Information & Management 57 (2020) 103334

Contents lists available at ScienceDirect

Information & Management

journal homepage: www.elsevier.com/locate/im

Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system

Romilla Syed

Department of Management Science and Information Systems, College of Management, University of Massachusetts, Boston, 100 Morrissey Blvd., Boston, MA 02125, United States

ARTICLE INFO

Keywords:
Cybersecurity vulnerability
Vulnerability management
Social media intelligence
Ontology
Conceptual modeling
Cyberalerts

ABSTRACT

Effective vulnerability management requires the integration of vulnerability information available on multiple sources, including social media. The information could be used to inform common users about impending vulnerabilities and countermeasures. First, we present the Cybersecurity Vulnerability Ontology (CVO), a conceptual model for formal knowledge representation of the vulnerability management domain. Second, we utilize the CVO to design a Cyber Intelligence Alert (CIA) system that issues cyber alerts about vulnerabilities and countermeasures. We rigorously evaluated the CVO as well as the accuracy, performance, and usefulness of the CIA system. Key contributions of this study to research and practice are discussed.

1. Introduction

Recently, there has been an increase in the number of cybersecurity vulnerabilities discovered and reported. For instance, Common Vulnerabilities and Exposure (CVE), an industry standard for vulnerability and exposure identifiers, published 16,555 vulnerabilities between January 1, 2018, and December 31, 2018[1] [2] . This is the highest number of disclosed vulnerabilities in the last ten years. Additionally, in 2014, CVE adopted a new format for assigning vulnerability identifiers that no longer limits the number of CVE-IDs to 10,000 per year[3] . Furthermore, prior research notes several pathways to vulnerability disclosure [50,67]. While the Computer Emergency Response Team Coordination Center (CERT/CC) is the official body to disclose vulnerabilities publicly in the U.S., discoverers can also make vulnerability information available through social media, blogs, and wikis[4] . Particularly, research suggests that social media users often disclose and share vulnerabilities before official sources [e.g., 73, 88]. While most of the vulnerabilities are exploited between the time a vulnerability is discovered and a patch is installed [46], severe vulnerabilities such as Heartbleed and VENOM that received high exposure on social media were quickly exploited [70].

Although vulnerability information is disclosed and shared through multiple sources, including the CERT/CC, security agencies, mailing lists, as well as social media, subscribers to these sources are usually security professionals, vendors, or hackers [66,67]. In the extant literature, there has been some focus on vulnerability management [e.g., see 28, 95, and 96]; however, there is a lack of research focusing on the integration of vulnerability information from multiple sources that could be used to inform and protect common users from vulnerability exploits. Additionally, the integration of social media intelligence for vulnerability management has not received much attention. A few recent studies have proposed conceptual models to integrate social media information for cyber intelligence [e.g., 46, 61, and 70]; however, these studies do not account for concepts of vulnerability domain as described in the National Institute of Standards and Technology (NIST) handbook [16,28].

Informed by the preceding literature, we contend that the prevention of cyber exploits will require the integration of vulnerability information from multiple official sources with social media intelligence. The vulnerability information can be presented in a structured

E-mail address: Romilla.Syed@umb.edu.

[1] CVE is an international cybersecurity community effort. CVE is industry-endorsed through the CVE Numbering Authorities (CNAs), CVE Board, and several other organizations that include CVE identifiers in their products, services, alerts, and advisories. The CNAs are organizations across the world that are authorized to assign CVEs to vulnerabilities related to their products. See https://cve.mitre.org/cve/request_id.html#cna_participants for a list of participating CNAs. The CVE Board is comprised of several cybersecurity organizations including vendors, academia, and research and government institutions, among others. See https://cve.mitre.org/community/board/index.html#current_members for a list of current and past board members.

[2] https://www.cvedetails.com/vulnerability-list/year-2018/vulnerabilities.html (Accessed on April 15, 2019)

[3] https://cve.mitre.org/cve/identifiers/syntaxchange.html (Accessed on Jun 21, 2018)

[4] In this study, when we mention the CERT/CC, we mean the U.S. based CERT/CC unless otherwise noted.

https://doi.org/10.1016/j.im.2020.103334
Received 22 October 2018; Received in revised form 4 June 2020; Accepted 7 June 2020
Available online 11 June 2020
0378-7206/ Published by Elsevier B.V.

ontological format that will allow security analysts to assess the severity of vulnerabilities and issue cyber alerts about vulnerabilities and countermeasures to common users. Consequently, this study addresses the following research question: *How can we design a vulnerability management ontology to better inform users about impending vulnerabilities and countermeasures?*

In the extant literature, ontologies have been used to represent knowledge in a formal and structured format that supports better organization, communication, and reusability of knowledge as well as better computational inferences [36,42]. Ontologies model relationships on data and use the explicit knowledge of concepts and relationships to deduce implicit knowledge about the domain [1]. A well-defined ontology presents a formal logical system based on concepts and subconcepts, taxonomies, facts, rules, relationships, properties, axioms, and constraints pertaining to a particular domain [38]. Additionally, prior research has proposed ontologies for representing the vulnerability management domain [e.g., 1, 26]. However, existing ontologies are based on generic information security concepts and lack the context of vulnerabilities as specified by the CERT/CC and Common Vulnerability Scoring System (CVSS) framework[5] . CVSS is a published standard used by organizations worldwide to assess and prioritize the vulnerability management process properly. Existing ontologies also do not account for vulnerability information available on social media.

To that end, we first present a conceptual ontology, which we refer to as the *Cybersecurity Vulnerability Ontology (CVO)*. We use the conceptual modeling technique to design the CVO [93]. In information systems (IS) development, conceptual modeling is used in early stages to build a representation of selected semantics of a real-world domain [98]. A high-quality conceptual model forms the basis for designing and programming the system. In this study, the CVO provides a formal knowledge representation of the vulnerability management domain by integrating the vulnerability concepts defined by the NIST, CERT/CC, and CVSS. Additionally, the CVO maps the vulnerability concepts extracted from Twitter, a popular microblogging site. Next, we utilize the CVO to design an ontology-based alert system referred to as the *Cyber Intelligence Alert (CIA) system* to issue cyber alerts based on underlying rules. We evaluated the content and design of the CVO in a focus group session with domain experts to ensure the adequacy of the domain representation. The accuracy, performance, and usefulness of the CIA system are also rigorously evaluated.

As a design science research, this study contributes to research by designing two Information Technology (IT) artifacts – CVO and CIA [see 10]. The CVO conceptualizes the vulnerability domain based on multiple official sources and as well as leverages social media intelligence for vulnerability management. The CIA system utilizes and extends the CVO with concepts and properties required to issue cyber alerts. At the practical level, the CVO can be used as a general vocabulary of the vulnerability management domain. Vendors may find the CVO as a useful aid for their vulnerability analysis and management. Additionally, official agencies such as the CERT/CC and security analysts may find the CIA system beneficial to alert common users about new vulnerabilities, exploits, patches, and workarounds.

The remainder of the paper is organized as follows. We first present a review of existing research on cybersecurity vulnerability disclosure and vulnerability management ontologies. This is followed by a discussion on design science research methodology. We then present the conceptual modeling approach for ontology design and the emergent CVO. Next, we discuss the conceptual framework, architecture, and design of the CIA system. We discuss approaches for evaluating the CVO and CIA system and corresponding results. Finally, we discuss contributions of this study to research and practice. The paper concludes with a discussion on limitations of this study and future research areas.

[5] https://www.first.org/cvss/ (Accessed on Jun 21, 2018)

2. Background literature

Two bodies of research inform this study. The first relates to cybersecurity vulnerability disclosure, and the second relates to ontologies for vulnerability management.

2.1. Cybersecurity vulnerability disclosure

The US-CERT/CC defines cybersecurity vulnerability as a weakness in the computational logic (e.g., code) found in software and some hardware components (e.g., firmware) that, when exploited, results in negative impact on the confidentiality, integrity, or availability of information assets. Vulnerabilities disclosure involves gathering information from vulnerability discoverers, coordinating the disclosure of that information between relevant stakeholders, and disclosing the existence of vulnerabilities and the mitigation strategies to stakeholders, including public [44]. The stakeholders typically include a *discoverer* – an individual or organization that finds the vulnerability, *reporter* – an individual or organization (often same as the discoverer) that notifies the vendor about the vulnerability, *vendor* – an individual or organization that creates and maintains the product containing the vulnerability, *deployer* – an individual or organization that deploys the patch and takes other remedial actions, *coordinator* – an individual or organization that facilitates and coordinates the incident handling and response, and *general public or consumers* – an individual or organization who uses the vulnerable product. Because of several stakeholders involved in the process, the CERT/CC advocates the term *Coordinated Vulnerability Disclosure* to promote cooperation among different stakeholders and thereby protects the end users [44].

There are several pathways to vulnerability disclosure [50,67]. A discoverer may choose to report the vulnerability to the CERT/CC. The CERT/CC is tasked by the U.S. Government to serve as a trusted third-party in the vulnerability coordination and disclosure process. The Common Vulnerability Exposure (CVE) and the National Vulnerability Database (NVD), both sponsored by the CERT/CC, are two major entities involved in vulnerability disclosure. Usually, the CERT/CC provides a protective window of 45 days to vendors for releasing a patch. At the end of 45 days, CERT/CC discloses the vulnerability information to the public. This pathway is referred to as deferred disclosure. Alternatively, a discoverer may choose to disclose the vulnerability immediately by sharing it through mailing lists, such as BugTraq. A discoverer may also choose to sell vulnerabilities to agencies, such as iDefense and TippingPoint, for monetary rewards. Finally, a hacker may sell a vulnerability on the black market or may use it to attack a system. Such vulnerabilities are disclosed after an exploit is published, or a security analyst detects an intrusion. An exploit is a software that utilizes the vulnerability to achieve some effect [44]. The effect could be simply to demonstrate the existence of the vulnerability or to attack systems.

Recently Sen and Heim [75] studied the impact of vulnerability disclosure on publishing exploits, which is the appearance of an exploit software in the public domain. Sen and Heim [75] note that disclosing vulnerabilities to the public on the same day they are discovered, that is immediate disclosure, increases the risk of exploits becoming available publicly. Furthermore, the higher the severity of the vulnerability, the greater is the hazard of exploit publication. Ransbotham and Mitra [66] analyzed the impact of immediate vulnerability disclosure by security professionals on attack diffusion and volume. The authors note that immediate disclosure accelerates the risk of dissemination and penetration of attacks, but decreases the volume of the attack. However, vulnerabilities disclosed for monetary rewards delay the diffusion of attacks and reduce the risk of the first attack and the volume of attacks [67]. Furthermore, patched vulnerabilities attract more attacks than unpatched vulnerabilities [5]. Vulnerabilities are also noted to impact the market value of companies [86]. A related body of research examines vendors patch release behavior and disclosure policies

[4,54,87]. Prior research suggests that private agencies such as iDefense and TippingPoint do not influence the disclosure timing of public agencies, such as the CERT/CC [54]. Furthermore, withholding software vulnerability disclosure for a long time does not increase patch quality [6].

Recently, there has been some focus to explore the role of social media in vulnerability disclosure. Messages with the root-cause, patch, advisory, or exploit information are retweeted more than messages that simply alert about a new vulnerability [83]. Moreover, vulnerabilities shared by influential users generate more retweets. Although social media provides intelligence about vulnerabilities, it is time-consuming to gather and integrate intelligence from multiple heterogeneous sources. Hence, some attempts have been made to propose social media-monitoring approaches. For instance, Trabelsi et al. [88] proposed a vulnerability-monitoring system based on the security intelligence collected from Twitter. The authors found that Twitter users disclose vulnerabilities before official sources. Other studies have proposed methods to detect exploits, calculate risk, and prioritize response action based on social media feeds [46,61,70]. In a related study, Benjamin et al. [11] proposed a machine-learning approach to collect information from hacker forums, Internet Relay Chats, and carding shops for detecting cyber attacks. Finally, Lippmann et al. [55] proposed a human language technology classifier to detect cyber discussions on Stack Exchange, Reddit, and Twitter.

2.2. Ontologies

The International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) in their recent standards document, ISO/IEC:21838-1, define an ontology as[6] :

a collection of terms, relational expressions, and associated natural language definitions together with one or more formal theories designed to capture the intended interpretations of these definitions.

In the extant literature, Guarino [38] defines an ontology as "… an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words […] In the simplest case, an ontology describes a hierarchy of concepts related by subsumption relationships; in more sophisticated cases, suitable axioms are added in order to express other relationships between concepts and constrain their intended interpretation" (p.2). Ontologies allow formalized knowledge representation implemented in an ontology language.

Guarino [38] further notes that "an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e., its ontological commitment to a particular conceptualization of the world" (p. 7). The formal logic is specified using ontological vocabulary organized into a taxonomic class hierarchy. The vocabulary consists of concepts or classes, relationships, axioms, properties (attributes), and instances. The hierarchy is defined by a single root concept, referred to as *Thing* to which other concepts (classes) are related. Concepts are related to each other by *is-a* relation. Furthermore, concepts can be initiated by creating instances, which are referred to as individuals. The class instances can also be related to each other by relations other than *is-a*, which are referred to as object properties. Relations between classes lead to a hierarchical domain representation, whereas relations between instances lead to a network domain representation. The range and domain of object properties can be set as directed or restricted. The range and domain specify the individual type and define the object property. Finally, axioms can be added to object properties to represent principles in the conceptual domain and limit the possible interpretations of defined concepts [3].

