Software Engineering Institute
Carnegie Mellon University

An Insider Threat Indicator Ontology

Daniel L. Costa
Michael J. Albrethsen
Matthew L. Collins
Samuel J. Perl
George J. Silowash
Derrick L. Spooner

May 2016

TECHNICAL REPORT
CMU/SEI-2016-TR-007

CERT® Division

http://www.sei.cmu.edu

This material is based upon work funded and supported by Defense Advanced Research Projects Agency (DARPA) and Federal Bureau of Investigation (FBI) under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Defense Advanced Research Projects Agency (DARPA) and Federal Bureau of Investigation (FBI) or the United States Department of Defense.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

This report was prepared for the
SEI Administrative Agent
AFLCMC/PZM
20 Schilling Circle, Bldg. 1305, 3rd floor
Hanscom AFB, MA 01731-2125

DM-0002168

Table of Contents

List of Figures

List of Tables

Acknowledgments

The authors gratefully acknowledge support for this report from the Defense Advanced Research Projects Agency (DARPA) and the Federal Bureau of Investigation (FBI). The authors also wish to acknowledge contributions to this report from our former colleagues Alex Nicoll, David Mundie, and John McCloud.

The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Abstract

The insider threat community currently lacks a standardized method of expression for indicators of potential malicious insider activity. We believe that communicating potential indicators of malicious insider activity in a consistent and commonly accepted language will allow insider threat programs to implement more effective controls through an increase in collaboration and information sharing with other insider threat teams. In this report, we present an ontology for insider threat indicators. We make the case for using an ontology to fill the stated gap in the insider threat community. We also describe the semi-automated, data-driven development of the ontology, as well as the process by which the ontology was validated. In the appendices, we provide the ontology's user's manual and technical specification.

1 Introduction

This report documents the initial design and implementation of an insider threat indicator ontology. First we present a brief overview of the domain of insider threat and make a case for the need for an ontology in this domain. Next we provide a foundational review of the structure and applications of ontologies and the challenges associated with their development. We then detail our approach to the ontology development process, enumerate our goals and use cases, and describe how we addressed the challenges mentioned previously. Next we introduce our method for using automated text processing techniques to facilitate the selection of the concepts and relationships to include in our ontology. Finally we present our ontology, discuss its design, implementation, and validation, and identify the next steps in the development process. The user's manual for the ontology is provided in Appendix A. The ontology's technical specification is provided in Appendix B.

1.1 Background on Insider Threat Detection

The CERT® Division of Carnegie Mellon University's Software Engineering Institute defines a malicious insider as "a current or former employee, contractor, or business partner who meets the following criteria:

- has or had authorized access to an organization's network, system, or data
- has intentionally exceeded or intentionally used that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems" [1]

"Insider threats are influenced by a combination of technical, behavioral, and organizational issues and must be addressed by policies, procedures, and technologies" [1]. Organizations can use existing technologies in new ways to prevent, detect, and respond to malicious insider activity, but organizations need to design their infrastructure with the malicious insider in mind. For example, intrusion detection systems (IDSs) should be placed in front of key servers and enclave ingress/egress points. When the IDS detects suspicious insider activity, it could send alerts to a security information and event management (SIEM) system. This type of alerting can occur in near-real time and allow an organization to respond appropriately. Existing log collection and analysis practices can also be applied to insider threat detection. Many of the same observable technical behaviors can be detected for both insider and external threats.

Once suspected malicious activity has been identified, organizations will often perform forensic investigations of affected assets. A forensic examination of machines involved in malicious insider activity allows an organization to assess the damage, identify other areas to examine, and implement strategies to mitigate the risk of similar incidents in the future. Forensic examination can be thought of as a type of detection and response activity.

® CERT is a registered mark owned by Carnegie Mellon University.

Many other approaches have been proposed to identify potential insider threats [2-13]. Through extensive analysis of these existing approaches, we have concluded that it is currently impossible to cost-effectively share and communicate indicators of insider threat. The fragmentation of data in the insider threat domain has created the need for a well-defined and agreed-on vocabulary.

1.2 The Need for an Ontology

An ontology provides a "coherent set of representational terms, together with textual and formal definitions, that embody a set of representational design choices" [14]. An ontology of indicators of insider threat would provide a common language with which to represent and share knowledge. This ontology could be used to consistently model indicators of insider threat across organizations.

All entity and relationship data models, including semantic data models, have their limitations [15]. Models are extremely formal by design and can encounter problems when representing the variety of actions involved in an actual case. In addition, the data on cases of insider threat is often gathered from legal judgments and outcomes, whose documentation is itself highly variable. As a result, insider threat domain experts tend to rely on natural language to document their cases and findings, or they only briefly summarize the events. Though natural language is more expressive than a model, we believe the insider threat domain will benefit from the development of an ontology. Our interest in building an ontology, developed from our observations of the field today, is driven by the following factors:

- We expect rapid growth in the data being collected and shared by organizations, specifically about insider threats. Some organizations have already stated that overcoming this challenge is one of their top priorities, and we have begun seeing anecdotal evidence that other organizations are working toward this goal.
- The insider threat research community lacks a defined, formal model that is machine readable, human understandable, and transferrable with limited sharing barriers for use by the community. Starting a model of this kind, using the data we have already collected, could accelerate this process within the community, as has been done in other fields [16, 17].
- We are willing to accept some loss of descriptive power for individual cases, provided we can analyze large populations of cases by computation. We expect insider threat teams (both in research and in operations) to be asked to detect insider threat activity by analyzing a growing quantity of data from new sources, but in a limited amount of time.
- It will be easier to create an ontology for the insider threat domain and, most importantly, easier for our community to collectively curate it than to use existing alternative technologies.

2 Background

Ontologies are a formal mechanism for expressing a set of concepts and their interrelationships. They also contain assertions about the individuals or instances of things that are known to exist within a knowledge base. Certain types of ontologies also facilitate the sharing and use of the information contained in a knowledge base.

2.1 Ontology Components

Many formal languages exist for constructing ontologies. Typically, ontologies comprise the same foundational components regardless of the ontology language used to develop them. This section presents brief descriptions of these common components.

2.1.1 Classes

Classes represent the concepts of a domain and provide a mechanism for specifying logical groupings of concepts [18]. Classes can have subclasses and be subclasses of other classes to develop a hierarchical "is-a" arrangement of concepts. For example, consider an ontology with the classes *Person* and *Man*. The *Man* class can be defined as a subclass of *Person*, which captures the fact that all men are people.

2.1.2 Relations

Relations define how ontology components can interact with one another. Relations can be used to describe the domain-applicable relationships between classes [19]. For example, if we have an ontology with a *Car* class, we can define an *ownsCar* relation to capture the concept of a person owning a car. Typically, relations are not applicable to all classes in an ontology. For example, if our ontology also contained a *Book* class, it would not make sense to make the *ownsCar* relation available to the *Book* class. In most ontology languages, the valid components for a relation can be restricted by fully defining which classes are disjoint and specifying a domain (which defines the class of the subject of the relationship) and range (which defines the class of the object of the relationship) for the relation.

2.1.3 Attributes

Attributes represent properties of classes. Attributes provide the ability to assign specific values to instances of classes. For example, we can add the *Age* attribute to our simple ontology and use it to assign specific ages to instances of the *Person* class.

2.1.4 Individuals

Individuals are specific instances of classes, relations, and attributes. In our toy ontology, we defined a *Man* class. We can create an individual named *Bob* that is an instance of the *Man* class. Additionally, we can create an instance of the *Car* class named *1966 Ford Mustang* and an instance of the *ownsCar* relation that relates the individuals *Bob* and *1966 Ford Mustang*.

2.1.5 Knowledge

2.1.5.1 Terminology Knowledge

Terminology knowledge (also referred to as the TBox) is the part of a knowledge base that specifies the vocabulary of terms that exist within a knowledge domain. Because ontologies are a formal specification, the terms within them have specific definitions. The creation of classes (or concepts) and the hierarchy of those classes, such as *a Man is_a Human*, and the rules to specify class membership are examples of TBox knowledge. The TBox typically contains both the names of things and the constraints that form them [20]. Said another way, names are typically names of concepts, and constraints are the rules that apply to those concepts. A classic example of a constraint is to name a class *Human Child* and then constrain it with a statement such as *only Humans can have Human Children* [21]. Both types of knowledge are examples of TBox knowledge. Ontologies are not required to express all of the formal constraints on a class in the domain [22].

2.1.5.2 Assertional Knowledge

Assertional knowledge (also referred to as the ABox) is the part of the knowledge base that contains knowledge about the individuals that exist within a given domain. The names of the individuals or instances represented in the ontology are examples of ABox knowledge. For example, *Bob is_a Person* declares membership in the *Person* class for the individual *Bob*.

Our ontology contains both TBox and ABox knowledge. We expect our knowledge in the ABox to increase as we apply the ontology to instances of insider threat cases from our data set and from new data sets. The TBox will likely change as well, and a person or team will need to curate those changes.

2.1.5.3 TBox and ABox Examples

Figure 1 shows TBox and ABox examples.

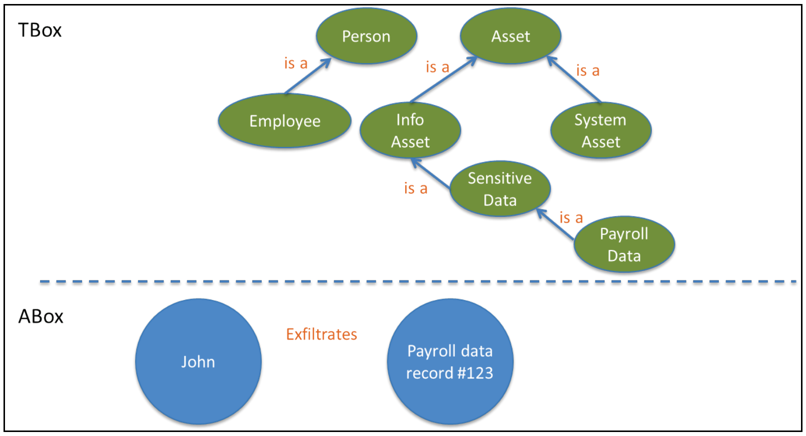


Figure 1: Sample TBox and ABox

Figure 2 shows the following assertions:

- Individual *John* is a member of class *Employee.*
- Individual *Payroll data record #123* is a member of class *Payroll Data.*
- Individual *Exfiltrates* is a member of object property *Exfiltrates.*

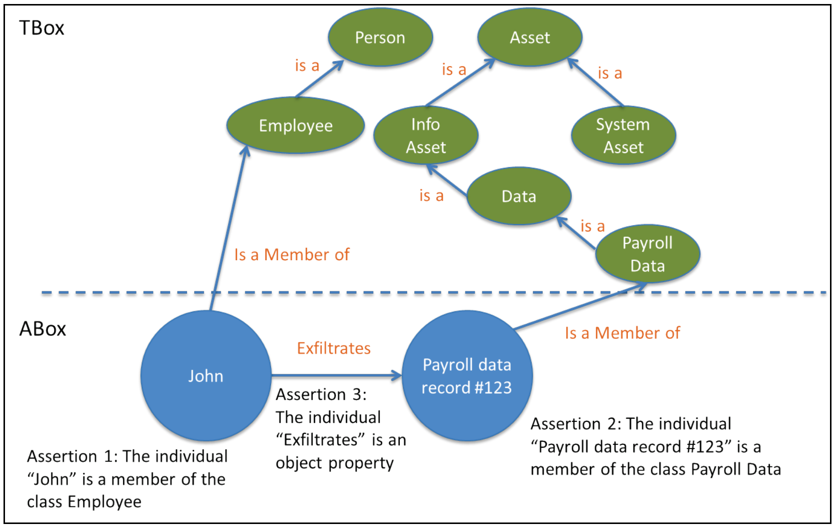

TBox
Person
Asset
is a
is a
is a
Employee
Info Asset
System Asset
is a
Data
is a
Payroll Data
Is a Member of
ABox
John
Exfiltrates
Payroll data record #123
Is a Member of
Assertion 3: The individual "Exfiltrates" is an object property
Assertion 2: The individual "Payroll data record #123" is a member of the class Payroll Data
Assertion 1: The individual "John" is a member of the class Employee

Figure 2: TBox and ABox Modified by Assertions

Figure 3 shows how the ontology and a reasoner can be used to make inferences about individuals. In this case, the reasoner infers, based on our assertions and a defined class for *Insider Threat Indicator #1*, that *John* is a member *of Insider Threat Indicator #1*.

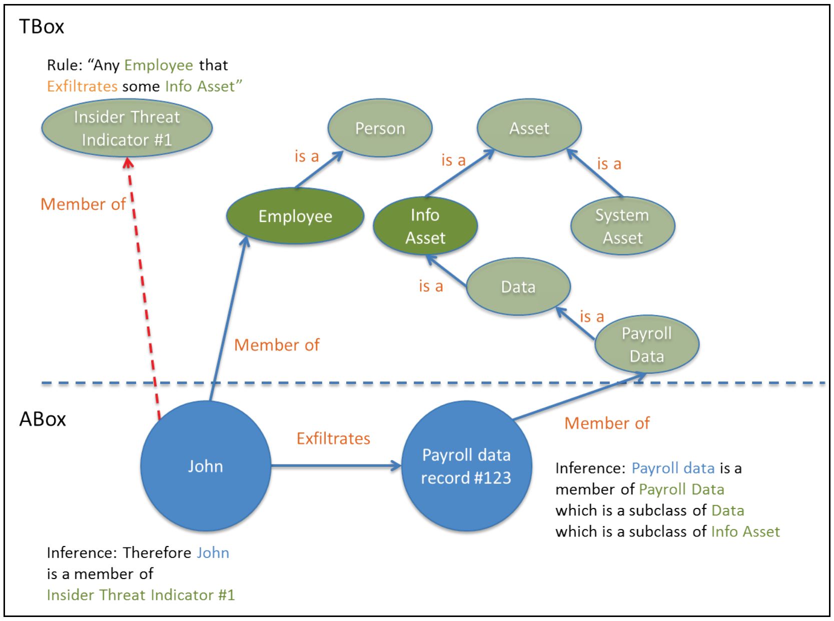

TBox
Rule: "Any Employee that Exfiltrates some Info Asset"
Insider Threat Indicator #1
Person
Asset
is a
is a
is a
Member of
Employee
Info Asset
System Asset
is a
Data
is a
Payroll Data
Member of
ABox
John
Exfiltrates
Payroll data record #123
Member of
Inference: Payroll data is a member of Payroll Data which is a subclass of Data which is a subclass of Info Asset
Inference: Therefore John is a member of Insider Threat Indicator #1

Figure 3: An Inference by the Reasoner

Separating the TBox and ABox provides the following benefits:

- One group's TBox knowledge can be used against another group's data, if the data can be properly asserted into the ABox.
- Informal rules can be formally translated into class definitions for exchange.
- Asserted knowledge allows inference of other knowledge. This new knowledge may be unknown or unexplored by the end user.

