. *Common Vulnerabilities and Exposures* https://cve.mitre.org/. Date of Last Access: 06/15/2019.

[26] Microsoft. *The STRIDE Threat Model*. 2009. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20). Date of Last Access: 09/01/2019.

[27] Pardue, J., J. Landry, and S. Purawat, *A Database-driven Model for Risk Assessment.* 2014.

[28] Corporation, M. *Common Platform Enumeration*. https://cpe.mitre.org/. Date of Last Access: 06/15/2019.

[29] Corporation, M. *CWE-Common Weakness Enumeration*. https://cwe.mitre.org/. Date of Last Access: 09/01/2019.

[30] Oates, B., *Researching Information Systems and Computing*. 2006, London: Sage Publications. 341.

[31] National Institute of Standards. *National Vulnerability Database*. https://nvd.nist.gov/. Date of Last Access: 09/01/2019.