In the extant literature, several ontologies have been proposed to represent the information security domain. Recently, Rosa et al. [69] provided a review of security ontologies and taxonomies for risk assessment. Mainly security ontologies can be classified as *generic* and *specialized ontologies*. Generic ontologies aim to model the main concepts of the information security domain [35,48,71,79,103]. The specialized ontologies model subdomains such as risk management [13,28,56], security policies [25], incident analysis [15,60], and attack patterns [51,91]. In contrast to ontologies, the taxonomies represent an extended vocabulary, a glossary, or a list of security concepts. The existing taxonomies describe both generic information security metrics [45,74] and specialized concepts such as security attacks [22], security risks [64,94,97], and threats and defenses [7,30]. Taxonomies to model specialized domains such as mobile application environments [27] and the Internet of Things (IoT) [9] have also been proposed.

2.2.1. Vulnerability management ontologies

In this study, we conceptualize an ontological representation of the cybersecurity vulnerability domain, which is a specialized application domain of information security. In the literature, several ontologies focusing on vulnerabilities have been proposed. For instance, based on the existing information security literature, Fenz and Ekelhart [28] present a generic conceptual ontology for the information security domain with some focus on vulnerabilities. The purpose of the ontology is to provide a model and knowledge base for the information security domain. The security ontology is based on the security relationship model described in the NIST handbook [16]. The main concepts include asset, threat, vulnerability, and control. The authors, however, extended their proposed ontology with additional concepts referencing several sources, including German IT Grundschutz manual [32], Internet security glossary [76], and cryptosystems taxonomy [14,41]. The ontology was evaluated by a group of experts using formal and informal competency questions. While some questions focused on relations between vulnerabilities and related controls, the ontology does not focus on the comprehensive representation of the vulnerability domain.

Wang et al. [1,95] proposed an ontology for vulnerability management (OVM). Informed by the description logic knowledge engineering approach [8], the ontology conceptualizes the vulnerability domain based on CVE and related standards. OVM captures important concepts such as vulnerability, attack, and countermeasure, as well as relations among concepts to characterize software vulnerabilities. The ontology is populated with vulnerability data from NVD. The formal logic coded in OVM allows to assess the security level of software products as well as to identify similar vulnerabilities. While OVM is one of the earliest attempts to formalize the vulnerability management domain, the ontology is limited to NVD and does not consider vulnerability concepts from other important sources [28]. Additionally, ontology concepts and reasoning capability are not evaluated for adequate domain representation.

In another study, several ontologies, including the security attack ontology, security defense ontology, and security asset-vulnerability ontology, are integrated to develop a comprehensive vocabulary on security attacks and countermeasures [90]. Particularly, the asset-vulnerability ontology integrates concepts from other ontologies and presents vocabulary to link security policies and goals with countermeasures to mitigate the exploitation of vulnerabilities. While the focus of asset-vulnerability ontology is on the broader information security domain, the authors note that the design of the asset-vulnerability ontology is based on their expertise of the information security domain as well as knowledge derived from previous studies. However, the correctness and usefulness of the ontology are not evaluated.

Informed by the conceptual modeling technique, Elahi et al. [26] proposed a vulnerability ontology to analyze the effect of vulnerabilities on a system. The proposed ontology is designed to help security analysts elicit security requirements related to vulnerabilities. However, the ontology is an abstract metamodel, and the correctness and usefulness of the ontology are not evaluated. Ontology-based systems to

[6] https://standards.globalspec.com/std/13386210/iso-iec-dis-21838-1 (Accessed on January 17, 2020)

Table 1
Approach to design the CVO and CIA system.

Design Science Activity*	Description*	Designing the CVO and CIA System
1. Problem identification and motivation	Define the research problem and justify the value of a solution	- Prevention of cybersecurity exploits require integration of vulnerability information from official sources with social media intelligence - Existing vulnerability management ontologies are limited to generic security concepts specified by the NIST and do not account for vulnerability information provided by the CERT/CC and CVSS - Little research has focused on integrating social media intelligence for cybersecurity vulnerability management - There is also a lack of research focusing on designing systems to alert common users about vulnerabilities and countermeasures - See Sections 1 and 2
2. Objectives of the solution	Infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible	- Develop a Cybersecurity Vulnerability Ontology (CVO) to represent the domain. The CVO integrates vulnerability concepts provided by the NIST, CERT/CC, CVSS, and as well as concepts extracted from Twitter - Develop a Cybersecurity Intelligence Alert (CIA) system to alert common users - See Sections 1 and 2
3. Design and development	Create the artifact	*Designing the CVO* - Researched the literature to identify appropriate concepts for representing the vulnerability domain - Used the conceptual modeling approach to design the CVO - See Section 4 *Designing the CIA System* - Developed an ontological representation of the CVO using Protégé software - Designed and developed other components of the CIA system to issue cyber-alerts - See Section *5*
4. Demonstration	Demonstrate the use of the artifact to solve one or more instances of the problem	- We collected the vulnerability information shared on Twitter - We also collected the data about authors' of tweet postings - Next, we collected the vulnerability information from CVE and NVD databases - Finally, we collected product and patch information from vendor sites - We instantiated the CIA system to issue cyber alerts based on reasoning rules - See Section *5*
5. Evaluation	Observe and measure how well the artifact supports a solution to the problem	- The CVO is evaluated by domain experts to determine the correct representation of the domain - The accuracy, performance, and usefulness of the CIA system is evaluated - The other subcomponent of the CIA system, "Social Media Intelligence Extractor-Tagger" is also rigorously evaluated - See Section 6
6. Communication	Communicate research process and outputs in publications	- We presented details about the need, design approach, and usefulness of artifacts in this manuscript

* The activities and description are as per the design science research process by Peffers et al. [65].

predict and classify web application attacks have also been proposed [72]. In their study, Salini and Shenbagam [72] utilize the ontology-based approach for attack prediction using the context of vulnerabilities, threats, and consequences. Specifically, this study proposes three ontology models: threat model, vulnerability model, and attack model. Models are linked to derive inferences and predict attacks. However, the three ontological models do not present comprehensive subconcepts. For instance, instead of specifying vulnerability subconcepts such as severity, impact, and products impacted, authors simply list different types of vulnerabilities such as Denial of service, SQL injection, and client spoofing in their vulnerability model. Likewise, instead of specifying threat characteristics, authors list different types of threats such as eavesdropping, passwords sniffing, and code injection in their threat model. Although experimental results suggest that the approach has a high prediction rate for web application attacks, the design of the three ontological models is not evaluated.

Finally, Mittal et al. [61] adopted the Unified Cybersecurity Ontology (UCO) to provide their proposed vulnerability management system with cybersecurity domain information. Authors note that vulnerabilities are temporal in nature, as vulnerability information can be considered vital only for a specific time. UCO, however, is not suitable for temporal reasoning over cybersecurity events [84]. Moreover, UCO does not contextualize the vulnerability information publicly available on the Internet. To integrate the temporal characteristics of vulnerabilities, Mittal et al. [61] enhanced their system by creating another ontology referred to as intelligence ontology. The intelligence ontology stores the temporal information of cybersecurity events such as the number of tweets related to vulnerability, first and latest tweet posting time, and products impacted. However, the ontology does not represent the comprehensive set of vulnerability concepts [see 28]. Additionally, patch-related details are not included in the ontology, which is important to discard vulnerabilities that have already been addressed by vendors.

In conclusion, existing research has enhanced our understanding of vulnerability management. The need for an ontological representation of the vulnerability domain has also been noted in the literature. However, there are two gaps.

(1) The existing vulnerability ontologies are modeled on generic information security concepts [1,26] and lack the context of vulnerability domain as specified by the US-CERT/CC. Furthermore, existing ontologies do not model vulnerability characteristics as provided by the CVSS framework, an industry standard to assess the severity of vulnerabilities. Additionally, vulnerability information is also made available through social media. While calls to analyze and extract vulnerability information from traditional and non-traditional data sources, including social media, have been made [61,84], existing ontologies do not model vulnerability concepts from social media. It is important to integrate social media intelligence for vulnerability management, as research suggests that social media users disclose vulnerabilities before official sources [88]. Further vulnerabilities that receive high social media attention can be quickly exploited [70].
(2) While vulnerability information is available through multiple sources, including the CERT/CC, mailing lists, security agencies as well as social media, the subscribers are usually security

professionals, vendors, or hackers [66,67]. To the best of our knowledge, there is no system currently that integrates the information from multiple sources to inform or alert common users about vulnerabilities and countermeasures. Recently there have been attempts to conceptualize social media-based models for generating cyber alerts about vulnerabilities [61]. However, the model does not consider vulnerability information from other official sources [28]. Moreover, the model does not account for patch details, which is important to identify vulnerabilities that pose the highest risk of exploitation. It is also important to consider social media users who post the vulnerability information, which establishes the trustworthiness of information [88] and influences the subsequent diffusion of vulnerability information on social media [83].

In this study, we address noted gaps by designing a conceptual cybersecurity vulnerability ontology. The CVO integrates the vulnerability concepts from several sources, including the NIST, CERT/CC, CVSS framework, and Twitter. Next, we utilize the conceptual CVO to design an ontology-based CIA system. Based on underlying reasoning rules, the CIA system generates cyber alerts to inform common users about vulnerabilities and countermeasures.

3. Design science research methodology

In the literature, several techniques are used to design and develop artifacts. A design artifact refers to "a thing that has, or can be transformed into a material existence as an artificially made object (e.g., model, instantiation) or process (e.g., method, software)" [34, p. 341]. In this study, we follow a generalized design science research approach proposed by Peffers et al. [65] to design two IT artifacts that are the CVO and CIA system. Table 1 summarizes our design approach.

This research is motivated by the fact that the prevention of cybersecurity exploits requires the integration of vulnerability information from multiple official sources with social media intelligence for security analysts to assess the severity and inform common users about vulnerabilities and countermeasures. While prior studies have proposed an ontological representation of the vulnerability domain, existing ontologies are limited to generic information security concepts defined by the NIST and do not consider vulnerability concepts specified by the CERT/CC and CVSS framework [1,26]. Additionally, while recent research suggests the relevance of social media for vulnerability management [61,83], the integration of social media intelligence with the generic vulnerability concepts has not received much attention. Hence, in this study, we first design a conceptual ontology referred to as the CVO to represent the vulnerability management domain. The CVO integrates vulnerability concepts from the NIST, CERT/CC, CVSS as well as Twitter. Next, we design the CIA system that utilizes the CVO for issuing cyber alerts to common users based on underlying reasoning rules.

The resources required to move from research objectives to design and development requires the knowledge of theory that can be brought to bear in a solution [65]. At the design stage, we ensured that the CVO is grounded in ontological theory. Specifically, based on prior research [92,98], this study is informed by Bunge's [19] ontological theory. According to Bunge's [19] theory, the world is comprised of constructs such as things, properties, attributes, schema, states, events, interactions, and systems. While *properties* are intrinsic to *things*, *attributes* represent characteristics assigned to a thing. A thing is modeled as a *functional schema* represented by properties and attributes. A *state* represents a set of values representing the functional schema of a thing. Bunge [19] further notes that things change, which are manifested as changes in states of things. A change of state is referred to as an *event*. States and changes are constrained by properties referred to as *laws*. Finally, a *system* can be viewed as a thing or an aggregate of interacting things. The *interaction* among things occurs as a change of state of one thing is affected by the presence of another thing.

Additionally, we systematically reviewed the existing literature to ensure mapping of vulnerability concepts with the security domain. We conducted an in-depth review of vulnerability, security ontologies and taxonomies, and software vulnerability literature to identify appropriate techniques and concepts that could be used to inform the ontology design. In this study, we adopted the conceptual modeling technique to design the CVO. Our choice is informed by prior IS research that purports the relevance of Bunge's [19] ontological theory for conceptual modeling [see 92, 98]. Additionally, the conceptual modeling technique is used in early stages of IS development to build a representation of selected semantics of a real-world domain [98]. The conceptual model is then used as a basis for designing and programming the system. Details of the conceptual modeling technique and the resultant conceptual ontology are described in the next section. We utilized the CVO to design the CIA system. In particular, we used Protégé software to implement the ontology. We also implemented other components of the CIA system, which we describe in Section 5.

We demonstrated the application of the CIA system by instantiating the ontology with representative data collected from multiple sources. We implemented the predefined reasoning rules to issue cyber alerts. Details of the application serve as a proof of concept for the alert system [65]. Finally, we adopted several approaches to assess the adequacy of the design artifacts [59,89]. Specifically, the CVO is evaluated by domain experts using a criteria-based evaluation approach to ensure the adequacy of the domain representation [37,89]. We also rigorously evaluated the accuracy, performance, and usefulness of the CIA system. Evaluation details and results are presented in Section 6.

4. Design of the cybersecurity vulnerability ontology (CVO)

In this section, we discuss the conceptual modeling approach for designing ontologies and the resultant CVO.

4.1. Conceptual modeling approach

Conceptual modeling refers to building a formal representation of a specific phenomenon pertaining to a real-world domain [93]. The conceptual modeling approach uses predefined concepts and rules for building a graphical representation of a domain. Concepts, rules, and symbols used for specific modeling purposes constitute a modeling language, which is implemented using a modeling tool [31,81]. In IS development, conceptual modeling usually occurs in early stages of system design. A high-quality conceptual model not only allows users to design and analyze real-world domains effectively but also allows the early detection and correction of errors in domain representation [98]. Prior research shows that conceptual models have superior cognitive effectiveness in comparison to other software engineering approaches [21,52,62]. Additionally, in the extant literature, ontological theories have been used to inform conceptual modeling approaches. As Weber [98] argues, the ontological theory proposed by Bunge [19] lies at the core of conceptual modeling. Bunge's [19] ontological theory provides the basic constructs to represent a real-world phenomenon in the conceptual model.