Figure 4 shows the true scale of the TBox to the ABox. The TBox of the ontology is small compared to the number of individuals that are expressed using the terms of the ontology.

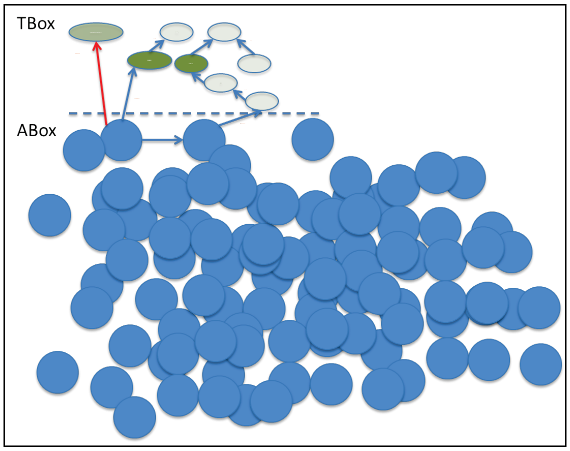


Figure 4: True Scale of the TBox and the ABox

2.2 Challenges

As previously stated, semantic data models are formal by design and have their own limitations when compared to the descriptive power of natural language and the nuances of events it can express. We encountered a variety of challenges to creating an ontological model of insider threat that also satisfied our competency questions and goals for intended use. Below is a summary of the most important challenges encountered.

2.2.1 Identifying the Domain

Building an ontology for a particular knowledge domain requires a thorough understanding of the scope of that domain. The insider threat domain presents a challenge because it reaches across multiple domains: information technology, human behavior, interpersonal relationships, and workplaces are just a few. Given the breadth of related domains, the challenge is to model enough of each domain to satisfy the competency questions of the ontology while avoiding the urge to model all of the related domains in their entirety.

2.2.2 *N*-ary Relationships

Modeling the actions of an insider threat brought us quickly to the challenge of *n*-ary relationships, which are relationships involving the connection of more than two things. The actions of insider threats and consequent events are often complex and require an amount of descriptive detail that can only be stated using *n*-ary relationship modeling patterns [23]. Choosing to use an *n*-ary relationship to model a concept is a non-trivial decision. Among other challenges, recognizing an *n*-ary relationship in source data and properly extracting it is still an area of active research [24].

2.2.3 Intended Use

Ontologies should be developed with a purpose in mind andthat purpose should include the questions that querying the ontology will answer and the ontology's intended application. Competency questions are a typical way of capturing intended usage and can help determine the requirements or goals of the ontology [25, 26]. The competency questions can help the modeling team find appropriate scoping limits and can provide guidance when modeling problems are encountered (such as the choice of using or not using an *n*-ary relationship).

2.2.4 Open-World Assumption

Traditional software systems are often built using a closed-world assumption. In those solutions, the absence of data can be used to make a decidable outcome. This allows systems to work within defined constraints and use only data that is available. The absence of data in a system designed using a closed-world assumption is treated as proof that the data does not exist. The opposite is true in systems that employ the open-world assumption.

Many of the most widely used ontology languages employ the open-world assumption. The basic use of the open-world assumption allows a system to hold open possible outcomes in the event that data is missing. In the insider threat domain, information is often missing: past events may not be remembered; recordings, files, or backup tapes indicating malicious activity may be lost or mishandled; and accidents can and do happen. Prosecutors of insider threat cases often highlight the presence of a single event because it gives the plaintiff the best chance of proving the existence of malicious activity in court.

Open-world solutions do not treat missing data as proof of non-occurrence, nor as proof of occurrence. The data simply remains missing until it is found and declared to the system. If it is never found, the system simply waits. The benefits of this assumption shift are both an advantage and a challenge. Open-world systems typically require closure of some kind to be able to declare anything (closure axioms) [27]. Designing the right amount of closure to include in an open-world system while still maintaining the benefits of using a system at all requires delicate and thoughtful modeling choices.

2.2.5 Temporality

The chronology of actions and events is a key part of the insider threat domain. All insider cases contain both events and actions, and their specific times of occurrence sway many human opinions or judgments. The accurate representation of chronology becomes even more important when actions and events that are related to or caused by the actions of an insider threat occur on information technology (IT). We had to think carefully about how we would represent the multiple types of time data and time events. Additionally, it is often difficult to reconstruct or record all of the events that occurred throughout the insider's activities.

3 Approach

3.1 Purpose and Application of Ontology

Our ontology is built to support the detection, creation, sharing, and analysis of indicators of insider threat. Because insider data is sensitive, insider threat teams frequently work only with data from inside their own organizations. These records frequently include documented employee behaviors, organizational intellectual property, employee activity on networks, and information on organizational proprietary networks and IT architecture. Organizations and teams are unlikely to release this information due to the risk of breaching employee privacy, releasing sensitive organizational information, or unnecessarily losing a competitive advantage. A shared ontology will allow these teams to share indicators of insider threat—initially we have focused on cyber indicators—without sharing their own sensitive organizational data.

For many organizations, establishing an insider threat program and beginning to look for potentially malicious insider activity is a new business activity. In particular, Executive Order 13587 has recently prompted government organizations to begin building insider threat programs. This and the National Insider Threat Policy describe minimum standards for establishing an insider threat program and monitoring employee use of classified networks for malicious activity, and can be used as guidance for all organizations looking to build insider threat programs [28-30].

Our desired outcome is to allow teams to share detection rules. We made our design choices for the ontology with an eye toward extensibility, semi-automation of indicator creation, and the ability of the community to benefit from investigations performed by others.

Competency questions are a typical way of capturing intended usage and can help determine the requirements or goals of an ontology [25, 26]. Our proposed competency questions for the insider threat indicator ontology are

- What indicators of insider threat activity are other teams using for detection?
- What insider threat indicator schema can I use to create and store my own indicators using a commonly accepted format that can be analyzed by myself and other teams?
- How can I participate in a community to both share and receive indicators of insider threat activity without divulging internal information?

3.2 Domain Scoping

We chose our domains based on our competency questions and intended applications of the ontology. To further assist our domain scoping, we built concept maps from our source data to identify important and frequently occurring concepts and relationships.

Concept maps are used to graphically organize and represent knowledge [31]. At their core, concept maps are made up of *triples*, which include two concepts and some relationship label that links them. The concepts from the triples are the important domain elements of the ontology, and the relationships show how the concepts are linked. Using the concept maps to express our information sources allowed us to better understand and identify indicators of insider threat. For this work, we adapted an approach from past work that also used concept maps as the first step to building an ontology [32]. We then developed a method for automatically producing concept maps from our data sources [33-35]. This method involved using text analysis with the Natural Language Toolkit to identify concepts and relationships and automatically extract triples [36]. These triples were then converted to concept maps, which we viewed in CmapTools and manually analyzed for indicators of insider threat [37].

3.2.1 Cyber Assets

One of the biggest hurdles during the creation of our ontology was determining where our domain stopped. We discussed the need to model the knowledge necessary to detect malicious activity occurring against an organization's critical assets, specifically, the assets supported by or located on IT systems. This helped us clarify that the cyber domain needed to be represented in our ontology. We determined that the model for our cyber domain should include at least the important computing systems, networks, technology, physical items, virtual items and activities, programs, infrastructure, devices, data, and operational processes that organizations commonly use. The cyber domain on its own is not enough, however. We also needed to include elements of the weaknesses, threats, problems, failures, vulnerabilities, and other accidents that could occur in such systems. We consulted with numerous previous ontologies for inspiration, including Network Services, IT Systems, IT Security, Mobile Devices, and more [38-41]. Though we found their decisions on the domain extremely useful, we mostly used them for inspiration rather than re-use. Our key decision criteria for when to perform re-use were based on our goals and intended use of the ontology.

3.2.2 Organizations

We used existing schema to describe an organization. Most of the organizations in our case data were some kind of legally recognized entity such as a limited liability corporation, partnership, or non-profit. We also included some special organizations such as government entities and law enforcement offices. We borrowed other concepts for describing our organization class from the organization classification at schema.org [42].

3.2.3 Organizational Environment

Insider threat actions are sometimes a subtle and debatable offense. The activities of employees or other insiders, such as reading the newspaper, playing games, or chatting in the hallway, are often not directly in pursuit of an organization's mission or bottom line. However, innovative cultures think about employee time differently [43]. One argument is that free time encourages employee innovation; for example, hallway chats create cross-team connections and can contribute to improved collaboration, and reading the news can help employees generate ideas for new products. In summary, the culture, policies, and attitude at the organization may matter as much or more than the act performed by an insider threat.

3.2.4 Events, Actions, Activities, Time, and Importance (Also Referred to as Context)

Another important element of our domain is the complexity of the actions and activity needed to accurately describe what is happening leading up to, during, and after an insider threat event occurs. Not only did we need an accurate description of actions and activity, but we also needed to attach specific details (also called *properties*) to those actions, such as, was the action deliberate? When did it occur? Why did the person decide the action was necessary? We also needed an element of time or temporality because many insider cases are in fact a series of important activities that are chained together to create a summary of major events. We again borrowed from existing literature on how other teams model temporality in other domains [44].

Our ontology design allows for the description of important events, including the ability to link the actions of humans leading up to or specifically causing the event. It also allows for the linking of detailed instances that occur in the IT domain as evidence of the activity of a human or a human creation in the form of programming code or even code created by other programs [45]. At the beginning of modeling an insider threat event, it is often not known which events are important, so we have focused the bulk of our modeling effort on modeling actions. These actions can be linked into chains and represented as events, or they can be kept at the action level. This approach allows a certain amount of drill-down from an important event into the actions that contributed to the event's occurrence. Actions can also leave behind information at lower levels, particularly in the IT domain. We have left certain details (such as the list of all instructions sent to a processor to open a connection) for later effort.

3.2.5 People

One of the key distinguishing factors for the insider threat domain is its intersection with both the social and psychological behavior of individual people. Drawing on previous research and definitions of insider threat activity [46, 47], we attributed each insider threat with some level of existing trust relationship with their victims and some activity that is outside the expectations of that trust relationship. This is a long way of saying that the insider had some level of approved access to something inside the organization and exceeded that level of access. From employees to business partners and CEOs to entry-level personnel, insiders act outside the trust expectations that others in the organization set for them. Unfortunately, the human domain is complex, and as a result the reasons behind certain behaviors are inherently complex. Because of this complexity, we have attempted to describe a core model that fits our application of the ontology and that allows for other experts to hang more nuanced information and interactions on our classes.

3.2.6 Human Emotion, Behavior, and Intent

We chose to model a few choice properties of people that would be relevant for describing their motivation for an action, including emotion, behavior, and intent. We again relied on existing schema for the human domain [48] and also consulted theories of human intent [49]. We also drew inspiration for our model from insider threat studies on human behavior [2, 50, 51]. The modeling of human intent remains a work in progress, and not all of our thinking on this property has made its way into our formal ontological model. However, it is safe to say that some insider threat actions were preceded by a human intention of some kind, and we will eventually need some way of storing information related to this concept.

3.2.7 Human Networks, Interactions, and Information Exchange

The domain of human social networks is also inherently complex. Our specific interest is typically in detecting the networks of humans that are also insider threats, and this can overlap with the inner workings of crime activity such as conspiracy. We reviewed existing ontologies for the domain of human networks [52] and found many inspiring and relevant classes, but they did not quite meet our need for describing malicious activities. The class of social circle has relevancy for insider threat groups and actions, which can be conducted with conspirators in the insider's social circle. The primary goal of the friend-of-a-friend (FOAF) ontology is to link content created on the web with the people that created it (such as the output of a social circle). This is similar but not quite the same as our interest, which is to model cases where malicious activity is the primary goal such as a ring of insiders committing fraud at a company. Malicious group activity probably best fits as an expansion of the FOAF class for a Group Project, and we are continuing to consult the FOAF ontology to evaluate its core for describing groups of insiders. We also consulted with other ontologies of criminal acts [53] and made our own adjustments to meet our stated guidelines and key focus areas.

3.2.8 Malicious Activities, Including Deliberate and Intentional Harm, Theft, and Sabotage

We have attempted to model the common actions that humans perform, especially those occurring in a cyber context. But we also focused our modeling on actions that are malicious and that can be or were specifically performed by someone with inside information on the organization. During our prior work studying the patterns in different types of insider crimes, we distinguished different types of intentionally harmful behavior toward a specific desired outcome such as IT sabotage, fraud, or theft of intellectual property [54]. Our approach for this domain was to incorporate the common actions taken by insider threats toward those outcomes as they were documented in our case data. This remains an area of active research, and we expect to continue adding new actions as they are encountered.

3.3 Ontology Architecture Decisions

We chose to implement our ontology using the second version of the Web Ontology Language (OWL 2). The primary reasons for this decision are as follows:

- maturity and wide use—OWL 2, published by the World Wide Web Consortium (W3C) in 2008, is an extension of OWL 1, which was published in 2002. OWL is endorsed by the W3C, the main international standards organization for the World Wide Web. OWL is highly conducive to formal knowledge sharing and has been used as a formal representation for a wide range of knowledge bases [55].
- interoperability—The OWL format is supported by a multitude of editors, visualization tools, description logics, and many other applications. OWL allows us flexibility in the applications and use cases our ontology can support. Furthermore, the XML-based OWL format lends

itself to automated creation of ontology components. OWL is also supported by many semantic reasoners, which are applications that can make inferences from a set of assertions.

- deterministic—OWL provides a mechanism for validating classes against axioms and, in a sense, helps to close the open world.

3.4 Ontology Construction Method

We constructed the ontology with incident story summaries from our MERIT database (see Section 3.4.1.1, Insider Threat Databases). The story summaries are sanitized descriptions of real cases of malicious insider threat and include details about the insider, the attack, and sentencing. We extracted triples, consisting of two concepts and a relation label [34], from these story summaries. These triples were then used to build concept maps (see Section 3.2, Domain Scoping), which helped to develop our focus competency questions (see Section 3.1, Purpose and Application of Ontology).

After developing our competency questions, we applied the questions to the triples to derive entities and object properties. We then tested the ontology on our data to determine how effectively our ontology can express indicators of insider threat. As we collect data from various sources, we will repeat the process of extracting triples and adding them to the ontology with the end goal of improving the ontology's ability to express indicators of insider threat.

3.4.1 Data Sources

We used a variety of data sources to develop and construct our ontology. Our primary resource used for the Insider Threat Indicator Ontology is the collection of insider threat cases from our MERIT database and the collection of espionage cases in our SpyDR database. We analyzed the data from these resources to develop a set of indicators that occurred across multiple cases. We then modified the ontology to make it capable of expressing these indicators. We also modified the ontology to be able to express artifacts from Microsoft Windows event logs, in addition to the content of our databases. These artifacts are valuable in expressing an end user's actions that can be a potential indicator of insider threat.