In this study, we choose conceptual modeling to understand the broader social and technical context of cybersecurity vulnerability management [93]. In addition, we use conceptual modeling as an overarching methodology for analyzing and designing the vulnerability domain [2,81]. The process of developing a domain ontology first requires choosing an upper-level ontology to base the domain ontology and then developing the domain ontology with reference to the upper-level ontology. One can use an already existing upper-level ontology or develop a new ontology [85]. In doing so, the analyst identifies and describes relevant domain concepts, determines upper-level concepts that domain concepts refer to, and identify constraints (or business rules) that affect concepts [63,82]. Following this process, we based the

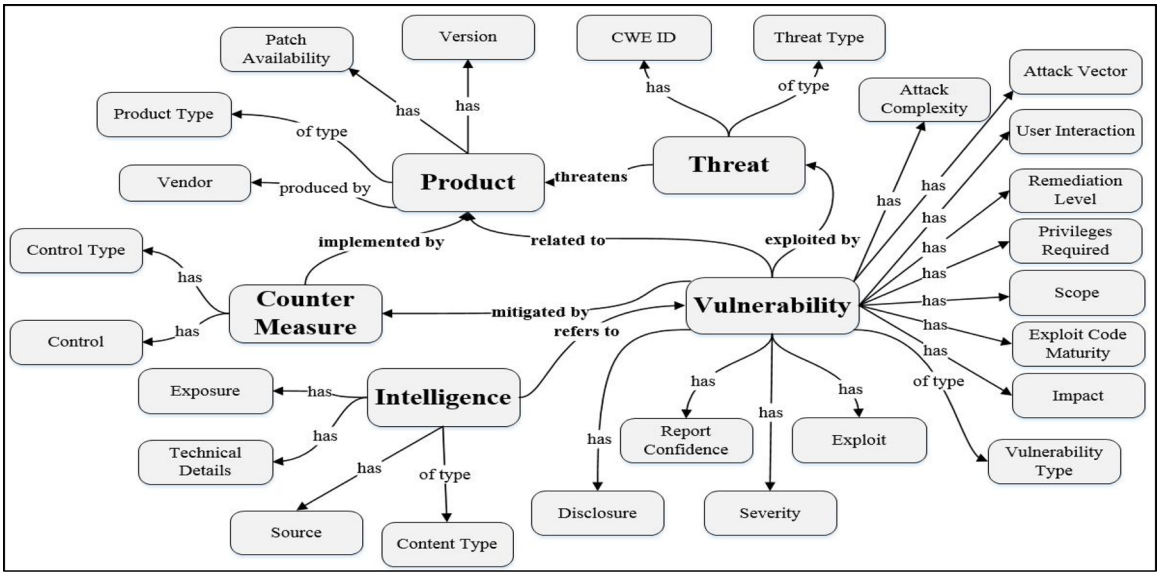

Fig. 1. Conceptual Cybersecurity Vulnerability Ontology (CVO).

CVO on well-established upper-level ontology informed by the NIST handbook [16,28]. To identify relevant lower-level ontological concepts, we analyzed the literature on vulnerability management and social media, and existing ontologies for representing the vulnerability management domain.

We now exemplify steps to refine our domain ontology based on the *vulnerability* concept. The first step is to identify relevant domain concepts. The *vulnerability* concept is the most obvious as it represents the critical component of the problem domain. Next, the concept must be described adequately. One can borrow the definition from the literature, or define the new concept if a definition is not available. This is followed by the categorization of the concept into upper-level ontology concepts. For instance, the *vulnerability* concept refers to the upper-level concept *thing*. The categorization necessitates the definition of properties, relationships, and constraints among concepts. For instance, the *vulnerability* concept is related to *thing* concept by *is-a* relationship and to *threat* concept by *is-threatened-by* relationship. One of the relevant properties of the *vulnerability* concept, for instance, the *impact* could be used as a major criterion for managing threats. Finally, one must identify constraints or business rules of the domain. We relied on the CVSS framework to define underlying rules. For instance, vulnerability severity is determined by the impact of the vulnerability on the confidentiality, integrity, and availability of a system.

4.2. Cybersecurity vulnerability ontology (CVO)

Fig. 1 illustrates the conceptual CVO. The ontology complies with the information security standards and taxonomy provided by the NIST handbook [16], CERT/CC, and CVSS framework. Moreover, the ontology is enhanced with vulnerability concepts extracted from social media intelligence.

The ontology consists of five core concepts: vulnerability, intelligence, threat, product, and countermeasure. NIST defines vulnerability as the weakness in computational logic, which could be exploited by a threat [16]. Following CVSS metrics specifications[7], the *Vulnerability* concept consists of nine subconcepts: *Attack Vector* (i.e., the context by which vulnerability exploitation is possible), *Attack Complexity* (i.e., conditions under which an attacker can exploit a vulnerability), *Privileges Required* (i.e., privileges required by an attacker to successfully exploit a vulnerability), *User Interaction* (i.e., whether user interaction other than that of hacker is required to exploit a vulnerable system), *Scope* (i.e., whether exploiting a vulnerability can affect resources beyond authorization privileges intended by the vulnerable component), *Impact* (i.e., impact on the confidentiality, integrity, and availability of a system), *Exploit Code Maturity* (i.e., the probability that a vulnerability will be attacked), *Remediation Level* (i.e., the extent that a vulnerability can be solved), and *Report Confidence* (i.e., amount of information known about a vulnerability). Additionally, based on CERT/CC specifications, we added four more subconcepts: *Severity* (i.e., the base score ranging from 0 to 10 and the related qualitative severity ranking), *Exploit* (i.e., the number of times a vulnerability is exploited), *Vulnerability Type* (i.e., the vulnerability type provided by CVE[8]), and *Disclosure* (i.e., whether a vulnerability has immediate or deferred disclosure, and the date of disclosure).

Our *Intelligence* concept is informed by Twitter data [61,83]. Following Mittal et al. [61], we define intelligence as actionable information about vulnerabilities posted on social media that interests a human analyst. We defined four subconcepts: *Content Type* (i.e., whether a tweet text contains alert, advisory, patch, exploit, or root-cause information), *Exposure* (i.e., the number of times a vulnerability is (re) tweeted); *Technical Details* (i.e., the URL and Hashtags referred in a tweet posting), and *Source* (i.e., whether a tweet author is a security professional, security firm, or other) [83].

Next, following Fenz and Ekelhart [28], p.1], we define *Threat* as "a potential danger to the assets and affects specific security attributes as soon as it exploits a vulnerability in the form of a physical, technical, or administrative weakness, and it causes damage to certain assets." The threat concept consists of two subconcepts: *Threat Type* (i.e., the cause of vulnerability as specified in the Common Weakness Enumeration (CWE) Specification[9]) and *CWE ID* (i.e., a unique identifier). CWE specification provides a common language of discourse for discovering and solving software vulnerabilities found in code, system design, or architecture. Each CWE ID represents a single vulnerability type.

The *Product* concept is defined as software that can have potential vulnerabilities [4,87]. The product concept consists of four

[7] https://www.first.org/cvss/specification-document (Accessed on June 26, 2018)

[8] https://www.cvedetails.com/vulnerabilities-by-types.php (Accessed on June 26, 2018)

[9] https://nvd.nist.gov/vuln/categories (Accessed on June 26, 2018)

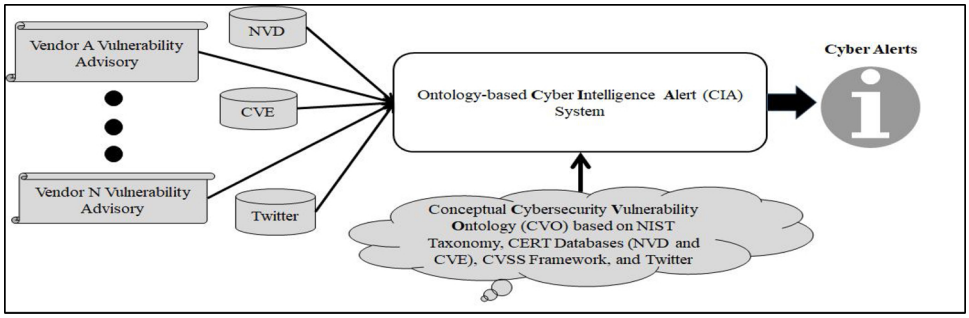

Fig. 2. Conceptual model of the Cyber Intelligence Alert (CIA) system.

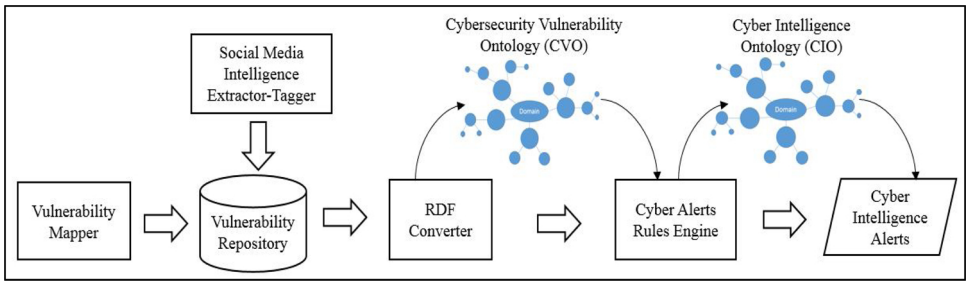

Fig. 3. The architecture of the Cyber Intelligence Alert (CIA) system.

subconcepts: *Product Type* (i.e., the product classification provided by CVE[10]), *Vendor* (i.e., the name of the vendor who created the product), *Version* (i.e., the product version number), and *Patch Availability* (i.e., whether a patch is available). Finally, following prior research [26,28], we define the *Countermeasure* concept as a protection mechanism instituted to secure the system. Countermeasures can include actions, processes, devices, solutions, or systems such as firewalls, authentication protocols, and digital signatures. Informed by German IT Grundschutz manual [32], we defined two subconcepts for Countermeasure: *Control Type* (i.e., infrastructure, organization, personnel, and hardware and software) and *Control* (i.e., the measures to mitigate a vulnerability).

5. Design of the cyber intelligence alert (CIA) system

Fig. 2 presents the conceptual model of the CIA system. At a high level, the CIA system collects vulnerability information from multiple sources, including CVE, NVD, Twitter, and vendor sites. The CIA system integrates and reconciles heterogeneous information as per the vulnerability domain, which is represented by the CVO. The CIA system then enhances the CVO with additional concepts required to issue cyber alerts and generates the Cyber Intelligence Ontology (CIO). The CIO serves as a knowledge base for alert information.

Fig. 3 presents the architecture of the CIA system. The main components include the vulnerability repository, Social Media Intelligence Extractor-Tagger (SMIET), vulnerability mapper, RDF converter, CVO, CIO, and cyber alerts rules engine. We discuss each of the components in the following subsections.

5.1. Cyber intelligence dataset and vulnerability repository

The purpose of the CIA system is to issue alerts based on the vulnerability information available from official sources such as the CERT/CC and vendor sites, and social media such as Twitter in our case. In developing the alert system, we started with Twitter to collect vulnerability information. We used Twitter's public API to retrieve tweets related to hashtag "#CVE." For each tweet, we also retrieved the author's information. Next, we extracted the distinct CVE-IDs mentioned in our tweets to collect the corresponding vulnerability information available on CVE and NVD databases, both maintained by the CERT/CC. The CVE database also allowed us to identify products impacted by each vulnerability. We then collected the relevant patch-related information from corresponding vendor sites. The final dataset has 13,277 tweets containing 3389 distinct CVE codes related to 236 products posted by 2865 unique users. Following prior research [24,53], we divided tweets into training (90 %) and testing (10 %) sets, while ensuring that both sets represent different types of vulnerabilities. Our training set comprised of 11,950 tweets related to 2,977 vulnerabilities and 214 products posted by 2,303 users. Our testing set comprised of 1,327 tweets related to 912 vulnerabilities and 71 products posted by 1,593 users.[11]

We also included data for threats and countermeasures. Specifically, as noted earlier, we used the German IT Grundschutz manual to create a list of countermeasures [32]. The manual lists four types of controls: infrastructure, organization, personnel, and hardware and software. In addition, several controls are specified for each control type. For instance, typical hardware and software-related controls include password protection for IT systems, anti-virus software, encryption, checksums, and digital signatures. The manual also lists suggested controls for different types of vulnerabilities, which helped us to write rules described in Section 5.6. Finally, we used the NIST classification for threat data. As noted earlier, our threat data refer to the CWE specification provided by the NIST. CWE classifies vulnerability causes into three categories: code, design, and system architecture. The CVE database lists the CWE-ID for each vulnerability. This allowed us to categorize each vulnerability into code, design, or system architecture.

5.2. Social media intelligence extractor-tagger (SMIET)

The SMIET labels tweets to indicate content categories. Based on

[10] https://www.cvedetails.com/product-list.php (Accessed on June 26, 2018)

[11] The number of CVE-IDs, users, and products in training and testing sets may not add up to corresponding total numbers as each tweet may refer to multiple vulnerabilities, and each vulnerability may be tweeted by multiple users and may relate to multiple products.

prior research [83], we classified tweets into five content categories: 1) *alert* – tweets that simply alert about a new vulnerability; 2) *advisory* – tweets that offer advisory information about a vulnerability; 3) *patch* – tweets that provide patch information; 4) *exploit* – tweets that share exploit or proof-of-concept information; and 5) *root cause* – tweets that provide technical details about the vulnerability. Additionally, based on the tweet author's profile description, SMIET classifies each author either as a *security professional*, a *security firm*, or *other* [83]. In doing so, we assert that tweets posted by *security professionals* or *firms* have more influence in diffusing the vulnerability information than those posted by *others*.

We used a two-step process to label tweets and tweet authors. First, two research assistants manually tagged 1500 tweet postings and 350 tweet author descriptions. The intercoder reliability was measured using Cohen Kappa and was quite strong both for content categories ($\kappa = 0.89$) and author profiles ($\kappa = 0.93$). Next, we used Natural Language Processing (NLP) technique to extend coding to a full set of tweets and tweet authors in the training set. As noted earlier, we used 90 % of tweets to train the model, and 10 % of the data to test the model. We applied supervised NLP and utilized manually coded tweets as a training set. While there are several NLP methods available, we followed the *ensemble learning* technique for data coding (see [24]). Ensemble learning is a machine learning technique that combines the output of several different classifiers to improve the accuracy of classification. By combining methods, ensemble learning overcomes the tradeoff in recall and precision of individual classification methods. It is also widely used for extracting information from social media data [49,53]. Following Lee et al. [53], we combined the statistical classifiers with a rule-based approach to generate the classification. Specifically, we trained the classifier in four steps.

1) We first cleaned raw tweet postings that were previously coded by research assistants. We used Python NLP framework, *NLTK* [12], for stopwords removal (i.e., removing punctuations and articles), tokenization (i.e., breaking tweets into words and phrases referred to as "tokens"), stemming (i.e., reducing a word to its basic form, e.g., "coding" reduces to "code"), and part of speech tagging (i.e., determining parts of speech in a tweet posting) [47].
2) Next, an algorithm extracts tweet-level attributes and tweet structure rules to identify content categories and tweet authors[12] . Some examples of tweet-level attributes and tweet-structure rules are: collecting all words in tweets and frequency of words in each tweet (bag-of-words approach), adjacent words (bigrams), the ratio of part of speech (nouns, verbs, etc.) used, Term Frequency and Inverse Document Frequency (TF-IDF) weighs each word based on their occurrence in the entire dataset and each tweet [49,53], and "whether a specific keyword is present" rule.
3) We used the rule-based method to identify the occurrence of certain words for classifying tweets based on content categories and tweet author profiles. The key to designing a successful classifier is to determine what do human interpreters notice about tweets classified into different categories. We obtained a set of training tweets that were previously tagged manually by researchers. Specifically, we identified keywords in the tweet text that researchers used to determine tweet content categories. For instance, tweets informing about patches have a high chance of containing words "patch," "fix," and "update." We also identified keywords that were used to determine tweet author profiles. This allowed us to categorize tweets based on "whether a specific keyword is present" rule. At the end of this step, the algorithm generates a matrix that correlates tweet-level attributes generated as above (the *x*-variables) corresponding to a series of binary (content present or not present) content labels coded by research assistants (the *y*-variables).
4) Next, we trained a classification model for each binary content label by combining multiple statistical classifiers. The statistical classifiers are binary classification machine learning models. We applied the ensemble method by combining results of different classifiers [53]. In our case, we combined the Support Vector Machine that achieved high precision and low recall with Naive Bayes based classifier that achieved high recall and low precision. We assessed the performance of the classifier rigorously, the details of which are presented in Section 6.2.1. Based on the labeling output of SMIET, we retained only tweets that are classified into at least one of the five content categories, which is important to generate correct alerts. We repeated steps until the entire 90 % of the data are used to train the model with desired performance measures.

5.3. Vulnerability mapper

A cybersecurity vulnerability may be tweeted multiple times. Furthermore, a vulnerability may impact multiple vendors. Hence, we wrote a program referred to as vulnerability mapper, to map vulnerabilities with vendors. For each vulnerability, the vulnerability mapper extracts the CVE-ID of the vulnerability mentioned in a tweet and then maps the corresponding information provided by vendors, CVE, and NVD with the tweet. Moreover, a vulnerability may be updated by CVE or NVD multiple times. For instance, while a vulnerability may be disclosed publicly, patch details may become available at a later time. To accommodate the historical information, we need to store all instances of a particular vulnerability. The vulnerability mapper does just that. Particularly, we created a temporal property *hasLastUpdatedTime* for both *Vulnerability* and *Intelligence* concepts. This property stores the system timestamp of when a vulnerability was last updated. The vulnerability mapper populates the value of the property for each vulnerability instance.

5.4. RDF converter and cybersecurity vulnerability ontology (CVO)

The vulnerability repository stores the vulnerability information from multiple heterogeneous sources. Hence, it is imperative that concepts and terms might be ambiguous. For instance, McCafe[13], like many other software vendors, uses their own vulnerability scoring levels, which could differ from the CVSS scheme. The ability of the CIA system to map and retrieve values of semantically similar concepts is essential to issue correct alerts. An ontology is capable of representing the equivalence among concepts, and thus address the semantic heterogeneity. In this study, we used the conceptual CVO to integrate vulnerability information from multiple sources to represent the domain.

Following the conceptual design, we used Protégé 5.2 software to build the CVO. As shown in Fig. 4, we presented five core concepts of the CVO (i.e., Vulnerability, Intelligence, Threat, Product, and Countermeasures) and the related subconcepts as classes and subclasses. We will use the term "class" and "subclass" from now on. Each data point in the vulnerability repository becomes an individual. The relationship between classes is defined by property. Finally, we used the Cellfie plugin available in Protégé 5.2 to import and store the vulnerability data in the CVO as RDF triples consisting of a subject, a predicate, and an object.

5.5. Cyber intelligence ontology (CIO)

The CIA system represents a specific application area of the CVO. We created a second ontology referred to as the Cyber Intelligence Ontology (CIO) to issue cyber alerts (see Fig. 5). We enhanced the CIO

[12] https://www.nltk.org/book/ch06.html (Accessed on October 1, 2019)

[13] https://www.mcafee.com/enterprise/en-us/threat-center/product-security-bulletins.html (Accessed on June 28, 2018)

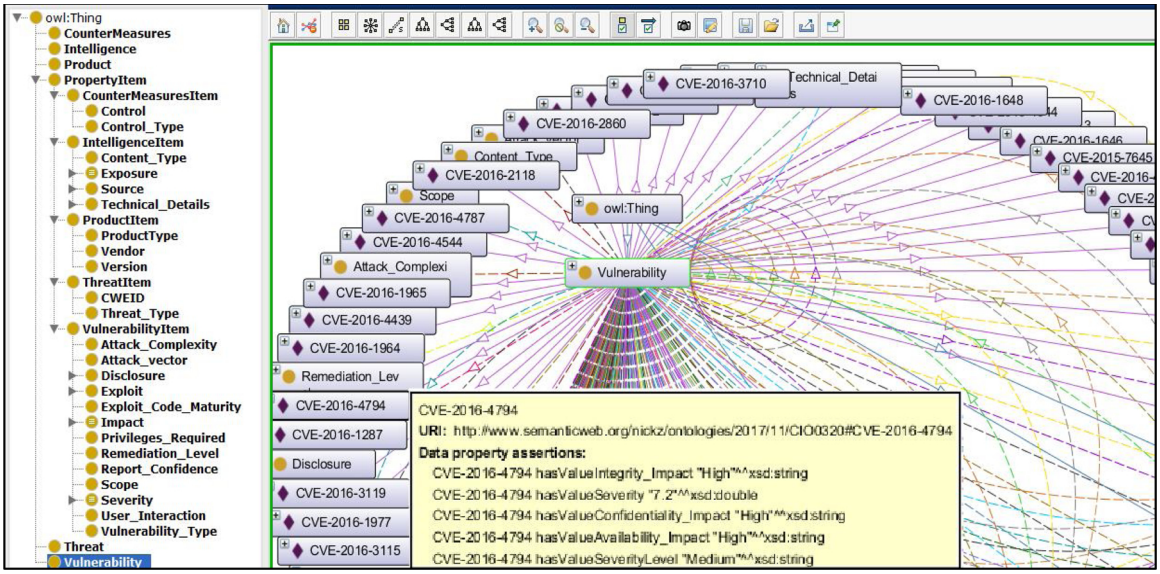

Fig. 4. Cybersecurity Vulnerability Ontology (CVO): Classes and subclasses (left side) and instances of the Vulnerability class (right side) along with values of an instance in a box.

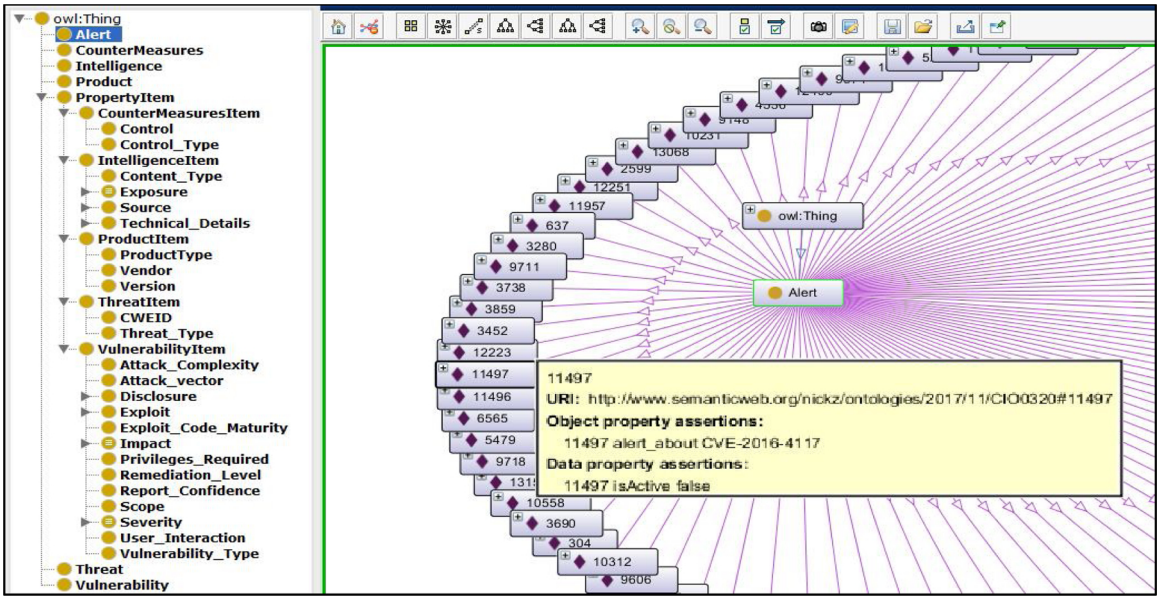

Fig. 5. Cyber Intelligence Ontology (CIO): Alert class with its subclasses (left side) and instances (right side) along with values of an instance in a box.

with additional concepts and inferred properties, which will be utilized to issue cyber alerts. Specifically, we added the concept *Alert* to centralize all active cybersecurity alerts. As noted before, it is important to alert common users about an impending vulnerability and available countermeasures. The *Alert* class refers to the *Vulnerability* class. Based on prior research [23], we represent alerts for each vulnerability as instances of *Alert* class in the CIO. We used the Cellfie plugin to import all instances of the CVO into the CIO. During the import, the plugin creates a corresponding alert instance for each vulnerability. The data property *isActive* is set to Null by default for each *Alert* instance.

Additionally, we included 12 data properties in the CIO presented in Table 2. Following prior research [78], we used SWRL rules to infer data properties. SWRL rules consist of two parts, antecedent (or conditions) and consequent (or action) (see [43] for details on SWRL rules). A SWRL rule can be interpreted as: if the antecedent holds true, then the consequent must hold. For instance, the following SWRL rule sets the value of data property *hasDisclosure* to "deferred" or "immediate" depending on values of patch release date and vulnerability published date. If *hasValuePatchReleaseDate* is less than *hasValuePublish_Date*, then *hasDisclosure* is set to "deferred" otherwise, it is set to "immediate." The SWRL rule is written as follows:

Vulnerability(?v) ^ hasValuePatchReleaseDate(?v,? y) ^ hasValuePublish_Date(?v,? z) ^ temporal:before(?y,? z) - > hasDisclosure (?v, "deferred")

Vulnerability(?v) ^ hasValuePatchReleaseDate(?v,? y) ^ hasValuePublish_Date(?v,? z) ^ temporal:after(?y,? z) - > hasDisclosure (?v, "immediate")

Likewise, based on the CVSS specification, we set the data property *hasSeverityLevel* of a vulnerability to "critical" if the severity score of the vulnerability is greater than 9, and is set to "high" if the severity score of the vulnerability is between 7 and 9. The SWRL rule is coded as follows:

Vulnerability(?v) ^swrlb:greaterThanOrEqual(?s, 7) ^ hasValueSeverityScore(?v,? s) ^ swrlb:lessThan(?s, 9) - > hasSeverityLevel (?v, "High")

Vulnerability(?v) ^ swrlb:greaterThanOrEqual(?s, 9) ^

Table 2
Data properties of the Cyber Intelligence Ontology (CIO).