To date, we have collected approximately 800 cases in which insiders used IT to disrupt an organization's critical IT services, commit fraud against an organization, steal intellectual property, or conduct national security espionage. We have also collected cases of insiders using IT in a way that should have been a concern to an organization. This data provides the foundation for our insider threat research, insider threat lab, insider threat assessments, workshops, exercises, and the models developed to describe how the crimes evolve over time [56].

The following are the sources of information used to code insider threat cases:

- public sources of information
  - media reports
  - court documents
  - publications
- nonpublic sources of information
  - law enforcement investigations

- organization investigations
- interviews with victim organizations
- interviews with convicted insiders

3.4.1.1 Insider Threat Databases

The CERT Insider Threat Center has two databases containing structured information about insider threat. The MERIT database contains information about cases of malicious insider threat involving fraud, sabotage, or the theft of intellectual property. The SpyDR database contains cases of national espionage. The CERT Insider Threat Center uses the cases from these databases to develop indicators of malicious insider activity, which themselves are used to develop best practices. These best practices can be found in the CERT Insider Threat Center's *Common Sense Guide to Mitigating Insider Threats, 4th Edition* [1].

The databases have been built over time using public and private data sources. We code the information from our data sources into structured and free-text fields in the database. Coding insider threat cases requires information about three entities: the organization(s) involved, the individual perpetrator (subject), and the details of the incident. Figure 5 shows the primary relationships among these three entities [54].

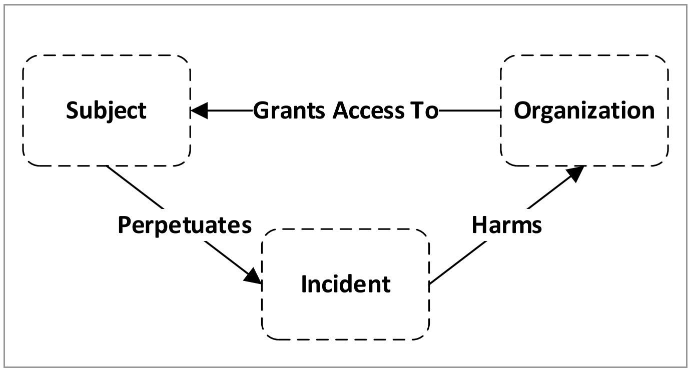


Figure 5: MERIT Model

3.4.1.2 Forensics Toolkit and Other Sources

In addition to information from our insider threat databases, we also designed the ontology to handle information from digital forensics data. Locard's Exchange Principle, a concept from crime scene forensics, is the premise that "every contact leaves a trace" [57]. Locard's principle can be applied to digital forensics as well as physical crimes.

By default and without requiring the user to enable any additional options, Microsoft Windows collects a large amount of information about a user's activities on the system. Windows uses this information to enhance a user's experience. For example, Microsoft Windows can auto-complete certain types of information or provide the user with a list of most recently used documents. Information to enable these features is stored in various operating system files and can be of great use when conducting a forensic examination of a system that has been used by a malicious insider.

Artifacts a user leaves behind on a machine describe who, what, when, where, and why something occurred. SANS places artifacts into one or more of eight categories [58]:

- File Downloaded
- Program Execution
- File Opening/Creation
- Deleted File or File Knowledge
- Physical Location
- USB or Drive Usage
- Account Usage
- Browser Usage

Artifacts from each of these categories can be used to paint a picture of what a malicious insider may have done to carry out their specific crime. For example, a malicious insider who is exfiltrating intellectual property is likely to leave behind artifacts in the categories of File Opening/Creation, Deleted File or File Knowledge, USB or Drive Usage, and Browser Usage.

3.4.1.3 STIX and CybOX

To further describe forensic artifacts, we also included relevant concepts and definitions from the MITRE Corporation's Cyber Observables (CybOX) [59] and Structured Threat Information Expression (STIX) [60] standards. CybOX provides structured representations for enterprise cybersecurity observables, and STIX provides structured representations for descriptions of cyber threats. STIX uses CybOX to describe specific observables. For the purposes of this report's research, a group of subject matter experts (SMEs) examined all CybOX objects as well as the STIX indicator components as references for ontology concepts. The SME group achieved consensus on which STIX and CybOX concepts and ideas would be included in the ontology. Generalized, higher level concepts were included, whereas highly specific concepts, such as DNS record or network route objects, were discarded so that the ontology is able to operate at a higher conceptual level.

3.4.2 Text and Language Processing

Due to the size of our corpus, we chose to use natural language processing to help extract the concepts and relationships that are representative of our data and domain. We developed Python scripts that leveraged the Natural Language Toolkit (NLTK) library [36].

To identify the concepts of interest, we used the following approach:

1. Collect all the words from our corpus.
2. Sort the word list by term frequency, and remove stop words and words that appear fewer than 10 times.
3. Use a custom script to show a human evaluator the contextual uses of each word in the corpus and its synonyms, which the evaluator would use to assign a specific word to a high-level category.
4. Use group consensus to break high-level categories into subcategories.
5. Identify the "is-a" relationships between subcategories to build out a hierarchy.

We made one of our high-level categories "Actions" and used it to group the verbs and actions we found in our corpus. We used our subject matter expertise, domain scoping, and competency questions to manually trim the action list to approximately 200 terms. Once the concepts were converted into a hierarchical arrangement of ontology classes, we used the following process to identify the relationships between the classes that the ontology needed to express:

1. For each case in our corpus
    a. Tokenize each case description into sentences.
    b. Identify the parts of speech for each word in each sentence.
    c. Use a regular expression parser to extract concepts (noun phrases) and relationships (verb phrases) from each sentence.
    d. Use parts-of-sentence grouping to create triples (concept, concept, relation label).
2. Using the collection of triples from the previous step as input
    a. Use a custom script to find all triples in the corpus that contain a tense or plurality variant of the action.
    b. Leverage our subject matter expertise, domain scoping, and competency questions to identify the relation labels that represented domain-relevant actions.
    c. Store the concepts associated with each action/relation label to facilitate ontology domain and range-setting activities.

3.4.2.1 Part-of-Speech Tagging

The part-of-speech tagging used for this work was performed by a custom-built part-of-speech tagger. To maximize precision and cover, we built our part-of-speech tagger by using the NLTK library and a series of cascading n-gram taggers [61]. This means that our tagger first attempted to assign a part of speech to a word by looking at the word and its two preceding words. If the tagger could not make a reasonable prediction at the trigram level, it would try to assign the part of speech by looking at the word and its preceding word. If this second attempt was unsuccessful, the tagger would use just the word of interest alone. The part-of-speech tagger was trained on the Brown Corpus [62], a collection text samples containing more than a million words with manually tagged parts of speech assigned to each word. We ran tests to identify the most accurate training/test split for our tagger against the Brown Corpus, and we found that a 90/10 training/test split produced the most accurate tagger.

3.4.2.2 Part-of-Sentence Tagging

The part-of-sentence tagging performed in this work used a parser that used regular expressions to group specific sequences of parts of speech as parts of sentences. Our goal for tagging parts of sentences was to extract concepts and relationships between concepts from sentences. Because our data entities were largely written in the same style, we chose to focus on extracting concepts and relationships from sentences using the basic subject-verb-object syntax.

The regular expressions used for concepts (noun phrases) and relationships (verb phrases) were developed using a two-step approach. Initial expressions were first created by looking at the part-of-speech tags associated with manually tagged noun and verb phrases in a set of training data.

These were then modified as exceptions were found when analyzing test data. Relationship triples were extracted by looking at each verb phrase and identifying the immediately preceding and proceeding noun phrases.

4 Implementation

This section presents our ontology from a design perspective. It provides a high-level overview of the classes, relationships, data attributes, naming conventions, and other implementation considerations of our ontology. For complete documentation of the ontology, see Appendix A.

4.1 Entity Model

Our top-level logical entity model is comprised of five classes: *Action*, *Actor*, *Asset*, *Event*, and *Information*. To better model temporality, *Action* and *Event* are technically subclasses of *TemporalThing.* However, they can conceptually be thought of as siblings with the other top-level classes. The following subsections present class hierarchy diagrams for each top-level class.

4.1.1 Actor

The *Actor* class contains subclasses that represent people and organizations.

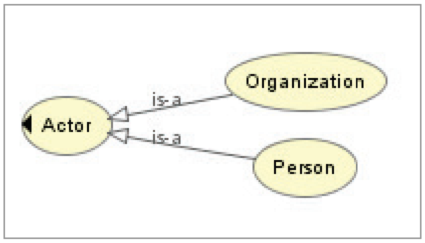


Figure 6: Actor Class Hierarchy

4.1.2 Action

The *Action* class and its subclasses define the actions that actors in our domain can perform. The *ActionModifier* subclass contains subclasses that are qualitative modifiers that are meant to be used in combination with other subclasses of *Action*. For example, to model a suspicious search action, an individual could be assigned to the classes *SearchAction* and *SuspiciousAction*.

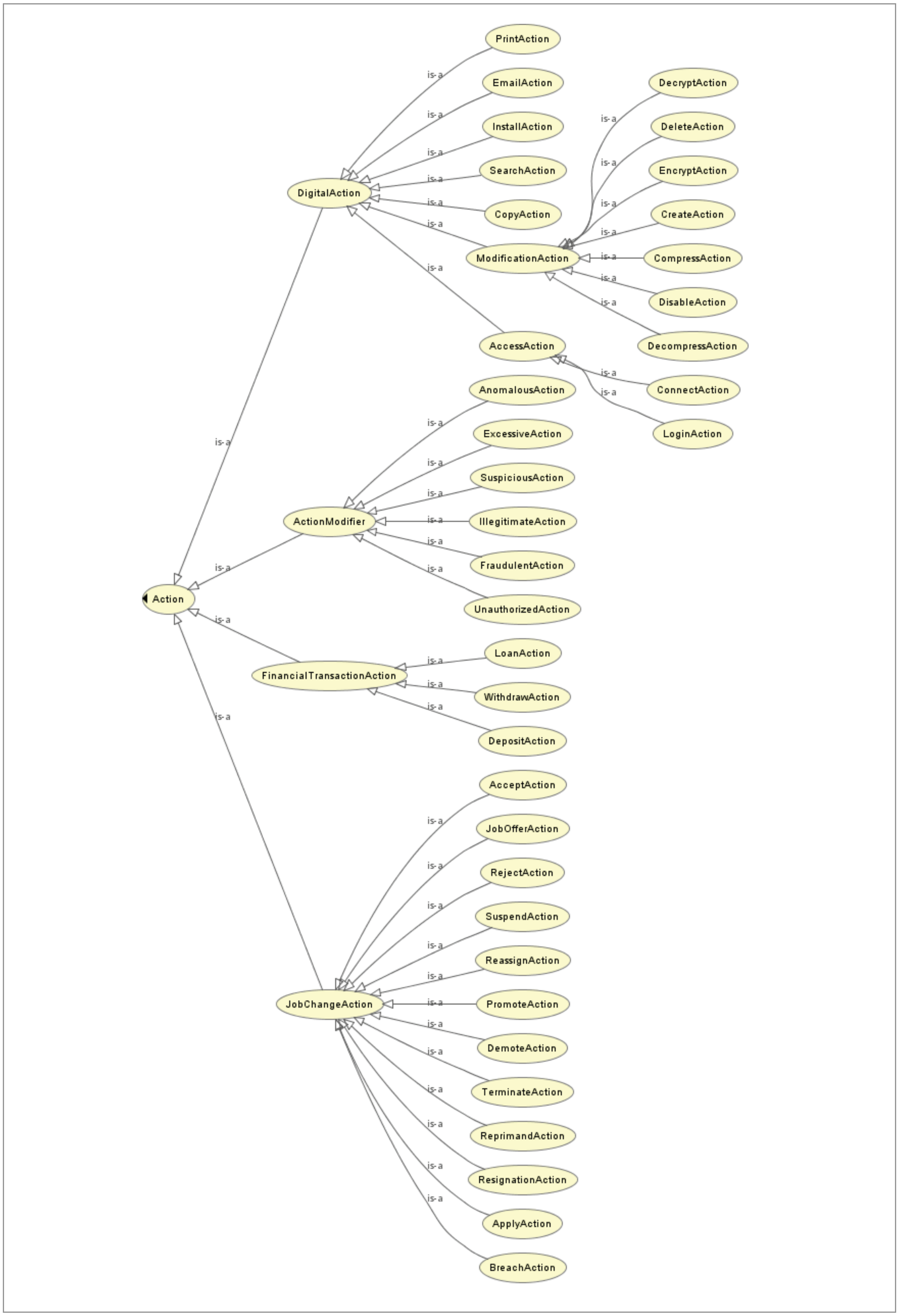


Figure 7: Action Class Hierarchy

4.1.3 Event

We ultimately chose to represent the actions of insiders as one class and to separately model events as their own class.

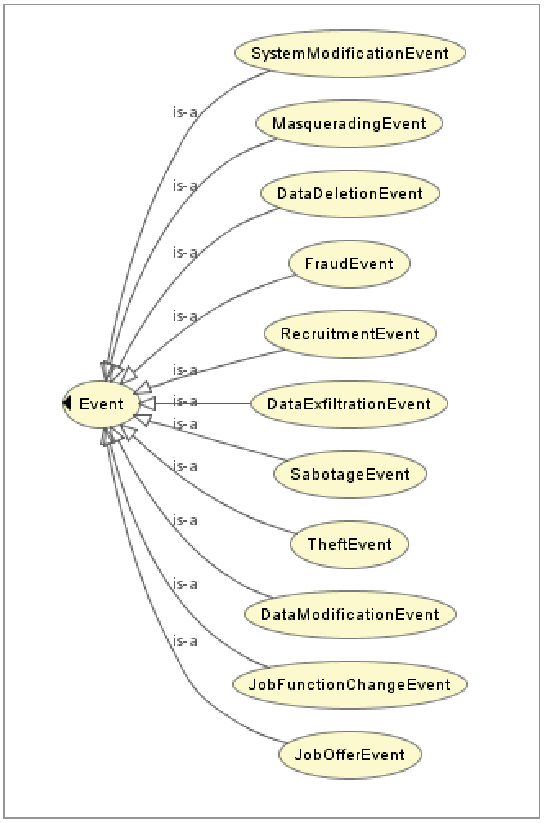


Figure 8: Event Class Hierarchy

Events are the mechanism by which multiple actions can be grouped together and related by some qualitative or contextual analysis. To put a finer point on the differentiation between actions and events, we classify actions as what is observed and events as what is inferred. The following example from our ontology illustrates this difference. The ontology contains a subclass named *DataExfiltrationEvent*. Data exfiltration is the unauthorized copying, transferring, or retrieving of data from a computer or server [63]. Data exfiltration itself is not technically observable, but the specific actions of copying, transferring, or retrieving data associated with the exfiltration are observable. Some qualitative analysis of these actions would be required to determine whether or not they were unauthorized. If so, the specific action could then be said to correspond to a data exfiltration event.