Data Property	Description
hasDisclosure(vul:Vulnerability, X)	This data property is set to "immediate" if the patch release date is later than the vulnerability disclosure date. Otherwise, the data property is set to "deferred"
hasSeverityLevel(vul:Vulnerability, Y)	This data property represents the qualitative rating of the severity score. As per CVSS 3.0 specification, if the score is 0, the value is set to "None"; if the score is between 0.1 and 3.9, the score is set to "Low"; if the score is between 4.0 and 6.9, the score is set to "Medium"; if the score is between 7.0 and 8.9, the score is set to "High"; if the score is between 9.0 and 10.0, the score is set to "'Critical"
hasLastUpdatedTime(vul:Vulnerability, Z)	This data property stores the latest timestamp of a when vulnerability information is updated or tweeted
isOpen(int:Intelligence, L):	This data property determines whether a vulnerability is open. An open vulnerability is one for which the patch is not available yet and is temporally significant
hasTweetCounter(vul:Vulnerability, M)	This data property determines the number of times a vulnerability is tweeted
hasPoCExploit(int:Intelligence, N)	This data property is set to True if Proof-of-Concept (POC) exploit information is available in the wild
hasPatch(vul:Vulnerability, O)	This data property is set to True if a patch is available
isSoftwareType(ven:Vendor, P)	This data property indicates whether a vulnerability impacts "Application software" or "System software"
isVendorType(ven: Vendor, Q)	This data property indicates whether a vendor is "Open source" or "Proprietary"
isActive(ale: Alert, A1)	This data property is set to True if the intelligence is temporally significant
hasAlertTime(ale: Alert, A2)	This data property stores the system time when an alert is generated
hasAlertEndTime(ale: Alert, A3)	This parameter can be set by a security analyst. For this study, we set this as "PatchReleaseTime" + 24 h, after which the alert becomes inactive

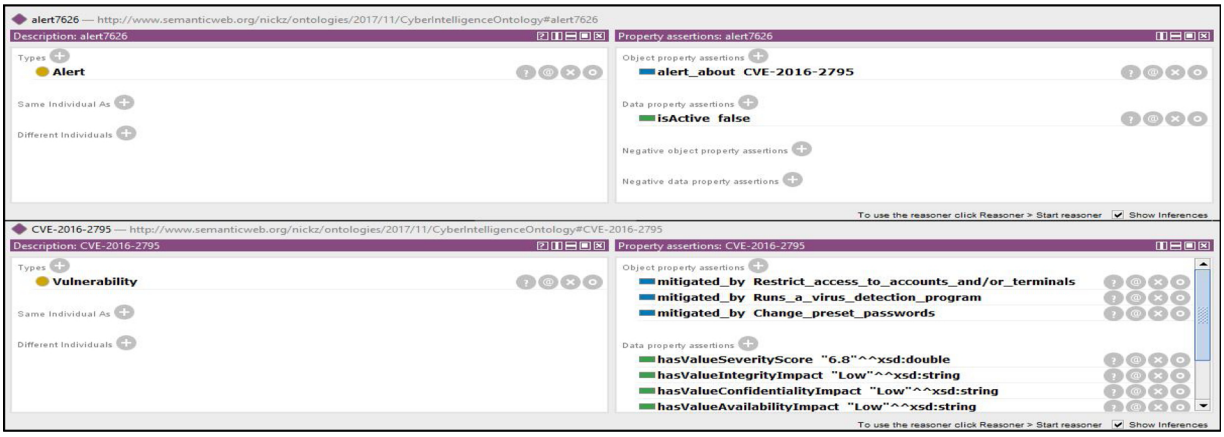

Fig. 6. An example of a cyber-alert instance.

hasValueSeverityScore(?v,? s) - > hasSeverityLevel (?v, "Critical")

5.6. Cyber alerts rules engine

Finally, the CIA system reasons over the knowledge base to determine if an alert should be raised to inform users about vulnerability and possible countermeasures. Fig. 6 shows an instance of an alert. Following prior research [61], we use the data property *isOpen(int:Intelligence, L)* to determine whether an "Intelligence" is "Open" and then issue the alert for open intelligence. We used the SWRL rule to compute the inferred property *isOpen(int:Intelligence, L)*, which depends on how "new" is the intelligence and whether a patch is available for the corresponding vulnerability. We defined a system parameter *T* to specify the time of "new" intelligence. We set the parameter to 24 h, which indicates that the intelligence instance is updated in the last 24 h. Following is an example of SWRL rule when the intelligence is new, and a patch is not available:

Intelligence(?i)^Vulnerability(?v)^ refers_to(?i,?v) ^ hasPatch (?v,false) ^ hasValueFirstTweetTime(?i,?t)^ temporal:durationLessThan (24,? i,"now", "Hours")- > isOpen(?i, true)

We also specified a few more SWRL rules to raise alerts. For instance, an alert is raised if *isOpen(int:Intelligence, L)* is "open," and the severity level of a vulnerability is "high." This sets the property *isActive* for the alert instance as "True."

Alert(?a) ^ Intelligence(?i)^Vulnerability(?v)^ refers_to(?i,?v) ^alert_about(?a,? v)^ isOpen(?i, true)^ hasSeverityLevel (?v, "High") - > isActive(?a, true)

Another example of a rule to raise an alert is if *isOpen(int:Intelligence, L)* is "open" and the vulnerability has *hasTweetCounter(int:Intelligence, M)* equal to or greater than the set threshold. For instance, we set the threshold to 10, which indicates the vulnerability has been retweeted at least 10 times. The SWRL rule is written as follows:

Alert(?a) ^ Intelligence(?i)^Vulnerability(?v)^ refers_to(?i,?v) ^alert_about(?a,? v)^ isOpen(?i, true)^hasTweetCounter (?v,? c) ^swrlb:greaterThanOrEqual(?s, 10) - > isActive(?a, true)

Finally, we set alerts as "False" after 24 h of patch release time. The SWRL rule is written as follows:

Alert(?a) ^alert_about(?a,?v) ^ Vulnerability(?v) ^ hasPatch (?v,true) ^ hasValuePatchReleaseDate(?v,?t) ^ temporal: durationGreaterThan(24,? t, "now", "Hours") - > isActive(?a, false)

A user can also query the knowledge base to retrieve alert information from the CIO. Usually, the SPARQL interface is used to query ontologies, which is recommended by the W3C as the standard query language[14] . The query uses PREFIX clauses to define namespaces and identify queried classes and properties. For instance, a query to find all active alerts with "high" severity is shown in Fig. 7. The variables *alert* and *vulnerability* that meet a set of conditions are requested: *alert* variable is of type *Alert*, which has the property *alert_about*. The *alert_about* property refers to a vulnerability instance.

6. Evaluation

In this section, we discuss approaches to evaluate the CVO and CIA

[14] https://www.w3.org/TR/sparql11-query/ (Accessed on June 22, 2018)