4.1.4 Asset

The *Asset* class contains subclasses that represent the targets of actions, or instruments used objects of actions in our domain.

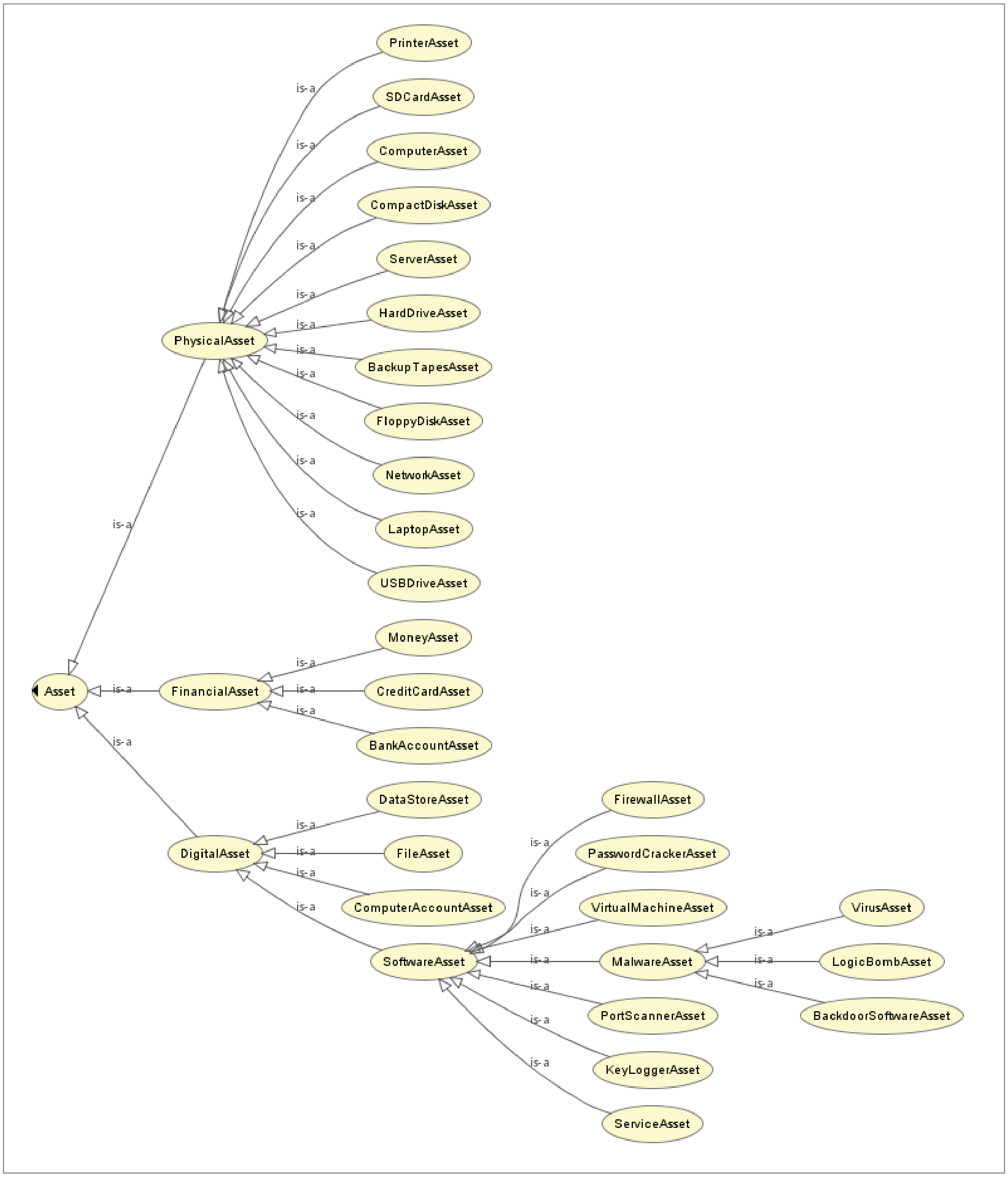


Figure 9: Asset Class Hierarchy

4.1.5 Information

The *Information* class contains subclasses for types of information affected by actions.

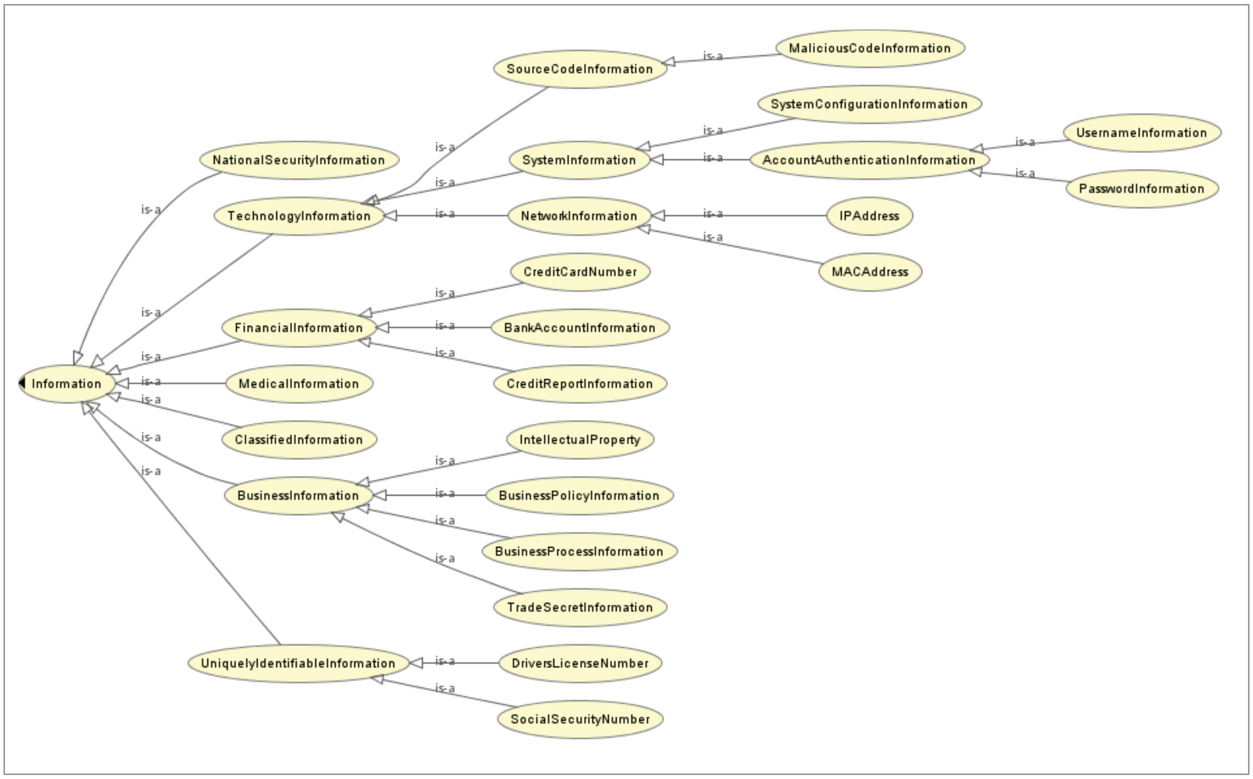


Figure 10: Information Class Hierarchy

4.1.6 Annotations

Definitions for each class are needed to ensure that the terms have the same meaning to everyone using the ontology. We defined each class according to the *rdfs:isDefinedBy* annotation. We drew the conceptual content of class definitions from various subject matter expert sources, such as the Society for Human Resource Management and the Office of the Comptroller of the Currency. We derived some additional class definitions from other internet sources that the CERT Insider Threat Center has generally accepted, as well as from the CERT Insider Threat Center's published works. Sources for the definitions are denoted by the *rdfs:definitionReference* class.

Some of the classes are domain specific—that is, they describe a malicious insider threat incident. However, they may have other meanings outside of the insider threat domain. We limited our definitions to those that are applicable to malicious insider threat incidents.

For some classes in the ontology, semantic synonym sets are included and are annotated using the *rdfs:seeAlso* annotation. The semantic synonym sets capture equivalent classes and relationships relative to the domain of our ontology. We decided to not explicitly create equivalence classes and relationships in our ontology, primarily to minimize ambiguity by limiting the number of ways a concept or relationship can be ontologically expressed. It is still important to capture equivalency relationships for two major purposes: to provide users of the ontology additional reference in using the ontology components, and as a resource for automatic creation of individuals within the ontology.

Automated tools can use the semantic synonym sets as candidate individuals, meaning that if a term appears in a particular class or object property's list of semantic synonyms, that term can be added as an individual instance of that class or property. If an instance of a specific action can be identified in a corpus, its associated events (and in turn, that event's other associated actions, actors, and assets) can be searched for in the corpus.

4.2 Object Properties

The object property hierarchy provides the ability to specify various types of familial, work-based and event-based relationships between actors. The object property also provides relationships for associating various actors and assets to actions via properties such as *hasActor*, *hasAsset*, *hasObject*, and *hasInstrument*. The object property hierarchy also specifies a subproperty hierarchy for temporality, which is discussed in Section 4.3. For a complete listing of the ontology's object properties, refer to Appendix B.

4.3 Temporality

The *Action* and *Event* classes are logical top-level class elements, but in our actual implementation, they are subclasses of the *TemporalThing* class. This is so that actions and events can leverage the same object property hierarchy for temporality. Actions and events can be temporally related to direct points in time (using the *TemporalInterval* subclass hierarchy), or to a relative sequence of other actions or events.

We have chosen to reuse components from Eric Peterson's SpaceTime Ontology [64] to model temporality in our ontology. The SpaceTime ontology is an extensive semantic model of entities and relations having to do with spatio-temporal reasoning. From the SpaceTime ontology's entity model, we have reused the *TemporalThing* class (which is the parent class of our *Action* and *Event* class hierarchies, as described above), and the *TemporalInterval* subclass hierarchy. From the SpaceTime's object property model, we have chosen to reuse a small subset of properties which map directly to Allen's Interval Algebra [65], a calculus for temporal reasoning, as the basis for many of the SpaceTime object properties. (See the *temporallyRelatedTo* object property hierarchy in Appendix B for the full object property listing.) Allen's interval algebra specifies the following base relations as being able to capture the possible relations between two intervals, X and Y:

- X takes place before/after Y
- X meets Y (the end of X is equal to the beginning of Y)
- X overlaps with Y (the end of X occurs before the end of Y, and Y starts before X ends)
- X starts Y (X and Y's starting times are equivalent, and X ends before Y ends)
- X during Y (X starts and ends in between the starting and ending of Y)
- X finishes Y (the start of X occurs in between the starting and ending of Y)
- X is equal to Y (the time intervals for X and Y are equivalent)

5 Validation

5.1 Introduction to Validation

We used a validation process to analyze the ontology's representation of important insider threat events. We wanted our ontology to retain enough detail to allow analysis of our insider threat case corpus, help us identify existing indicators of insider threat detection, or facilitate development of new indicators of insider threat detection. Some loss of case description was acceptable if we could still fulfill the needs specified in our competency questions. (See Section 3.1 for the full list of our competency questions.)

5.2 Ontology Validation Process

We validated our ontology design by selecting samples of indicators for insider threat that we have identified in insider threat cases. Our MERIT database has observation groupings used as identifiers to categorize specific details in a case. Each observation grouping has an observed detail from the incident and a relevant grouping for the detail. These observation groupings contain precursors, concerning behaviors, and additional details relevant to the incident. To validate our ontology, we selected the observation groupings that are potential cyber indicators of insider threat and modeled them using the ontology. This section discusses the process and results of our validation.

5.2.1 Collect Observation Groupings

Our MERIT database contains a table named "Incident Detail" that includes a step-by-step sequence of events that occur in a given case. Each line item includes a specific detail of the incident and an observation grouping. The observation grouping is a classification of the information described by the detail. In total, there are currently 142 different observation groupings that are organized into the following categories:

1. Personal Predispositions
2. Stressful Events
3. Technical Concerning Actions
4. Behavioral Concerning Actions
5. Actions Directly Related to the Attack
6. Organizational Vulnerabilities
7. Miscellaneous
8. Incident Response

5.2.2 Sample Selection from Observation Groupings

To select our sample, we ordered the observation groupings by the number of times they were used in the MERIT database. We reviewed this list and selected the following 10 most frequently used observation groupings related to the cyber domain:

1. Verification of Modification of Critical Data
2. Disgruntled Employee
3. Used Excessive Access Privilege—General
4. Unauthorized Data Exports—Unknown
5. Compromised Passwords
6. Email/Chat with External Competitors/Conspirators
7. Failure to Protect Critical Files
8. Violation of Need-to-Know Policy
9. Unauthorized Data Download to/from Home
10. Ability of Users with System Administrator Privileges to Sabotage Systems or Data

Appendix C defines these 10 observation groupings. After we identified the 10 most frequent observation groupings from the cyber domain, we selected two sample details from each observation grouping. The sample details are the specific details of the incident that fall into a category from Section 5.2.1. We then analyzed these samples for our validation of the ontology.

5.2.3 Sample Analysis Process

Below is a walkthrough of our sample analysis process, using an example sample detail in natural text:

> *"The insider modified critical data at the victim organization."*

Our key analysis activities during validation were to (1) determine if all the actions from the incident detail are represented in the ontology, (2) identify missing items, and (3) review the representation in our ontology against the real-world domain. We repeated the analysis activities for each action until all of the actions were successfully represented in the ontology. We verified that each action was successfully represented in the ontology by asking the question, "Could we model the necessary events to our desired level of detail in the sample using our current ontology (without modification)?"

From our sample detail above, the phrase "the insider modified critical data" requires the ontology to be able to express

- an action where the result is data modification
- important properties of the data, such as its criticality to the business
- important relationships to the action and the data objects, such as the person that performed the action and the owner of the data

After determining the important concepts to express, we labeled each element of the natural text with its semantic type as a preliminary step to modeling the activity

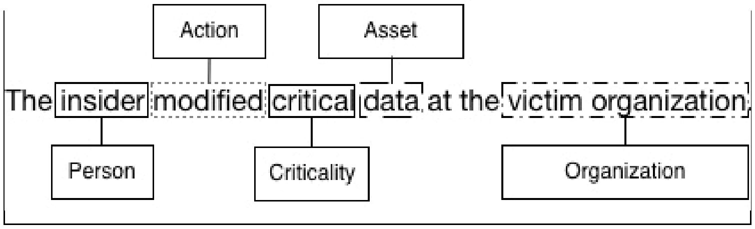

Action
Asset
The insider modified critical data at the victim organization
Person
Criticality
Organization

Figure 11 shows this process using our walkthrough example.

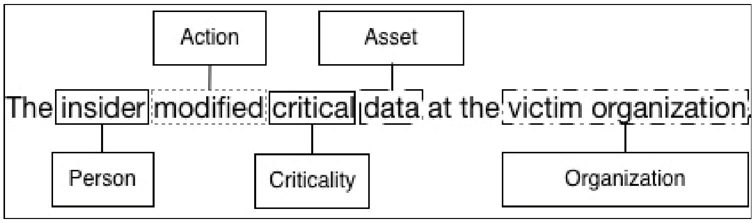

Action
Asset
The insider modified critical data at the victim organization
Person
Criticality
Organization

Figure 11: Analysis of Sample Indicator

We then evaluated the labeled sentence to identify any missing aspects that the ontology should be able to represent.

We also evaluated the labeled sentence for important transformations that would be required to translate the sentence into our ontology while still preserving its original meaning. Sometimes this requires using substitute terms. For example, the term "stole" may become a *TheftAction* with multiple properties. Typically actions or events were our starting point for expressing a given sentence.

Following the translation and the validation of the raw data to our ontology, we then diagrammed the model to provide visualization. Figure 12 shows a key with the symbols used to visualize the ontology.

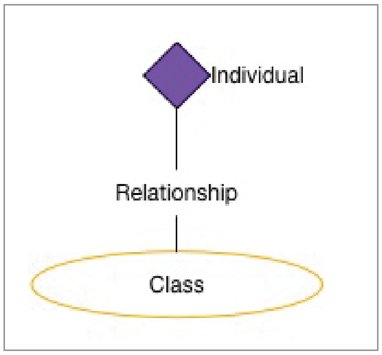


Figure 12: Diagram Key

The following steps summarize the process of going from text to representation in our ontology:

1. Label natural text. Add the semantic types to each part of the sentence.
2. Analyze labeled text. Check the necessary labels are listed for each type and that each concept is represented.
3. Translate labeled text. Represent the important case activities or events using the language defined in our ontology. For example, a description of data that is modified becomes an instance of the class *ModifyAction*, the object property *hasObject*, and an instance of the class *Data*.
4. Model translated text. Model the important aspects of the case activities or events and their important attributes and relationships. Figure 13 shows a model of our sample.

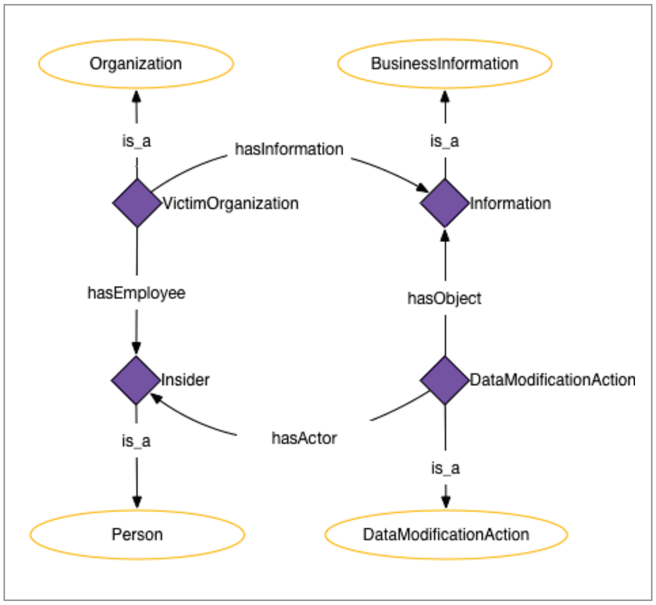


Figure 13: Analysis of Sample Indicator

5.3 Ontology Modeling of Insider Threat Activity

5.3.1 Example Insider Threat Activities in Our Ontology

The following diagrams model excerpts from anonymized versions of MERIT insider threat database cases. Each diagram is a result of labeling, analyzing, translating, and modeling a representative example of an observation grouping from Section 5.2.2.

“The insider stole a co-worker’s password credentials to log into the system and commit fraud.”

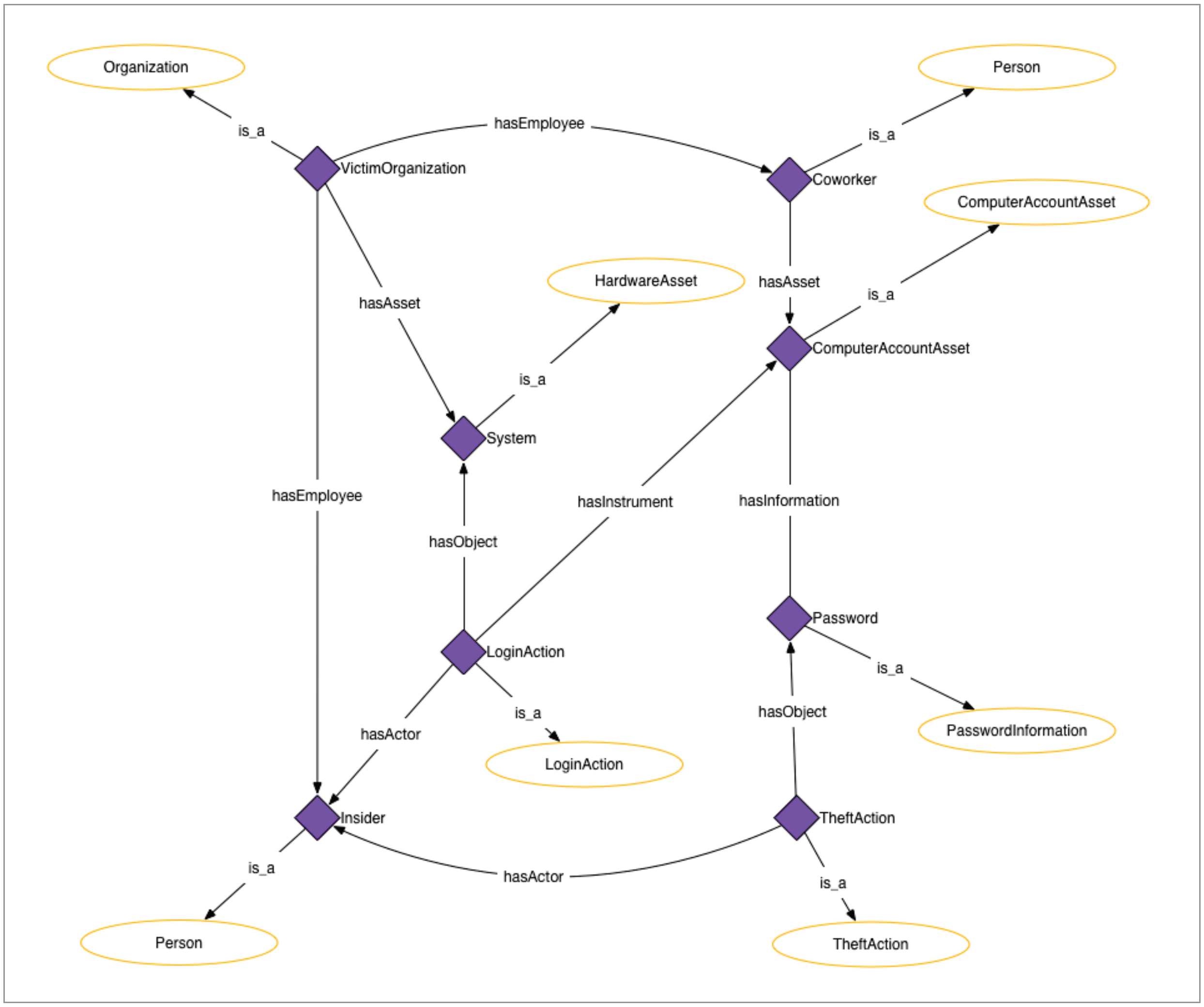


Figure 14: Compromised Passwords—Example 1

"The insider accessed a web server remotely with an administrator account and deleted approximately 1,000 files."

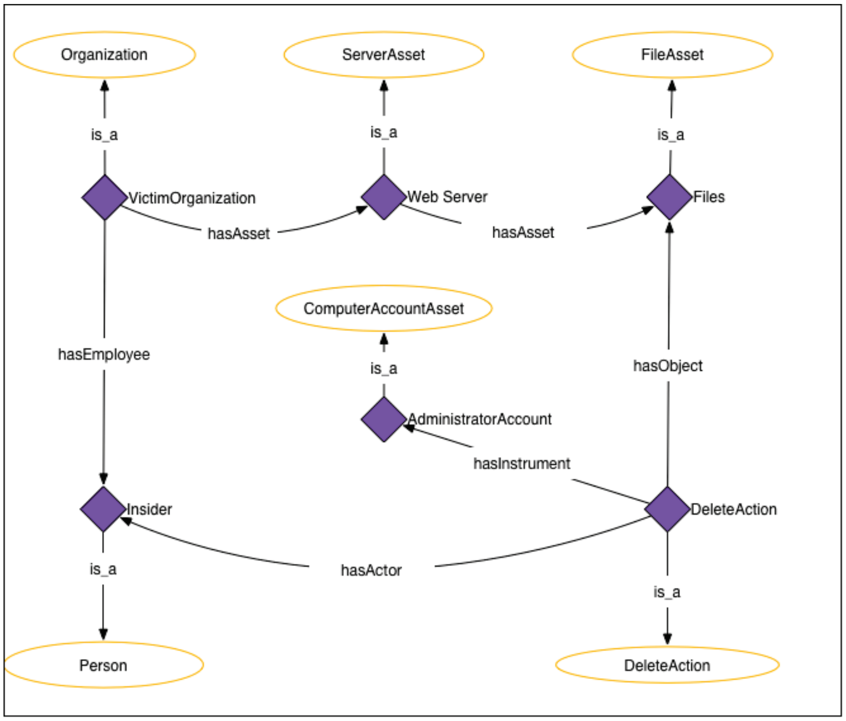


Figure 15: Ability of Users with System Administrator Privileges to Sabotage Systems or Data

“The insider fraudulently entered her husband’s name in the payroll database.”

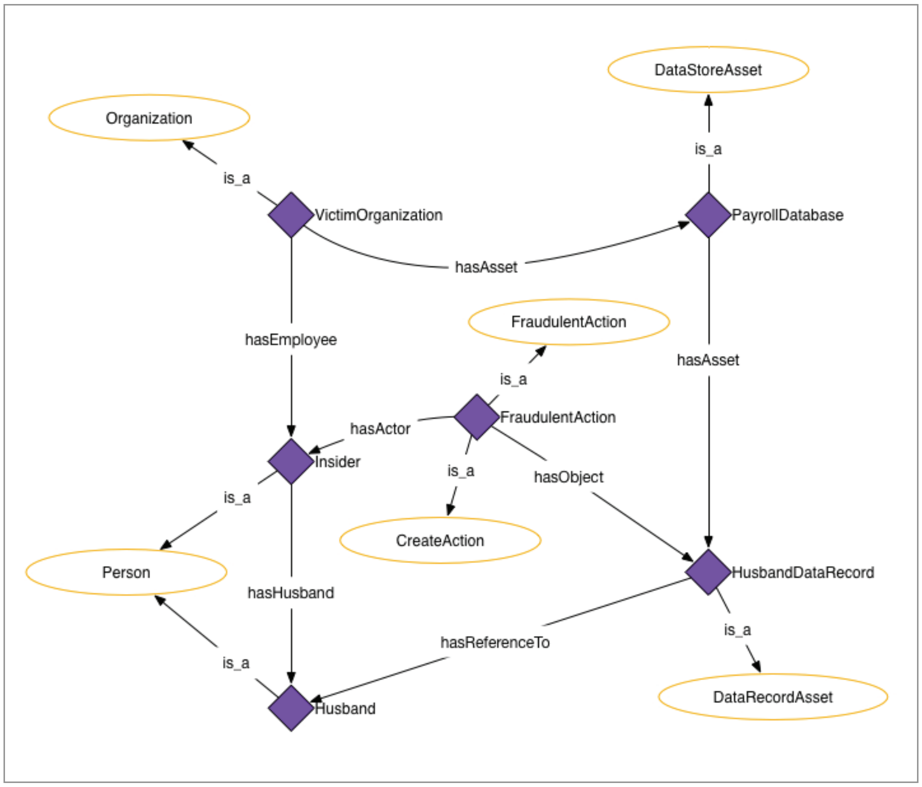


Figure 16: Verification of Modification of Critical Data—Example 2

"The insider made unauthorized copies of confidential information and moved the information to a laptop."

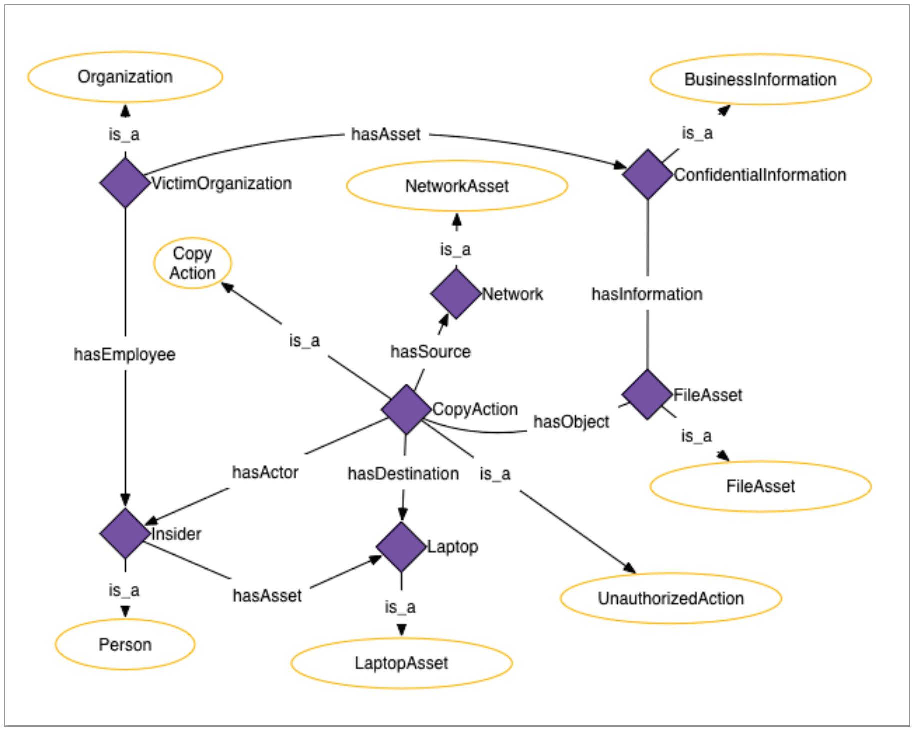


Figure 17: Unauthorized Data Exports—Example 1

"The insider used a co-worker's account to change inventory records."