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX cio: <http://www.semanticweb.org/nickz/ontologies/2017/11/CyberIntelligenceOntology#>
SELECT ?alert ?vulnerability ?severity_level
WHERE {
?alert rdf:type cio:Alert ;
cio:alert_about ?vulnerability.
?vulnerability rdf:type cio:Vulnerability .
?alert cio:isActive true .
?vulnerability cio:hasSeverityLevel ?severity_level.
FILTER (?severity_level = "High")
}
```

alert	vulnerability	severity_level
alert4191	CVE-2016-1734	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert7990	CVE-2016-2851	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert12578	CVE-2016-4538	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert13262	CVE-2016-4951	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert13264	CVE-2016-4951	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert5691	CVE-2016-2098	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert5773	CVE-2016-2098	"High"^^<http://www.w3.org/2001/XMLSchema#string>
alert5710	CVE-2016-2098	"High"^^<http://www.w3.org/2001/XMLSchema#string>

Fig. 7. An example of a SPARQL query.

system and corresponding results.

6.1. Evaluation of the cybersecurity vulnerability ontology (CVO)

6.1.1. Ontology evaluation approaches

In the literature, scholars have proposed several approaches to evaluate ontologies, which can be classified into five categories [17,102]. First, the *"golden standard" evaluation* approach compares the proposed ontology with the existing golden standards of the domain [58]. However, this approach may not be feasible if the "golden" standard is not available, or the quality of the standard is questionable [18,39]. Second, the *data-driven evaluation* approach compares the proposed ontology to a data corpus [18]. This approach is designed to measure the coverage of the ontology rather than the correctness, clarity, or usefulness of the ontology [39]. Third, the *human evaluation* approach assesses an ontology on predefined criteria. For instance, Lozano-Tello and Gómez-Pérez [57] proposed a method to assess the suitability of existing ontologies concerning system requirements. However, this approach is not suitable for evaluating ontologies that are designed from scratch [39]. Fourth, the *application or task-based evaluation* approach evaluates the proposed ontology by applying it to a specific application or task and assessing the correctness of results [17,102]. Although this approach is appropriate to evaluate the potential of an ontology to achieve its objective, appropriateness of the ontology content and design is not validated [39]. Finally, the *criteria-based evaluation* approach evaluates the proposed ontology against a set of predefined criteria [37]. This approach verifies the content and design of the ontology. In the literature, several criteria are defined to evaluate ontologies [39]. For instance, Wand and Weber [93] created a set of criteria to measure the quality of ontology design along two dimensions, *clarity* and *completeness*. Yu et al. [101,102] propose *coverage* as another criterion for ontology evaluation.

In this study, we adopted two approaches to evaluate the CVO [17,89]. First, we applied the criteria-based approach to evaluate the CVO. The criteria-based approach verifies the content and structure of manually constructed ontologies [17]. The purpose is to ensure that the ontology meets certain predefined design principles or criteria. Moreover, the content evaluation focuses on concepts, instances, and facts that have been included in the ontology, and the vocabulary used to represent these concepts. The structural evaluation focuses on the organization of the ontology and its suitability for further development. In this study, following Haghighi et al. [39], we selected seven criteria to evaluate the content and structure of the CVO. As criteria-based evaluation is conducted by human experts [17], we evaluated the CVO in a focus group session with domain experts. Furthermore, manual evaluation is suitable to evaluate the CVO as we based the ontology on well-established upper-level ontology informed by the NIST handbook [16,28,39]. We discuss details in the next subsection.

Second, we applied the application or task-based evaluation to the CVO. As noted before, an ontology is typically used in some application or task. The output or performance of the application partly depends on the underlying ontology. Therefore, a potentially effective ontology evaluation approach is to determine how useful a particular ontology is in the context of an application. A well-designed ontology helps the application that uses the ontology to achieve good results [17]. In this study, we utilized the CVO to design the CIA system that raises cyber alerts. Hence, we evaluate the correctness and usefulness of alerts, which in turn determines the effectiveness of the underlying CVO. The evaluation approach and results are presented in Section 6.2.

6.1.2. Ontology evaluation results and refinement

To evaluate the CVO, we first reviewed each concept of the ontology concerning the selected criteria. We conducted a focus group session with domain experts in February 2018. The focus group technique allows rich qualitative evaluation of ontologies if access to a group is feasible [77]. This technique is similar to group interviews and is driven by a facilitator with strong personal skills. While prior research has adopted focus group evaluations for ontologies [39], we selected this evaluation technique for three reasons: 1) We had access to highly qualified information security domain experts to assist us with ontology evaluation. 2) Two of the experts were experienced in ontology design and helped us moderate the discussion. 3) Finally, the focus group allowed us to avoid creating a long and complex questionnaire for validating various ontology concepts, subconcepts, and relations.

We selected 12 domain experts to validate the CVO. The participants represented diverse backgrounds, including five academicians and researchers, six information security professionals, and a security administrator working for a nonprofit organization. Participants were selected based on research expertise, domain knowledge, publications, and professional experience in information security as well as ontology design. The average age of participants was 34.2 years, and 33 % were female participants. All experts were based in the U.S. The purpose of the focus group was to validate the draft ontology and elicit expert feedback to enhance the clarity of the ontology content and structure.

We sent an email to participants the day before the focus group to explain the purpose of the session and emphasize the importance of their participation. We described the session as a "security forum" that would allow us to improve the vulnerability management process and better inform common users about vulnerabilities and countermeasures. On the day of the focus group session, we first presented and explained the purpose of the CVO. We explained each concept, subconcept, and relationships between concepts and subconcepts. Our aim was to ensure that participants understand the ontology structure and purpose well, so that they have well-developed thoughts before

suggesting any refinement. Additionally, we introduced two ontologies that are closely related to the CVO. The first ontology is proposed by Fenz and Ekelhart [28], and has some focus on vulnerabilities and controls, and the second ontology is proposed by Mittal et al. [61] and has some focus on social media intelligence. However, both of these ontologies do not represent a comprehensive set of vulnerability concepts. Our purpose in introducing these two ontologies was to facilitate the discussion on how the CVO improves the comprehensive representation of the vulnerability domain. The domain experts evaluated the content and design of the CVO on seven criteria: clarity, consistency, conciseness, expandability, correctness, minimal ontological commitment, and completeness [39]. We now discuss each criterion and the exemplary feedback we received to refine the CVO.

Clarity: Following Gruber [36], we applied three criteria to assess the ontology clarity: 1) The definitions of ontology terms should be formal and objective. 2) Ontology documentation should be in natural language. 3) The ontology terms should imply the intended meaning of the social situation and computation. We extracted terms of the CVO mainly from prior research and official sources such as the NIST, CERT/CC, and CVSS. We also documented the ontology in natural language. Finally, the feedback from domain experts helped us improve the clarity of concepts. For instance, we had *Tweet* as a key concept that had several subconcepts, including count, posting time, and creation date. During the focus group, the experts stated that *Tweet* and its subconcepts are specific to a particular platform and should be replaced by a meaningful and generic higher-level concept. Hence, the concept was renamed as *Intelligence*. We also had a subconcept labeled as *Severity Scale*. Based on the feedback, this was renamed as *Severity*.

Consistency: This criterion means that terms in the ontology should be logically consistent and unambiguous. There should be no contradiction among different concepts. Although we extracted concepts from official sources, domain experts noted some inconsistencies. For instance, we had a concept *Control* with the subconcept *Control Type*. Domain experts noted that *Control* is an ambiguous term that could have different meanings for vendors and common users. It was suggested to replace the term with *Countermeasure* with two subconcepts: *Control Type* and *Control*. Domain experts also reviewed the consistency of relationships between concepts. For instance, we have a consistent logical flow among *Vulnerability*, *Product*, and *Threat* concepts. *Vulnerability* is *exploited_by Threat*, *Vulnerability relates_to Product*, and *Threat threatens Product*.

Conciseness: Gómez-Pérez's [33] criteria of conciseness means that the ontology should not have any unnecessary or redundant concepts or attributes. We carefully reviewed and removed redundant concepts during the ontology design and evaluation. For instance, to begin with, we included a concept *IsRetweet* (i.e., whether a tweet is retweeted or not) in the CVO. However, this information can be inferred from the *Exposure* concept. Hence, we removed the *IsRetweet* concept.

Expendability/extendibility: This criterion ensures that the proposed ontology can be reused in the future by adding new concepts or applying to a specific application. The CVO was designed such that it can be extended and applied in different contexts. For instance, the *Intelligence* concept and subconcepts can be extended to other social media platforms. Furthermore, in this study, we utilized the CVO to design the CIA system. In particular, we extended the CVO to design another ontology referred to as the Cyber Intelligence Ontology (CIO) by adding a concept *Alert* and inferred properties to issue cyber alerts.

Correctness: This criterion means that the proposed ontology models correct real-world concepts [101]. The correctness of the CVO was rigorously evaluated during the focus group session. Specifically, concepts related to *Intelligence* and *Vulnerability* were revised several times to ensure the correct representation of the vulnerability management domain.

Minimal ontological commitment: Gruber [36] suggests that an ontology should have minimum claims about the modeled domain to allow flexibility and freedom in specializing the ontology. Yu et al. [101] suggest evaluating this criterion by checking if a concept can support different kinds of views. We evaluated this feature by executing different queries involving multiple views of the same concepts.

Completeness: Finally, the completeness criterion evaluates the completeness of individual definitions of the ontology [80,101]. Following Yu et al. [101], we evaluated the completeness of the CVO using competency questions involving queries and requirements that the ontology must support. The CVO can answer several important questions related to the vulnerability domain. For example, what vulnerabilities are related to a product? How many times has a vulnerability been tweeted? The questions involve concepts such as *Vulnerability*, *Product*, and *Intelligence*, and concepts are linked by relations such as *relates_to* and *refers_to*.

6.2. Evaluation of the cyber intelligence alert (CIA) system

6.2.1. Evaluation of social media intelligence extractor-tagger (SMIET)

We assessed the performance of the NLP algorithm that was used to classify tweets based on content type and author profiles. As mentioned earlier, we used the ensemble learning method that combines different classifiers to overcome precision-recall tradeoff in individual classifiers. We used 10-fold cross-validation criterion, which is critical to assess external validity for large-scale classifications [68]. Cross-validation is used for testing the performance of classifiers in the supervised-learning environment. Resulting estimates are highly reliable if the training dataset is large, and the testing dataset follows the same distribution as that of the training dataset. We are interested in estimating how well tweets are classified, given the data tagged by research assistants is used to train the classifier and subsequently applied to classify a new set of tweets. Using 10-fold cross-validation, we used 90 % of the data to build the model and 10 % of the data to test the performance of the model. For tagging a new set of data, tweets have to be cleaned, and tweet-level attributes and structure have to be extracted. Tweets are then subjected to the trained classifier to predict the categories.

We used four performance indicators to assess the accuracy of the classifier: 1) Accuracy – the total percentage of correctly classified

Table 3
Performance of the classifier using 10-fold cross-validation.

	Without Ensemble Learning (Naive Bayes)				Ensemble Learning (Naive Bayes + Support Vector Machine)			
Tweet Category	Accuracy	Precision	Recall	F Measure	Accuracy	Precision	Recall	F Measure
Alert	0.928	1	0.545	0.706	0.996	0.996	0.996	0.996
Advisory	0.940	0.976	0.381	0.548	0.994	0.989	0.999	0.994
Patch	0.972	1	0.831	0.908	0.999	0.999	1	0.999
Exploit	0.906	1	0.103	0.187	0.998	0.997	1	0.998
Root-cause	0.928	0.987	0.878	0.929	0.971	0.999	0.921	0.958
Tweet Author	**Accuracy**	**Precision**	**Recall**	**F Measure**	**Accuracy**	**Precision**	**Recall**	**F Measure**
Security Prof.	0.631	0.831	0.634	0.719	0.781	0.916	0.903	0.909
Security Firm	0.773	0.960	0.457	0.619	0.827	0.975	0.910	0.941
Other	0.691	0.747	0.463	0.572	0.790	0.801	0.925	0.859

Table 4
Comparison of the CIA system with previous work.

Results Data Sources	CIA System Twitter, CVE, NVD, vendor sites, NIST, and German IT Grundschutz manual	Mittal et al. [61] Twitter	Lippmann et al. [53] Stack Exchange, Reddit, and Twitter
Underlying Ontology	CVO and CIO	Intelligence ontology	None
Alerts Observed	71	15	Not specified
Sample Size	13,277	10,004	127
True Alerts	100 %	87 %	Not specified
False Alerts	0	13 %	10 %
Alerts Discarded	0.05 %	15 %	40 %

tweets. 2) Precision – out of the predicted ones, how many tweets are correctly categorized. 3) Recall – out of the actually categorized tweets, how many tweets are correctly predicted. 4) F-score – the harmonic mean of precision and recall and is measured as $F\_Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$. Table 3 presents the results.

Results suggest that the combined classifier achieved high performance on all measures for content categories and author types. While the ensemble approach did not achieve 100 % performance across all categories, the overall performance is very good, and comparable to those achieved by leading text mining systems [e.g., 40] as well as recent social media content classification approaches [e.g. 53]. However, in comparison to content categories, we achieved low performance for tweet authors. This is because author types are determined from Twitter user's profile descriptions, and users are usually creative in describing their online profiles. Nevertheless, our assessment results are better than those achieved by commonly used unsupervised NLP approaches. The unsupervised NLP approaches use existing databases such as WordNet or previously tagged text corpus to train the algorithm. In comparison, the supervised NLP technique used in this study relies on human coders to obtain the initial tagged data for training the algorithm. Hence, the performance of supervised NLP is better than unsupervised NLPs. However, the performance of supervised NLP depends on the quality of human coding. Future research could utilize crowdsourcing platforms such as Amazon Mechanical Turk to achieve robust tagging of data, which could further improve the performance of the algorithm [see 53].

6.2.2. Evaluation of the cyber intelligence alert (CIA) system

Finally, we evaluated the quality of alerts generated by the CIA system. As noted earlier, the quality of alerts generated partly depends on the underlying CVO. Following prior research [78], we used the following metrics to evaluate the performance of the alert system. Suppose that A denotes the total number of active alerts and I denotes the total number of inactive alerts. Then $na \in A$ is the number of correctly detected active alerts, $ni \in I$ is the number of correctly detected inactive alerts, na_i is the number of active alerts detected as inactive, and ni_a is the number of inactive alerts detected as active. For each alert, the following evaluation method is applied:

1 True Positive (TP): The number of alerts correctly classified as active. $TP = na/A$.