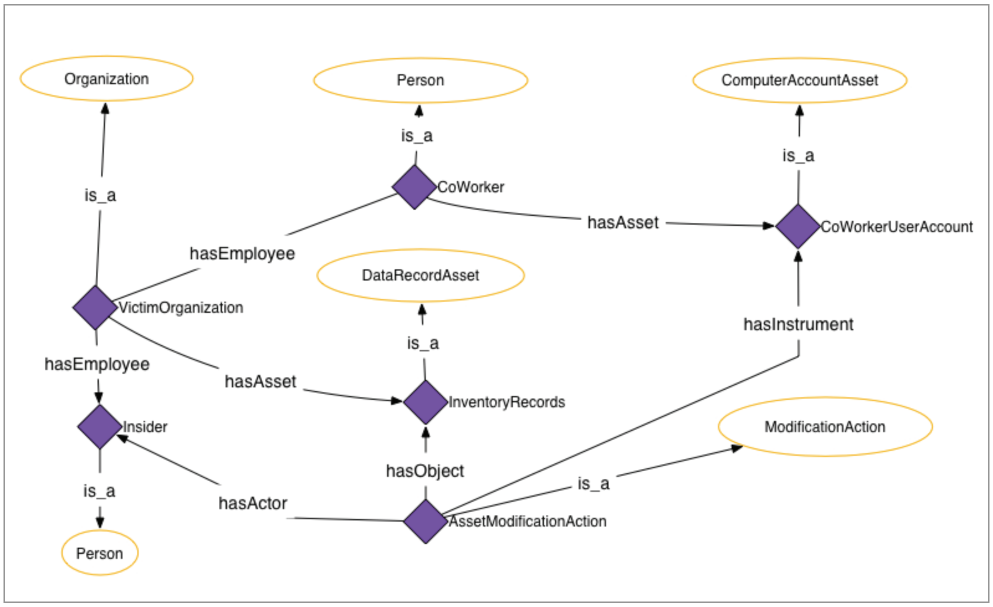


Figure 18: Compromised Passwords—Example 2

"The insider was able to implement his own private network within the organization."

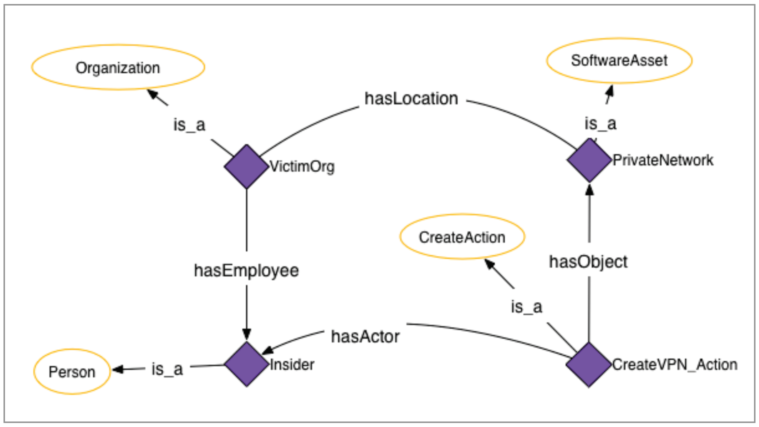


Figure 19: Used Excessive Access Privilege

"The insider changed addresses of medical service providers in the organization's database."

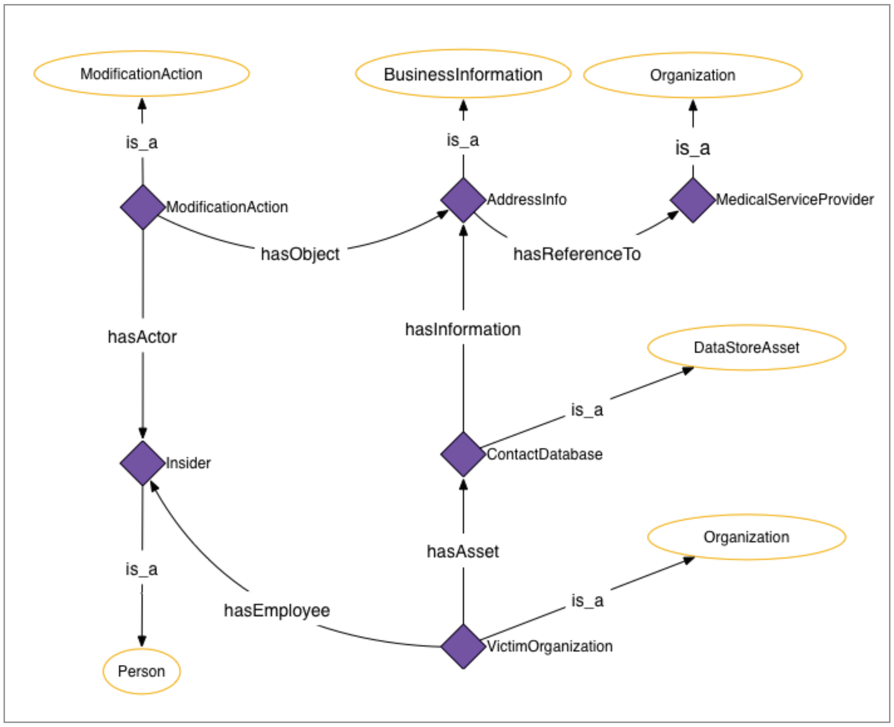


Figure 20: Verification of Modification of Critical Data—Example 2

"The insider transferred proprietary engineering plans from the victim organization's computer systems to his new employer."

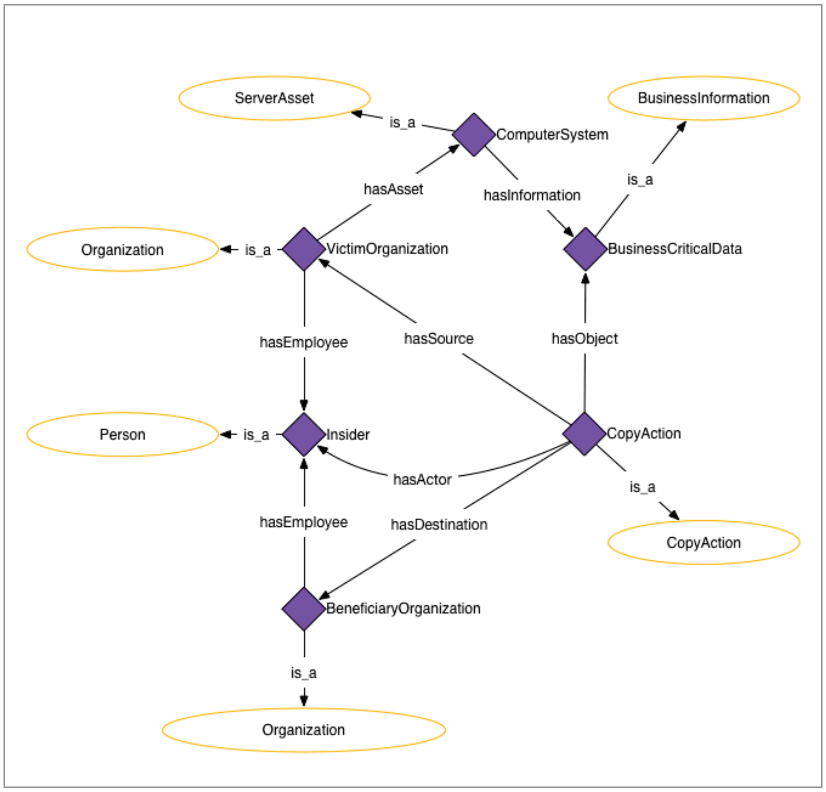


Figure 21: Unauthorized Data Exports—Example 2

5.4 Validation Conclusions

We were successful in expressing the important cyber actions and events in our observation grouping samples using the classes and object properties in our draft ontology. This indicates a successful initial ontology, based on our initial scoping goals. (See Section 3.2, Domain Scoping for a discussion of scope.)

Based on our initial validation efforts, we were able to improve the granularity of classes, and the updated validation cases reflect the ontology's more accurate descriptions of events. The continued validation effort helped us add to and prioritize our list of ontology expansion areas such as temporality of actions and events, verification status of actions, and intentions of actors. Based on the models generated during the validation effort, we have incorporated changes into our initial ontology. We intend to continue validating our initial ontology and incorporate new changes to it using insider threat case data.

6 Next Steps

We have built an ontology that expresses indicators of insider threat that we have found in our data. This ontology provides a starting point for us and others to review and improve on. Future work includes expressing an organization's data as indicators in terms of the ontology, expanding and maintaining the ontology to include indicators found in new data, and sharing these indicators with organizations using the ontology. The end goal is for organizations that use the ontology to be able to communicate indicators of insider threat consistently and without revealing sensitive information.

6.1 Expansion

After building the initial ontology, the next step is to express an organization's data in terms of the ontology. This step requires semi-automatically mapping the organization's monitoring and logging tools to terms used in the ontology. Mappings from data collected by an organization to terms used to express indicators of insider threat in the ontology will be performed on an organization-by-organization basis. Once the data can be expressed in terms of the ontology, organizations can compare their data directly to data from our collection of insider threat cases. Additionally, organizations can consistently express potential indicators of insider threat in their organization and better understand their data.

Our current ontology is scoped to focus only on cyber indicators of insider threat. Further work will need to be performed to consistently capture and express behavioral indicators of insider threat. The quality of the behavioral indicators will be correlated to the ability to automatically capture behavioral indicators, such as electronic badging records of entering a restricted area after hours. Developing a method consistently recording potential behavioral indicators is also a consideration for future work.

6.1.1 Support for Behavioral Indicators

As discussed in previous sections, our implementation efforts have focused on providing support for cyber (or technical) indicators. In future work, we will add ontology support for behavioral indicators of malicious insider activity. The method for extracting behavioral indicators from our data set will vary based on how behavioral data is captured and entered into our data sources.

6.2 Community Feedback

Once an organization expresses its data in terms of the ontology, it can search the data for indicators of insider threat. These indicators can come from our analysis across multiple cases of insider threat or from indicators found and reported by other organizations. The benefit of a widely accepted ontology is the use of consistent language with consistent meaning. If a valuable indicator is found and shared, it can quickly be applied by other organizations. Potential additional analysis across organizations may reveal indicators that are common among certain types of insider crime or in certain industry or government sectors.

Appendix A: Ontology User's Manual

Introduction

This appendix provides guidance about how to use the Insider Threat Indicator Ontology to model a series of indicators that comprise individual cases of malicious insider activity.

Ontology Modeling Prerequisites

Primary Resources

The following information is recommended for those who are trying to model their own data using our ontology. If you are not familiar with modeling, ontologies, and using an ontology to model activities in other domains, be aware that building a model is part science, part rules, and part art. Differences among analysts during language interpretation can affect modeling outcomes. The information in this section explores modeling by providing a brief introduction to ontologies, modeling using ontologies, and modeling using our Insider Threat Ontology. After reviewing the materials, we suggest you practice modeling using your own Insider Threat data.

Manchester Pizza Tutorial

The Manchester Pizza Tutorial [66] is a good introduction to ontologies and modeling with them. This tutorial explains the important basics involved in representing a concept like pizza using the language of ontologies. It specifically uses protégé and OWL, which is the same language we used to build the Insider Threat Indicator Ontology. The tutorial teaches Ontology Construction, OWL, and Ontology Modeling at the same time. At the time of this publication, this tutorial can be found online at: http://dio.freelabs.net/downloads/ProtegeOWLTutorialP4_v1_3.pdf

W3C OWL 2 Web Ontology Language—Structural Specification and Functional-Style Syntax (Second Edition)

This document defines OWL and provides specific examples of its usage in various syntaxes. The W3C also offers other documents to assist ontology developers and modelers. See the OWL 2 Web Ontology Language Document Overview (Second Edition), found on the W3C website (http://www.w3.org/TR/owl2-syntax/).

An Insider Threat Ontology: Development and Applications

Our team published a paper documenting the development and applications of our Insider Threat Indicator Ontology. This paper received the Michael Dean Best Paper Award at the 2014 Semantic Technology for Intelligence, Defense and Security Conference and can be found on the Software Engineering Institute website (http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=426803).

OWL Documentation

The W3C OWL Working Group maintains an overview document for the OWL ontology specification language [67]. This document describes ontologies and their syntax using semantic technologies, most specifically semantic web technology such as RDF and XML.

Schema.org

Schema.org [42] is a collection of schemas that major search providers have collectively agreed to use in search engine technology and is a good reference to explore schemas of many commonly encountered concepts such as ‘actor.’ We used schema.org as an inspiration for modeling many of the common concepts needed to express Insider Threat Indicators. Specifically, our ontology representation of “actors perform actions on objects with instruments” is adapted from schema.org.

Structured Threat Information eXpression (STIX)

STIX is a model that is mainly used for exchanging external threat information. We have reused some terms, ideas, and patterns from STIX, and we have reused the STIX community in our ontology to represent cyber threats to organizational assets. Because of this, it may be a valuable resource in terms of providing ideas for ontology expansions and getting used to modeling threats.

Cyber Observable eXpression (CybOX)

CybOX is an expression standard designed to provide structured representations for cybersecurity observables. Instead of direct translation into ontology individuals from operational data sources, we chose to translate the operational data into CybOX cyber observable files, and automatically create ontology individuals based on the contents of the CybOX files. This approach allowed us to focus on identifying the fields from CybOX that were applicable to our ontology classes and provide a translation mechanism for only those applicable fields. Without the CybOX translation layer, we would have had to develop ontology translation mechanisms for each type of operational data source we wish to support, which would require an infeasible level of effort, support, and maintenance. Additionally, CybOX provides an API for its XML file format, which facilitates the automated translation of any input data source into the CybOX format. (CybOX currently supports over 60 input data sources.)

The Modeling Process

Converting from natural text to an ontology model can be a complex process, involving many interoperating and potentially moving parts. The main goal is for multiple analysts to take a piece of natural language text, process it, and derive the same resulting model as output each time. This seemingly simple problem has proven to be a significant challenge since each analyst has a slightly different interpretation of natural language, and semantic technologies have not yet mastered the nuances of natural text for high-quality fully automated conversion. There are numerous efforts to improve the automated extraction of semantics from natural text, some of which would improve the current state of the art significantly. There are also efforts to automatically label semantics in natural text (e.g., the IARPA metaphor program [68]). However, we are currently using semi-automated human analyst translation for sentence meanings.

To mitigate the effects of different human interpretations of natural text, we developed a set of heuristics and modeling patterns for converting insider threat case description data into our ontology model. When we put the same natural language of insider threat indicators in front of a few of our own analysts, we found that when they followed the Rules of Thumb and the Design Pattern guidance, the resulting Insider Threat Ontology Models were very close.

Figure 22 depicts our model creation process.

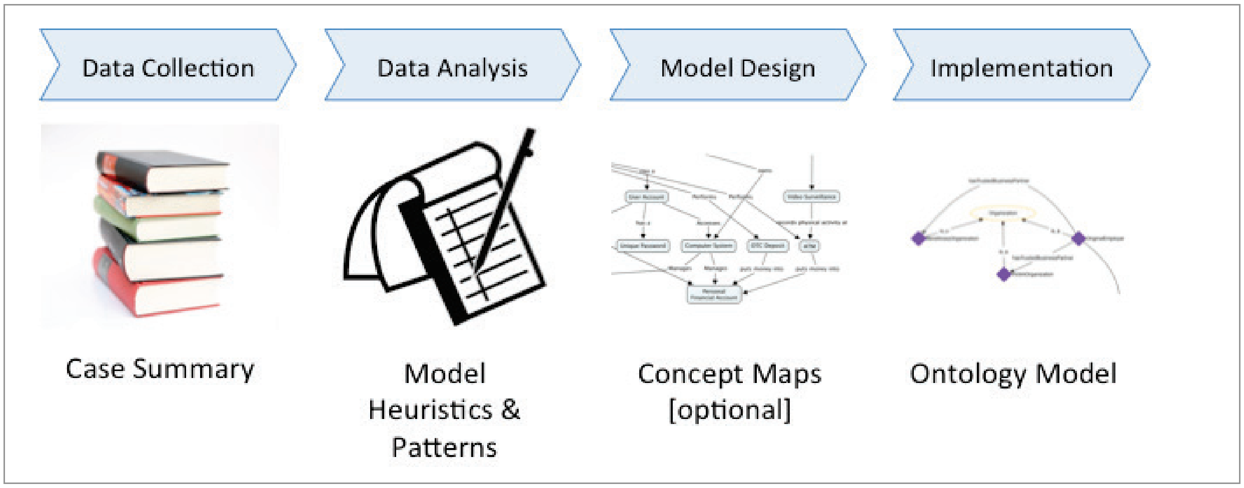


Figure 22: Model-Creation Process

Heuristics and Patterns

We established the following goals as we developed our heuristics and patterns:

- If we create a model from text today, using the following rules, we are able to recreate that model tomorrow.
- There are limited differences in resulting models when separate analysts model the same piece of text.
- There is an explainable way to derive the same model each time.

We created and use the following modeling heuristics to develop a repeatable model from story summary and indicator text data.

- Assume that “actors perform actions on objects with instruments.” Use this phrase when thinking about the meaning of the sentence you are modeling.
- When a term is encountered in a story summary, model it similarly each time (using a pattern if necessary).
- Think of temporality as a sequence rather than an absolute. For example, ‘this’ event happened before ‘that’ one.

- Represent all terms in the present tense. Models are typically always described in the present tense, even when representing chains of dependent events.
- Show explicit relationships only.
- Limit the inferences that cause models to have more description than natural language.
- Keep terms in the singular (avoid using plurals). Use a datatype to indicate amounts of things.
- Do not include actions that an insider could have done but did not do.
- Ignore the impact of the action such as outcome and restitution. Focus on modeling the action and chain of events that make up the attack.
- Include phrases like "insider claimed" where appropriate.

The continued use of a modeling technique to represent a concept is often called a 'pattern.' Patterns can then be reused if the situation is encountered in the future. The following is a short list of patterns that are more specific to the IT and Insider Threat domain. This list will continue to expand as we encounter patterns or hear about them from other teams.

- Actions are performed by actors on objects with instruments. This manifests itself in the ontology via the *hasActor*, *hasObject*, and *hasInstrument* object properties. When modeling actions, we recommend using this pattern wherever possible.
- Computer accounts are owned by people, provide access to IT assets, and contain information such as usernames and passwords. These relationships are expressed in the ontology using the *hasAsset*, *hasAccessTo*, and *hasInformation* object properties.
- Information assets are linked to the information they contain via the *hasInformation* object property.
- Events are often modeled as a series of actions. Actions are performed by actors using some kind of instrument. Actions can be sequenced together using the temporal relations, such as *takesPlaceBefore* and *takesPlaceAfter*.

A certain loss of fidelity is expected when going from natural text to a model. This tradeoff is acceptable for the task we want to accomplish: recognizing repeatable patterns that may indicate future negative events. Natural language is best for describing individual circumstances, but it can be hard to generalize, and once it is generalized, it is hard to analyze.

What We Do Not Model (Yet)

Some of the statements in the case that are not modeled may be relevant in the detection of future insiders, or may be of significance to the individual case in question. We do not dispute these possibilities; rather, we focus on analyzing the data that we currently have—a collection of Insider Threat case—and prioritize the activities that the data shows are frequent and important across many cases. We have also chosen to first model the Insider Threat actions and activities that occur within the IT domain in particular since, based on feedback from our indicator-sharing community, that is the prime area of interest. Unfortunately, this approach means that we do not model the full details of a case or summarize all events. In our current use of the ontology, we create a specific and generally repeatable summary of the Insider Threat IT indicators that are present in a given case.

In the example below, some of the details we omit from the current iteration of our ontology and case models include

- the scope of the damages sustained by the victims
- the investigation into and punishment (if any) of the insider's actions
- the restitution (if any) that was paid by the insider to the victim organization
- the personality characteristics, emotions, or intentions of the individual actors present in the case (This topic is a candidate for future work.)
- vulnerabilities within organizations that enabled the insider to carry out the attacks
- the history of suspicious behavior, or previous insider or criminal activity

This list is not exhaustive, yet it still clearly indicates that there may be domains that we should model to explore the total possible scope of all Insider Threat indicators. We will use this list as a starting point to build future models for ontology domain expansion. The Information Technology domain is just a starting point. Clearly there are many other activities, feelings, and complex situations happening outside this domain that may contribute to a better understanding of Insider Threats and the conditions that, if detectable, could indicate an elevated insider risk condition.

Illustrated Sentence-Modeling Steps

In this section, we describe our process and the steps we use to convert a text sentence from an insider threat case summary into our ontology model. We apply the modeling steps to the sentence fragment "Insider stole credentials."

Summary of Steps

1. Deconstruct sentences and label parts of speech.
2. Use the modeling heuristics and patterns to create individuals and assign classes.
3. Declare relationships (object properties).
4. Enter data and produce graphics.

Step 1: Deconstruct sentences and label parts of speech

The first step to converting natural text to an ontology model is to identify common parts of sentences, focusing on subjects, verbs, and objects. Figure 23 shows our example sentence with the parts of the sentence identified.

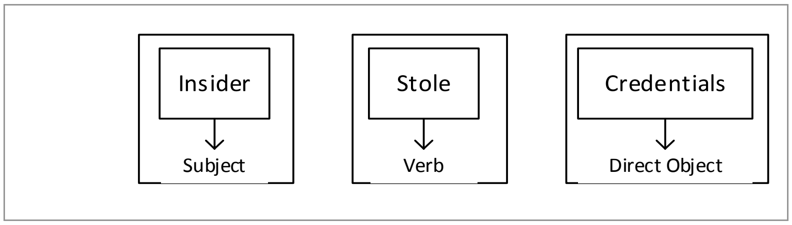


Figure 23: Example Sentence with Labeled Sentence Parts

Sometimes transformations are required to translate the sentence into our ontology while still preserving its original meaning, and sometimes this step requires using substitute terms. For example, the term “stole” may become a *TheftAction* with multiple properties. Typically actions or events were our starting point for expressing a given sentence.

Step 2: Use the modeling heuristics and patterns to create individuals and assign classes

Once the sentence has been deconstructed, use the modeling heuristics and patterns to create individuals for each word. In the example sentence, there is only one individual of each type; however, for larger blocks of text, there may be multiple instances of each individual. Because of this, it is advisable to number all individuals. Figure 24 illustrates this process using our walkthrough example.

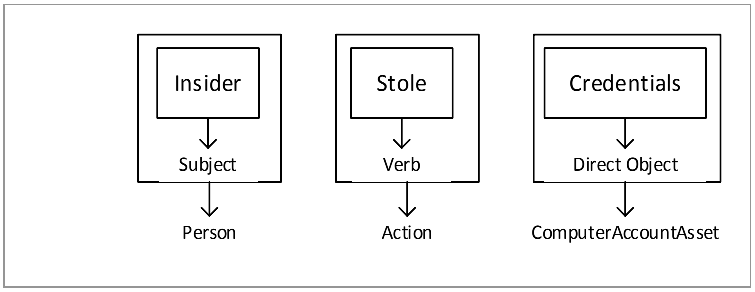
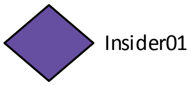
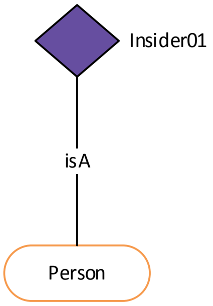


Figure 24: Example Sentence as Ontology Individuals

Step 3: Declare relationships (object properties)

Establish the individuals of the sentence and then identify the class of each individual, as shown in Table 1. In the example, starting with Insider01, we refer to the Insider Threat Ontology to determine that this individual has the class of *Person*, and we likewise determine the class of the remaining two individuals.

Table 1: Ontology Statement Notation and Graphical Representation

Statement	Notation	Graphical Representation
Create individual Insider01	Create individual Insider01	
Insider01 is a member of the Person class	Subclass(insider01,Person)	

Step 4: Enter data and produce graphics

Once each individual has been associated with a class in the Insider Threat Ontology, associate each individual to other corresponding individuals with the appropriate object property. We represent this process with the following pseudo code.

1. Insider
   a. Create an individual for this subject called *Insider01*.
   b. Associate this individual with the appropriate Insider Threat Ontology class, in this example the *Person* class.

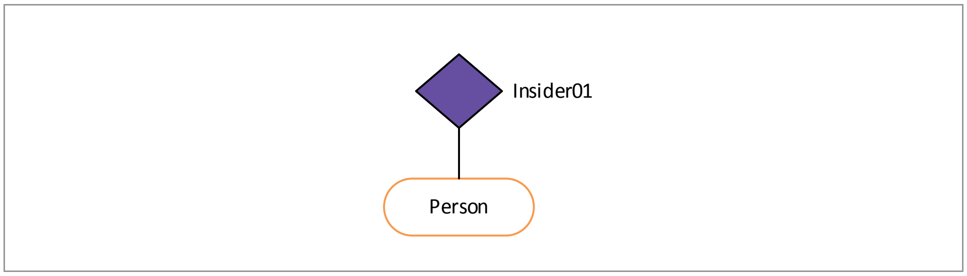


Figure 25: Class Assignment for Insider

2. Stole
   a. Create an individual for this verb called *TheftAction01*.
   b. Associate this individual with the appropriate Insider Threat Ontology class, *Action*.
   c. Associate these individuals with the appropriate Insider Threat Ontology object property, *hasIndividual*.

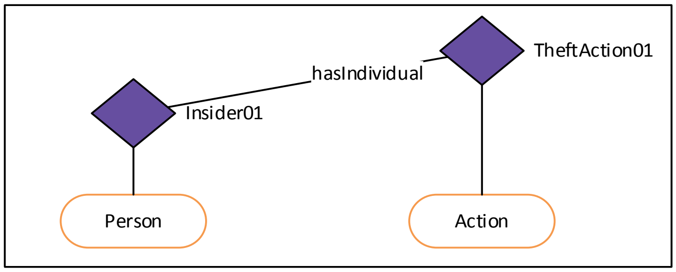


Figure 26: Class Assignment for Stealing

3. Credentials
   a. Create an individual for this direct object, *Credential01*.
   b. Associate this individual with the appropriate Insider Threat Ontology class, *ComputerAccountAsset*.
   c. Associate these individuals with the appropriate Insider Threat Ontology object property, *hasObject*.

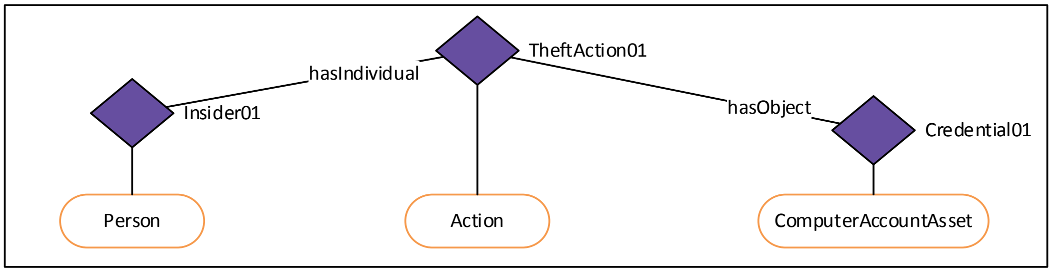


Figure 27: Class Assignment for Credentials

Converting Text into a Model

Our ontology is limited to a set of actions and focuses on those actions taken by insiders during the attack. We do not yet model employment history or other behavioral factors. In this section, we take sample summary of case data and outline the steps to create a model of insider threat indicators using the CERT Insider Threat Indicator Ontology.[1] The case summary follows.

> *The insider was originally employed as a switch design engineer and was later promoted to product design manager by an organization that sold computer networking products. The insider sought a new position with two of his employer's trusted business partner (TBP) organizations. The insider rejected an offer from one of the TBPs, the victim organization, and accepted an offer from the other TBP, the beneficiary organization. Subsequently, the insider announced his resignation. As a TBP, the insider's original employer had controlled access to the victim organization's trade secret information, which was maintained on its extranet for customers' access. In the month prior to leaving his original employer, the insider used this access to download the victim organization's trade secret files. The insider downloaded the trade secrets on at least three occasions while on-site and during work hours. Two days after starting his new job with the beneficiary organization, the insider loaded the victim organization's trade secret files onto his company-assigned laptop. A month later, the insider emailed the trade secret files to other employees at the beneficiary organization, which led to detection of the incident. The victim organization sustained substantial, unspecified economic loss due to the disclosure of its trade secrets to the beneficiary organization. The duration of the incident was three months.*

We use this summary to illustrate how to model using the CERT Insider Threat Indicator Ontology, and we follow a common diagram format. A purple diamond represents individuals, and an orange circle represents classes. Relationships between individuals are represented by directed arrows. The Protégé tool developed and maintained by Stanford University also displays individuals and classes in this way as they are entered into an ontology.

[1] We make no claims that the behavior depicted in the model is always an insider threat. This case is a specific example of a specific organization at a specific time, when some of the described actions were viewed by the organization as inappropriate.

Identify the Main Actors

Most of our stories begin from a common pattern template. Our cases usually contain a series of malicious actions performed by an actor. The actor has a type of employment relationship with one or many organizations. This process is repeated for each actor—remember that a specific organization is also a subclass of *Actor*.

Before we can begin modeling events, we need to create a base of the characters involved in our summary. We start by looking for specific actors in the text and then create individual actors. We apply this technique to our summary text.

Summary Text with Actors Highlighted

> *The insider was originally employed as a switch design engineer and was later promoted to product design manager by an organization that sold computer networking products. The insider sought a new position with two of his employer's trusted business partner (TBP) organizations. The insider rejected an offer from one of the TBPs, the victim organization, and accepted an offer from the other TBP, the beneficiary organization.…*

Modeling Actions

In the first sentence of this example, there are four actors: the insider, an organization that sold computer networking parts (the original employer), and two trusted business partners (one becomes the beneficiary organization, and the other becomes the eventual victim organization).