2 True Negative (TN): The number of alerts correctly classified as inactive. $TN = ni/I$.
3 False Positive (FP): The number of alerts incorrectly classified as active. $FP = ni_a/\mathrm{I}$.
4 False Negative (FN): The number of alerts incorrectly classified as inactive. $FN = na_i/A$.

Based on the above four rules, we derived four measures of quality [29,99]: 1) Precision is the ratio of how many alerts classified as active are correct. $Precision = TP/(TP + FP)$. 2) Recall is the number of correctly detected active alerts divided by the number of alerts. $Recall = TP/(TP + TN)$. 3) Accuracy is the number of correct detection divided by the number of total alerts. $Accuracy = (TP + TN)/(TP + TN + FP + FN)$. 4) F-measure is a measure of the test's accuracy, which represents the balance between precision and recall.

We applied the above-described approach to a random set of 125 vulnerabilities. We recruited two research assistants first to determine whether a particular vulnerability should have an associated active or inactive alert. Assistants manually analyzed related tweets and applied codified rules to annotate tweets. Next, assistants observed the system generated alerts for corresponding vulnerabilities. Results are averaged for 125 vulnerabilities. On average, the alert system shows reliable performance. Assistants agreed that out of 125 vulnerabilities, the system correctly set 71 vulnerabilities with active alerts and discarded 48 vulnerabilities that do not need to be alerted. However, consistent with prior research [61], we observed that 6 vulnerabilities were discarded because of unidentifiable characters. Overall, we achieved 95 % accuracy for all measures.

Table 4 compares results of the CIA system with those of previous work. However, it should be noted that, to the best of our knowledge, there is no prior system that integrates vulnerability information from multiple sources to alert common users. Our intention in comparing results is to merely highlight the high performance of the alert system compared to the ones that attempt to generate alerts based on Twitter data.

The first study is by Mittal et al. [61], who designed an intelligence ontology based on Twitter discourse about vulnerability information. However, the ontology does not represent the comprehensive set of vulnerability concepts from other sources [see 28]. Furthermore, the ontology does not include patch details and tweet source information. As noted earlier, these concepts are important to generate correct alerts [88]. Nevertheless, authors used 10,004 tweets to create 158 intelligence entities. Each intelligence entity provides information about vulnerabilities, including the number of tweets related to vulnerability, consequences, and product impacted. Using intelligence entities, the system issued 15 cybersecurity alerts. A small scale user study comprised of five assessors marked 13 alerts out of 15 useful. Furthermore, a significant loss of intelligence due to discarded tweets was observed. The annotators found that the classifier missed 44 relevant tweets with actionable intelligence out of randomly selected 300 tweets due to several reasons, including non-English words, wrong spellings, and unidentifiable characters.

The second study is by Lippmann et al. [55], who adopted the ensemble learning approach to develop a classifier based on cybersecurity discussions on Stack Exchange, Reddit, and Twitter. The cybersecurity discussions were related to attack methods, exploits, vulnerabilities, strategies, attack tools, defenses, and actual and potential victims. The proposed classifier is, however, not based on any underlying domain ontology. Using a small sample of 127 tweets, this study reports 10 % false alarms. Additionally, the classifier fails to detect 40 % of cybersecurity-related tweets.

Additionally, we conducted a further evaluation to determine the performance and usefulness of the CIA system. As noted earlier, a user can query the knowledge base to retrieve alert information from the CIO. We queried the knowledge base to determine the time it takes to execute different queries. Following [23], we wrote SPARQL queries

Table 5
Alert knowledge base query time.

Alert property queried	Alerts matched	Time (milliseconds)
High Severity	17	512
Patch Availability	13	473
Tweet Counter	21	570

based on several alert properties. An example of a query is shown in Fig. 7. Table 5 presents results for three properties: 1) alerts that have high severity, 2) alerts for which patch is available, and 3) alerts for which the tweet counter is greater than 10. Results suggest that the time to retrieve query results depends on the number of alerts that match the query, but not on the query itself. However, further tests can be performed with a larger set of alerts.

Next, following [61] and [20], we conducted a user-survey to assess the usefulness of alerts. We selected 22 participants using simple random sampling. The participants were pursuing a graduate degree in IT at a large US-based university. We first explained the purpose of the system and a sample query followed by a demonstration of query execution. We also explained results of the sample query. Next, each participant was provided SPARQL queries and asked to execute three to five queries and understand the results. Finally, participants were asked to take a survey to assess the usefulness of alerts and query engine. Fig. 8 presents the questionnaire. Each question was measured on a five-point Likert scale with responses ranging from "strongly agree" to "strongly disagree." The survey was hosted online using Qualtrics surveying software. Results are presented in Table 6. Overall, 82 % of users were satisfied with results of the alert system.

7. Discussion and conclusion

In this study, we posit that the prevention of cyber exploits will require the integration of vulnerability information from multiple official sources with social media intelligence. The information can be presented in an ontological format that will allow security analysts to assess the severity of vulnerabilities and issue cyber alerts about vulnerabilities and countermeasures to common users. To that end, we proposed a conceptual ontology for vulnerability management, referred to as the CVO. We utilized the CVO to build the CIA system that issues cyber alerts about impending vulnerabilities and countermeasures. In doing so, this study makes several contributions to research and practice.

In design science research, the IT artifact is regarded as a vehicle for research and practical impacts [10]. In this study, we build two IT artifacts. Hence, the contribution to research is two-fold. First, this study contributes to research by providing the conceptual cybersecurity vulnerability ontology. While prior research has proposed several ontologies for vulnerability management, the ontology structure is informed by generic information security concepts [e.g., 1, 26, 72, 90, 95]. Additionally, prior studies do not evaluate the ontology design and content for adequate domain representation. In this study, we integrated vulnerability information from multiple sources and presented it in an ontological format. Specifically, we integrated the information security concepts specified in the NIST handbook with CERT/CC specifications. The CERT/CC is responsible for the vulnerability coordination and disclosure process and provides several metrics to characterize vulnerabilities. We also included vulnerability characteristics provided by the CVSS framework, which is an industry standard to assess the severity of vulnerabilities.

Researchers and hackers have also begun to use social media as another pathway to disclose or share vulnerability information [61,83]. Research suggests that social media users often disclose the vulnerability information ahead of official sources, and vulnerabilities that receive high social media attention are often exploited [46,70,83]. Hence, it is important to integrate information from social media to adequately detect and prevent possible exploits. Research also suggests that while major software vendors have developed their scoring systems for assessing the risk of vulnerabilities, both CVSS- based approach, as well as proprietary vendor approaches (such as of Microsoft), have a very high false-positive rate of detecting exploitability [see 100]. Authors make a call to identify new metrics, specifically external factors that could improve exploit predictions. Hence, integrating vulnerability information from social media with other sources is important for vulnerability management.

Recently, there has been some focus on the ontological representation of social media intelligence for vulnerability management [61]. However, the ontology does not represent the comprehensive set of vulnerability concepts [see 28]. Additionally, patch-related details are not considered, which is important to identify vulnerabilities that pose high exploitability risk. Moreover, tweet author information is not accounted for, which determines the trustworthiness of vulnerability

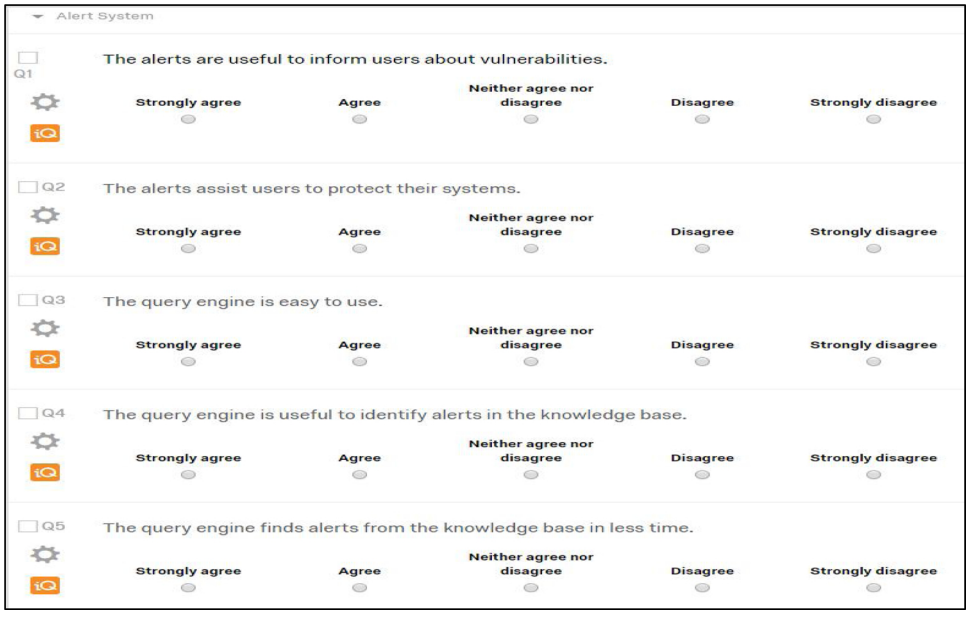

Fig. 8. Questionnaire for assessing user satisfaction.

Table 6
User satisfaction results.

Question item	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree	Total	Satisfaction (%)
Q(1)	12	8	2	0	0	22	91
Q(2)	8	7	4	3	0	22	68
Q(3)	9	7	4	2	0	22	73
Q(4)	10	9	3	0	0	22	86
Q(5)	14	6	2	0	0	22	91
Average	10.6	7.4	3	1	0	22	82

information [88] and influences the subsequent diffusion of vulnerability information on social media [83]. To fill the research gap, we enhanced the CVO with the *Intelligence* concept, which represents vulnerability information extracted from Twitter [61,83]. The four sub-concepts – *Content Type*, *Exposure, Technical Details, and Source* – characterize intelligence based on tweet text, tweet author details, as well as technical attributes of tweets. Furthermore, the *Intelligence* concept is related to the *Vulnerability* concept, which enhances the reasoning ability of the CVO related to other concepts such as *Threat*, *Product,* and *Countermeasure*. The content and design of the ontology are rigorously evaluated by domain experts in a focus-group setting using a criteria-based evaluation approach.

Second, this study contributes to research by designing the CIA system. The CIA system generates cyber alerts to inform common users about vulnerabilities and potential countermeasures for preventing exploits. While recent research has proposed a cybersecurity alert system, alerts are based only on Twitter data [61]. The vulnerability information available from other official sources such as the CERT/CC and vendor sites is not considered. Another system that integrates vulnerability information from multiple social media platforms is not based on any underlying domain ontology [53]. In designing the CIA system, we utilized and extended the CVO with concepts and properties required to issue cyber alerts. The emergent cyber intelligence ontology (CIO) provides a knowledge base to centralize the vulnerability information as well as the related alerts. We instantiated the CIO with the data from multiple sources, including Twitter, CVE, NVD, and vendor sites. The CIA system then reasons over the knowledge base to determine if an alert should be raised for a vulnerability. We used several state-of-the-art design approaches. Specifically, we used the ensemble learning approach, which is a machine learning technique to classify tweets based on content categories and author profiles. Results suggest that the combined classifier performed better than individual approaches. Finally, we rigorously evaluated the CIA system. Compared to previous approaches, results suggest that alerts are highly accurate (see Table 4). Additionally, the CIA system discards considerably low number of alerts due to unidentifiable characters in tweets. The performance and usefulness of the system are also high.

At the practical level, it is expected that the CVO will be used as a general vocabulary of the vulnerability management domain and might interest researchers to further explore the representation of the domain. Additionally, the CIA system might interest security analysts and vendors. The existing vulnerability prioritization frameworks are mostly based on the CVSS specifications and do not consider social media exposure. The CIA system and the underlying ontologies (that is, CVO and CIO) integrate information from multiple sources, which could help security analysts make better vulnerability prioritization decisions. Vendors could also share the alert knowledge base by integrating it with other systems through semantic web interfaces. Furthermore, mailing lists such as BugTraq alert subscribers about vulnerabilities who could then install countermeasures or develop patches; however, subscribers are usually security professionals, vendors, or hackers [67]. Likewise, other market-based mailing lists such as TippingPoint issue advisories to their subscribers. However, advisories are limited to general countermeasures and do not include vulnerability details. The proposed CIA system is designed for the benefit of common users who may not be aware of these sources, or the information may not be comprehensible to them. Official agencies such as the CERT/CC and security analysts may find the CIA system beneficial to issue cyber alerts about impending vulnerabilities and countermeasures, and thus help to protect common users.

This study has some limitations, which gives directions for future research. The *Intelligence* concept is primarily based on Twitter data. However, the vulnerability information is also shared through other social media platforms, wikis, blogs, hacker forums, and dark web. Future research should explore other sources of vulnerability information and integrate concepts in the CVO for comprehensive vulnerability management. Furthermore, we inferred the influence of intelligence *source* based on whether a tweet author is a security professional, security firm, or other. However, future research can utilize quantitative measures such as user mentions, activities, and followers as indicators of source influence[15] . Additionally, the *vulnerability* concept in the CVO is limited to US-CERT/CC specification. However, there are several other National CERTs around the world that are independent entities and handle vulnerabilities in their respective countries. Future research should focus on integrating the vulnerability information between US-CERT and other National CERTs to provide a more global perspective of vulnerability management.

Another limitation is that the CVO does not account for proprietary vulnerability scoring systems used by software vendors [100]. Future research could extend the proposed CVO with other scoring systems, which could further enhance decision-making. Furthermore, the CIA system is not entirely automated. For instance, we extracted the Twitter data using Python/R scripts, cleaned, and tagged the data manually before importing it into the CVO using the Cellfie plugin. As data on social media is generated at high speed and volume, and the vulnerability information is often time-sensitive, manual processing might introduce delays in issuing cyber alerts. Hence, future research should explore the automation of these tasks. Additionally, our purpose was to develop a proof of concept for generating cyber alerts. Hence, we did not code a comprehensive set of rules for generating alerts.

The evaluation approach used in this study has some limitations. We evaluated the CVO in a focus group comprised of domain experts. However, the manual evaluation of an ontology can suffer from evaluators' lack of knowledge, misinterpretations, cognitive overload, and human bias. Furthermore, we queried the ontology to determine the time it takes to execute different queries. While our results suggest a good performance, the lack of prior research prevented us from comparing the performance of the system. We, however, applied the CVO to build the CIA system and evaluated the performance of the CIA system and underlying components. Based on evaluation results of the SMIET classifier, alert quality, and user survey, we are confident of the correctness and usefulness of the CIA system. Nevertheless, consistent with prior research, we call for developing automated ontology evaluation methods and strategies [17,89]. Another limitation relates to the Twitter API that provided a random 1 % of data. While the extended period of data collection ensured decent coverage, more data can

[15] https://www.entrepreneur.com/article/313320 (Accessed on May 10, 2020)

further improve the performance of the system. Finally, while the CIA system represents one of the applications of the CVO, future research could explore other useful applications of the ontology.

Author statement

Romilla Syed: Conceptualization, Data collection and analysis, Design, Evaluation, Writing and revising the manuscript.

Acknowledgment

I highly appreciate the generous feedback and helpful suggestions, especially from associate editor and reviewers. I am also grateful for the excellent research assistance from Haonan Zhong.

References

[1] J. An Wang, M.M. Guo, J. Camargo, An ontological approach to computer system security, Inf. Secur. J. A Glob. Perspect. 19 (2) (2010) 61–73.
[2] A. Anaby-Tavor, D. Amid, A. Fisher, A. Bercovici, H. Ossher, M. Callery, M. Desmond, S. Krasikov, I. Simmonds, Insights into enterprise conceptual modeling, Data Knowl. Eng. 69 (12) (2010) 1302–1318.
[3] K. Arbanas, M. Čubrilo, Ontology in information security, J. Inf. Organizational Sci. 39 (2) (2015) 107–136.
[4] A. Arora, R. Krishnan, R. Telang, Y. Yang, An empirical analysis of software vendors' patch release behavior: impact of vulnerability disclosure, Inf. Syst. Res. 21 (1) (2010) 115–132.
[5] A. Arora, A. Nandkumar, R. Telang, Does information security attack frequency increase with vulnerability disclosure? An empirical analysis, Inf. Syst. Front. 8 (5) (2006) 350–362.
[6] A. Arora, R. Telang, H. Xu, Optimal policy for software vulnerability disclosure, Manage. Sci. 54 (4) (2008) 642–656.
[7] A. Avižienis, J.-C. Laprie, B. Randell, Dependability and its threats: a taxonomy, in: R. Jacquart (Ed.), Building the Information Society, Springer, Boston, MA, 2004, pp. 91–120.
[8] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, D. Nardi, The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, Cambridge, UK, 2003.
[9] S. Babar, P. Mahalle, A. Stango, N. Prasad, R. Prasad, Proposed security model and threat taxonomy for the internet of things (IoT), Proceedings of the International Conference on Network Security and Applications (2010) 420–429.
[10] R. Baskerville, A. Baiyere, S. Gregor, A. Hevner, M. Rossi, Design science research contributions: finding a balance between artifact and theory, J. Assoc. Inf. Syst. 19 (5) (2018) 358–376.
[11] V. Benjamin, W. Li, T. Holt, H. Chen, Exploring threats and vulnerabilities in hacker web: forums, IRC and carding shops, Proceedings of the IEEE International Conference on Intelligence and Security Informatics, Baltimore, MD, 2015, pp. 85–90.
[12] S. Bird, E. Klein, E. Loper, Natural Language Processing With Python: analyzing Text With the Natural Language Toolkit, O'Reilly Media, Inc., Sebastopol, CA, 2009.
[13] H. Birkholz, I. Sieverdingbeck, K. Sohr, C. Bormann, IO: an interconnected asset ontology in support of risk management processes, Proceedings of the Seventh International Conference on Availability, Reliability and Security, Prague, Czech Republic, 2012, pp. 534–541.
[14] M.A. Bishop, The Art and Science of Computer Security, Addison-Wesley, Boston, MA, 2002.
[15] C. Blackwell, A security ontology for incident analysis, Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, Oak Ridge, TN, 2010, p. 46.
[16] P. Bowen, J. Hash, M. Wilson, Information Security Handbook: a Guide for Managers, National Institute of Standards and Technology (NIST), 2007.
[17] J. Brank, M. Grobelnik, D. Mladenić, A survey of ontology evaluation techniques, Proceedings of the Conference on Data Mining and Data Warehouses, Ljubljana, Slovenia, 2005.
[18] C. Brewster, H. Alani, S. Dasmahapatra, Y. Wilks, Data driven ontology evaluation, Proceedings of the International Conference on Language Resources and Evaluation, Lisbon, PortUgal, 2004.
[19] M. Bunge, Treatise on Basic Philosophy: Ontology I: The Furniture of the World, Springer Science & Business Media, Boston, MA, 1977.
[20] Y.-J. Chen, H.-C. Chu, Y.-M. Chen, C.-Y. Chao, Adapting domain ontology for personalized knowledge search and recommendation, Inf. Manag. 50 (6) (2013) 285–303.
[21] P.C.-H. Cheng, R.K. Lowe, M. Scaife, Cognitive science approaches to understanding diagrammatic representations, in: A.F. Blackwell (Ed.), Thinking With Diagrams, Springer, Dordrecht, Netherlands, 2001, pp. 79–94.
[22] J.A. Clark, S. Stepney, H. Chivers, Breaking the model: finalisation and a taxonomy of security attacks, Electron. Notes Theor. Comput. Sci. 137 (2) (2005) 225–242.
[23] J.E.L. De Vergara, V.A. Villagrá, P. Holgado, E. De Frutos, I. Sanz, A semantic web approach to share alerts among security information management systems, in: C. Serrão, V. Aguilera, D. Cerullo (Eds.), Web Application Security, Springer, Berlin, Heidelberg, 2010, pp. 27–38.
[24] T.G. Dietterich, Ensemble methods in machine learning, Proceedings of the International Workshop on Multiple Classifier Systems, Günzburg, Germany, 2000, pp. 1–15.
[25] F.N. Do Amaral, C. Bazilio, G.M.H. Da Silva, A. Rademaker, E.H. Haeusler, An ontology-based approach to the formalization of information security policies, Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Workshops, Washington, DC, 2006, p. 1.
[26] G. Elahi, E. Yu, N. Zannone, A modeling ontology for integrating vulnerabilities into security requirements conceptual foundations, Proceedings of the International Conference on Conceptual Modeling, Gramado, Brazil, 2009, pp. 99–114.
[27] A. Evesti, S. Pantsar-Syväniemi, Towards micro architecture for security adaptation, Proceedings of the Fourth European Conference on Software Architecture: Companion Volume (2010) 181–188.
[28] S. Fenz, A. Ekelhart, Formalizing information security knowledge, Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, Sydney, NSW, Australia, 2009, pp. 183–194.
[29] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognit. Lett. 30 (1) (2009) 27–38.
[30] J. Friedman, D.V. Hoffman, Protecting data on mobile devices: a taxonomy of security threats to mobile computing and review of applicable defenses, Inf. Knowl. Syst. Manage. 7 (12) (2008) 159–180.
[31] A. Gemino, Y. Wand, Evaluating modeling techniques based on models of learning, Commun. ACM 46 (10) (2003) 79–84.
[32] Federal Office for Information Security, German IT Grundschutz Manual, 2004 Available at: https://www.bsi.bund.de/EN/Topics/ITGrundschutz/ITGrundschutzHome/itgrundschutzhome_node.html.
[33] A. Gómez-Pérez, Evaluation of ontologies, Int. J. Intell. Syst. 16 (3) (2001) 391–409.
[34] S. Gregor, A.R. Hevner, Positioning and presenting design science research for maximum impact, Mis Q. 37 (2) (2013) 337–355.
[35] M. Grobler, J.J. van Vuuren, L. Leenen, Implementation of a cyber security policy in south africa: reflection on progress and the way forward, Proceedings of the IFIP International Conference on Human Choice and Computers (2012) 215–225.
[36] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum. Stud. 43 (5–6) (1995) 907–928.
[37] M. Grüninger, M.S. Fox, Methodology for the design and evaluation of ontologies, Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995.
[38] N. Guarino, Formal ontology in information systems, Proceedings of the International Conference on Formal Ontology in Information Systems, Trento, Italy, 1998, pp. 3–15.
[39] P.D. Haghighi, F. Burstein, A. Zaslavsky, P. Arbon, Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings, Decis. Support Syst. 54 (2) (2013) 1192–1204.
[40] M. Hassan, B. Vikas, H. Fiorletta, Accurate information extraction for quantitaive financial events, Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, Scotland, UK, 2011, pp. 2497–2500.
[41] A. Herzog, N. Shahmehri, C. Duma, An ontology of information security, Int. J. Inf. Secur. Priv. 1 (4) (2007) 1–23.
[42] C.W. Holsapple, K.D. Joshi, A collaborative approach to ontology design, Commun. ACM 45 (2) (2002) 42–47.
[43] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, Technical Report, W3C (2004).
[44] A.D. Householder, G. Wassermann, A. Manion, C. King, The CERT Guide to Coordinated Vulnerability Disclosure, Software Engineering Institute, Carnegie Mellon University, 2017.
[45] J. Hu, P. Bertok, Z. Tari, Taxonomy and framework for integrating dependability and security, in: Y. Qian, D. Tipper, P. Krishnamurthy, J. Joshi (Eds.), Dependability and Security in Networked Systems, Elsevier, Burlington, MA, 2008, pp. 149–170.
[46] H. Joh, Y.K. Malaiya, A framework for software security risk evaluation using the vulnerability lifecycle and cvss metrics, Proceedings of the International Workshop on Risk and Trust in Extended Enterprises, San Jose, CA, 2010, pp. 430–434.
[47] D. Jurafsky, J.H. Martin, Speech and Language Processing, Pearson, Upper Saddle River, New Jersey, 2014.
[48] D. Jutla, L. Xu, Privacy agents and ontology for the semantic web, Proceedings of the Americas Conference on Information Systems, New York, 2004, p. 210.
[49] M. Kanakaraj, R.M.R. Guddeti, Performance analysis of ensemble methods on twitter sentiment analysis using NLP techniques, Proceedings of the 2015 IEEE International Conference on Semantic Computing (2015) 169–170.
[50] K. Kannan, R. Telang, Market for software vulnerabilities? Think again, Manage. Sci. 51 (5) (2005) 726–740.
[51] A.D. Khairkar, D.D. Kshirsagar, S. Kumar, Ontology for detection of web attacks, Proceedings of the International Conference on Communication Systems and Network Technologies, Gwalior, India, 2013, pp. 612–615.
[52] J.H. Larkin, H.A. Simon, Why a diagram is (sometimes) worth ten thousand words, Cogn. Sci. 11 (1) (1987) 65–100.
[53] D. Lee, K. Hosanagar, H.S. Nair, Advertising content and consumer engagement on social media: evidence from Facebook, Manage. Sci. 64 (11) (2018) 5105–5131.
[54] P. Li, H.R. Rao, An examination of private intermediaries' roles in software vulnerabilities disclosure, Inf. Syst. Front. 9 (5) (2007) 531–539.

[55] R.P. Lippmann, J.P. Campbell, D.J. Weller-Fahy, A.C. Mensch, W.M. Campbell, Finding Malicious Cyber Discussions in Social Media, MIT Lincoln Laboratory Lexington, United States, 2016.
[56] F.-H. Liu, W.-T. Lee, Constructing enterprise information network security risk management mechanism by ontology, Tamkang Journal of Science and Engineering 13 (1) (2010) 79r87.
[57] A. Lozano-Tello, A. Gómez-Pérez, Ontometric: A method to choose the appropriate ontology, J. Database Manag. 2 (15) (2004) 1–18.
[58] A. Maedche, S. Staab, Measuring similarity between ontologies, Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (2002) 251–263.
[59] S.T. March, G.F. Smith, Design and natural science research on information technology, Decis. Support Syst. 15 (4) (1995) 251–266.
[60] A. Martimiano, E. Moreira, An owl-based security incident ontology, Proceedings of the Eighth International Protege Conference (2005) 43–44.
[61] S. Mittal, P.K. Das, V. Mulwad, A. Joshi, T. Finin, Cybertwitter: using twitter to generate alerts for cybersecurity threats and vulnerabilities, Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, San Francisco, CA, 2016, pp. 860–867.
[62] D. Moody, The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering, Ieee Trans. Softw. Eng. 35 (6) (2009) 756–779.
[63] N.F. Noy, D.L. McGuinness, Ontology Development 101: a Guide to Creating Your First Ontology, Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, Stanford, CA (2001).
[64] E. Paintsil, L. Fritsch, A taxonomy of privacy and security risks contributing factors, Proceedings of the IFIP PrimeLife International Summer School on Privacy and Identity Management for Life (2010) 52–63.
[65] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, J. Manag. Inf. Syst. 24 (3) (2007) 45–77.
[66] S. Ransbotham, S. Mitra, The impact of immediate disclosure on attack diffusion and volume, in: B. Schneier (Ed.), Economics of Information Security and Privacy III, Springer, New York, NY, 2013, pp. 1–12.
[67] S. Ransbotham, S. Mitra, J. Ramsey, Are markets for vulnerabilities effective? Mis Q. 36 (1) (2012) 43–64.
[68] P. Refaeilzadeh, L. Tang, H. Liu, Cross-validation, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, Springer, 2009, pp. 532–538.
[69] Fd.F. Rosa, R. Bonacin, M. Jino, The security assessment domain: a survey of taxonomies and ontologies, Renato Archer Information Technology Center (CTI), Campinas/SP, Brazil, 2017.
[70] C. Sabottke, O. Suciu, T. Dumitras, Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits, Proceedings of the USENIX Security Symposium, Austin, TX, 2015, pp. 1041–1056.
[71] P. Salini, S. Kanmani, A knowledge-oriented approach to security requirements engineering for E-voting system, Int. J. Comput. Appl. 49 (11) (2012) 21–25.
[72] P. Salini, J. Shenbagam, Prediction and classification of web application attacks using vulnerability ontology, Int. J. Comput. Appl. 116 (21) (2015) 42–47.
[73] C. Sauerwein, C. Sillaber, M.M. Huber, A. Mussmann, R. Breu, The tweet advantage: an empirical analysis of 0-day vulnerability information shared on twitter, Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection (2018) 201–215.
[74] R. Savola, Towards a security metrics taxonomy for the information and communication technology industry, Proceedings of the International Conference on Software Engineering Advances (2007) 60.
[75] R. Sen, G.R. Heim, Managing enterprise risks of technological systems: an exploratory empirical analysis of vulnerability characteristics as drivers of exploit publication, Decis. Sci. 47 (6) (2016) 1073–1102.
[76] R. Shirey, RFC 2828 - Internet Security Glossary, (2000).
[77] J. Sim, Collecting and analysing qualitative data: issues raised by the focus group, J. Adv. Nurs. 28 (2) (1998) 345–352.
[78] S. Smadi, N. Aslam, L. Zhang, Detection of online phishing email using dynamic evolving neural network based on reinforcement learning, Decis. Support Syst. 107 (2018) 88–102.
[79] A. Souag, C. Salinesi, R. Mazo, I. Comyn-Wattiau, A security ontology for security requirements elicitation, Proceedings of the International Symposium on Engineering Secure Software and Systems (2015) 157–177.
[80] S. Staab, R. Studer, H.-P. Schnurr, Y. Sure, Knowledge processes and ontologies, IEEE Intell. Syst. 16 (1) (2001) 26–34.
[81] S. Strohmeier, F. Röhrs, Conceptual modeling in human resource management: a design research approach, Ais Trans. Hum. Interact. 9 (1) (2017) 34–58.
[82] V. Sugumaran, V.C. Storey, Ontologies for conceptual modeling: their creation, use, and management, Data Knowl. Eng. 42 (3) (2002) 251–271.
[83] R. Syed, M. Rahafrooz, J.M. Keisler, What it takes to get retweeted: an analysis of software vulnerability messages, Comput. Human Behav. 80 (2018) 207–215.
[84] Z. Syed, A. Padia, T. Finin, L. Mathews, A. Joshi, UCO: a unified cybersecurity ontology, Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security (2016) 14–21.
[85] R. Tairas, M. Mernik, J. Gray, Using ontologies in the domain analysis of domain-specific languages, Proceedings of the International Conference on Model Driven Engineering Languages and Systems, Toulouse, France, 2008, pp. 332–342.
[86] R. Telang, S. Wattal, An empirical analysis of the impact of software vulnerability announcements on firm stock price, Ieee Trans. Softw. Eng. 8 (2005) 544–557.
[87] O. Temizkan, R.L. Kumar, S. Park, C. Subramaniam, Patch release behaviors of software vendors in response to vulnerabilities: an empirical analysis, J. Manag. Inf. Syst. 28 (4) (2012) 305–338.
[88] S. Trabelsi, H. Plate, A. Abida, M.M.B. Aoun, A. Zouaoui, C. Missaoui, S. Gharbi, A. Ayari, Mining social networks for software vulnerabilities monitoring, Proceedings of the 7th International Conference on New Technologies, Mobility and Security, Paris, France, 2015, pp. 1–7.
[89] J. Venable, J. Pries-Heje, R. Baskerville, FEDS: a framework for evaluation in design science research, Eur. J. Inf. Syst. 25 (1) (2016) 77–89.
[90] A. Vorobiev, N. Bekmamedova, An ontology-driven approach applied to information security, Journal of Research and Practice in Information Technology 42 (1) (2010) 61–76.
[91] A. Vorobiev, J. Han, N. Bekmamedova, An ontology framework for managing security attacks and defences in component based software systems, Proceedings of the 19th Australian Conference on Software Engineering, Perth, Australia, 2008, pp. 552–561.
[92] Y. Wand, D.E. Monarchi, J. Parsons, C.C. Woo, Theoretical foundations for conceptual modelling in information systems development, Decis. Support Syst. 15 (4) (1995) 285–304.
[93] Y. Wand, R. Weber, Research commentary: information systems and conceptual modeling—a research agenda, Inf. Syst. Res. 13 (4) (2002) 363–376.
[94] H. Wang, C. Wang, Taxonomy of security considerations and software quality, Commun. ACM 46 (6) (2003) 75–78.
[95] J.A. Wang, M. Guo, Security data mining in an ontology for vulnerability management, Proceedings of the International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Washington, DC, 2009, pp. 597–603.
[96] J.A. Wang, M.M. Guo, J. Camargo, An ontological approach to computer system security, Inf. Secur. J. A Glob. Perspect. 19 (2) (2010) 61–73.
[97] G. Wangen, E. Snekkenes, A taxonomy of challenges in information security risk management, Proceedings of the Norwegian Information Security Conference/Norsk Informasjonssikkerhetskonferanse-NISK, Akademika Forlag, 2013.
[98] R. Weber, Conceptual modelling and ontology: possibilities and pitfalls, J. Database Manag. 14 (3) (2003) 1–20.
[99] T.D. Wickens, Elementary Signal Detection Theory, Oxford University Press, New York, USA, 2002.
[100] A.A. Younis, Y.K. Malaiya, Comparing and evaluating CVSS base metrics and microsoft rating system, Proceedings of the 2015 IEEE International Conference on Software Quality, Reliability and Security, 2015, pp. 252–261.
[101] J. Yu, J.A. Thom, A. Tam, Evaluating ontology criteria for requirements in a geographic travel domain, Proceedings of the OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Agia Napa, Cyprus, 2005, pp. 1517–1534.
[102] J. Yu, J.A. Thom, A. Tam, Ontology evaluation using wikipedia categories for browsing, Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (2007) 223–232.
[103] H. Zhu, Q. Huo, Developing software testing ontology in UML for a software growth environment of web-based applications, Software Evolution with UML and XML, IGI Global (2005) 263–295.

Romilla Syed is an assistant professor of Management Science and Information Systems at the University of Massachusetts, Boston, MA, USA. She received a Ph.D. in information systems from Virginia Commonwealth University. Her research interests include behavioral and organizational security and privacy, computer-mediated communication, and decision analysis. Her work has been published in various journals including Decision Sciences, Journal of Strategic Information Systems, Information and Management, Decision Support Systems, Computers and Security, and Computer in Human Behavior, and flagship conferences such as International Conference on Information Systems, European Conference on Information Systems, and Academy of Management, among others.