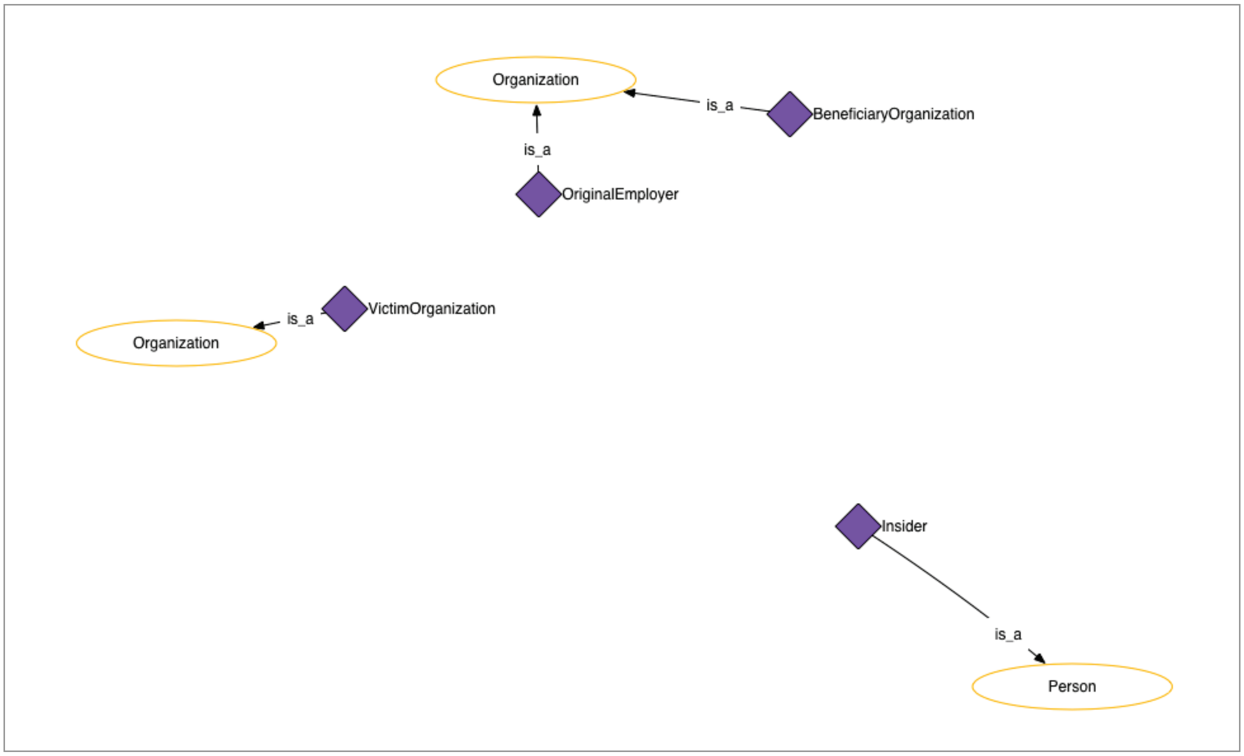


Figure 28: Main Actors

Add Relationships

The next task is to create the specified relationships among the four actors. Employee job titles are modeled as a relationship between an organization and an employee. We name the individuals according to terms in the text such as *BeneficiaryOrganization* to model the recipient of the insider's later illicit activities, even though the organizations do not always benefit in the long term. The name is simply a label we place on the individual word.

Summary Text with Relationships Highlighted

> *The insider was originally employed as a switch design engineer and was later promoted to product design manager by an organization that sold computer networking products. The insider sought a new position with two of his employer's trusted business partner (TBP) organizations. The insider rejected an offer from one of the TBPs, the victim organization, and accepted an offer from the other TBP, the beneficiary organization.…*

Modeling Actions

We identify relationships between the actors in the summary and add them to the model. Relationships link one individual to another and are bi-directional, unless specifically restricted as a one-way relationship. The inverse relationships are not shown in these diagrams for visual clarity.

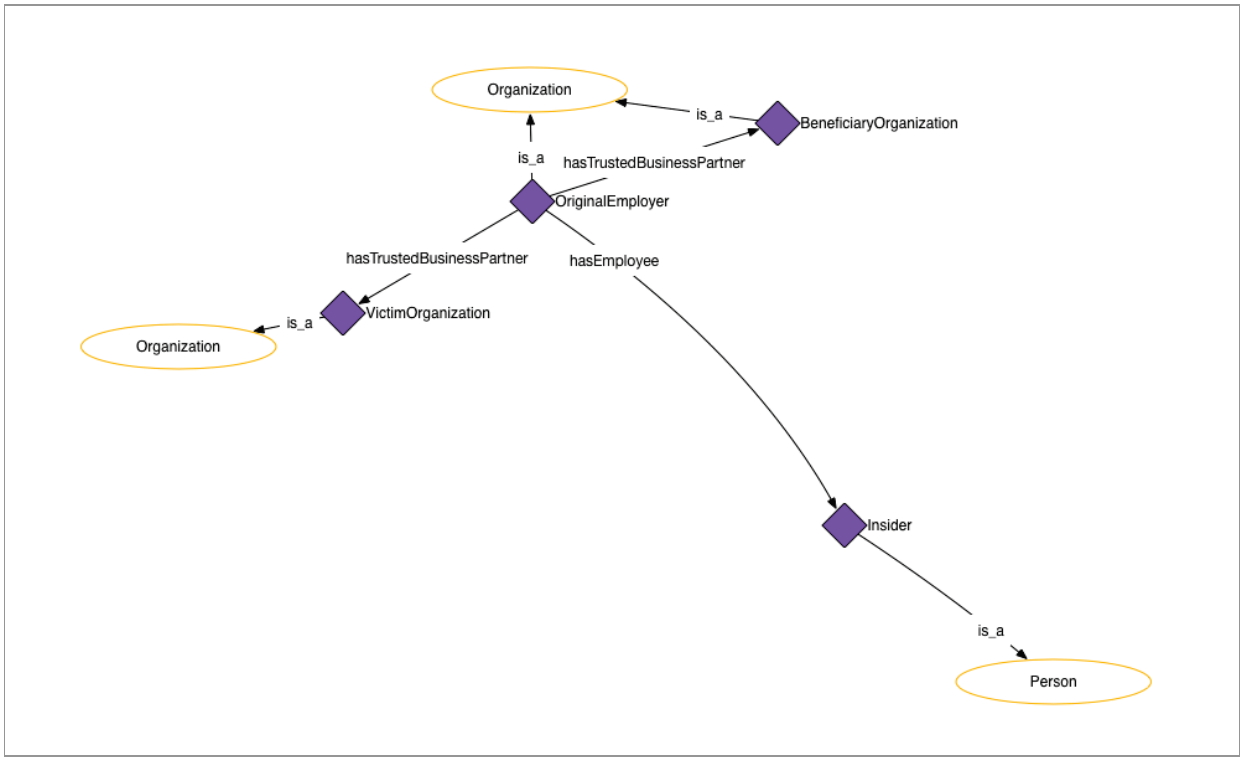


Figure 29: Relationships Between Main Actors

Model IT Infrastructure

We then identify the IT infrastructure that the insider uses to perform the activity. Our ontology is focused on describing IT actions; therefore, we have limited our initial scope to describe common IT infrastructure according to the cases we have collected so far.

Summary Text (with IT Infrastructure Highlighted)

> *…The insider's original employer had controlled access to the victim organization's trade secret information, which was maintained on its extranet for customers' access.…*

Modeling Actions

We add the extranet asset and the important trade secret information it contains. We also add the computer account that allowed controlled access to the extranet.

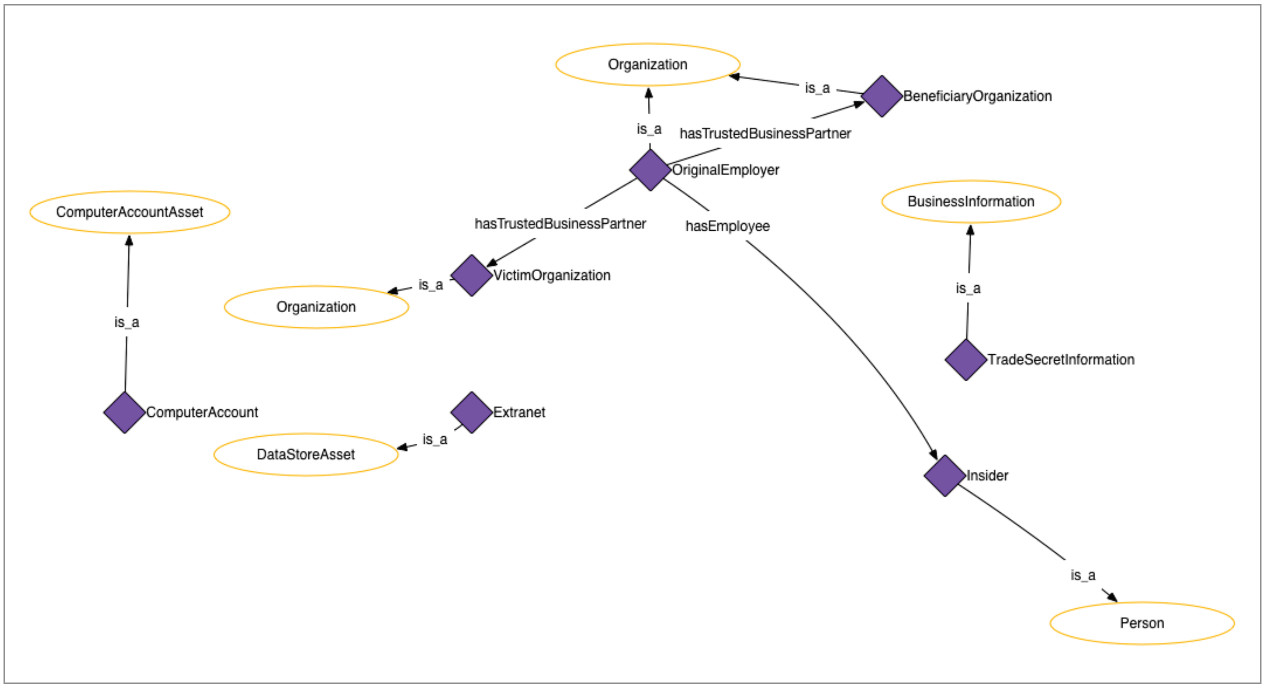


Figure 30: Addition of IT Infrastructure

Connect Infrastructure to Actors

After the IT infrastructure has been modeled, we connect it to the actors in the case. These connections are sometimes not specifically stated, so it is important to have domain knowledge of how different pieces of IT infrastructure are used by actors. It is often important to consider the ownership relationships of the assets. For example, it may be important to know whether the smart phone belongs to the employee or the organization.

Summary Text (Connections Highlighted)

> *…The insider’s original employer had controlled access to the victim organization’s trade secret information, which was maintained on its extranet.…*

The text does not always explicitly state all of the particulars of how the actors are connected to the IT infrastructure. Because of this, some human inference and background knowledge of how IT systems operate are required to make these connections.

Modeling Actions

We connect actors to their computer accounts, which are the accounts that can access the infrastructure. We state that the *ComputerAccount* is owned by the original employer and it can access the extranet. We state the extranet is an asset owned by the victim organization.

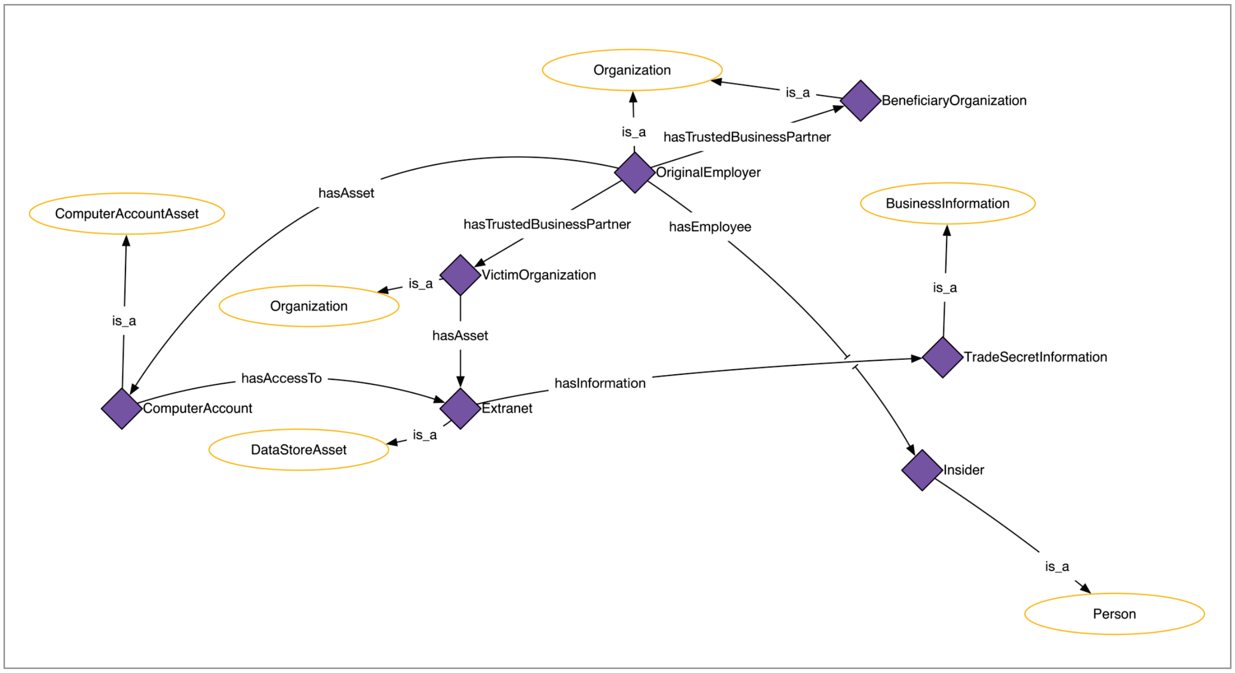


Figure 31: Connecting IT Infrastructure to Actors

Important IT Actions

We are now ready for the first insider IT action. This section outlines how the team models the first action using the ontology.

Summary Text (with IT Action Highlighted)

> *…The insider used this access to download the victim organization’s trade secret files. The insider downloaded the trade secrets on at least three occasions while on-site and during work hours.…*

Modeling Decisions

We represent the file download as a *CopyAction* in the ontology. In the next sentence, we identify another computer, so we add another computer asset to our model.

Since the download happened on three occasions, we would normally model three separate copy actions. However in the example below, for demonstration purposes, we show a single copy action.

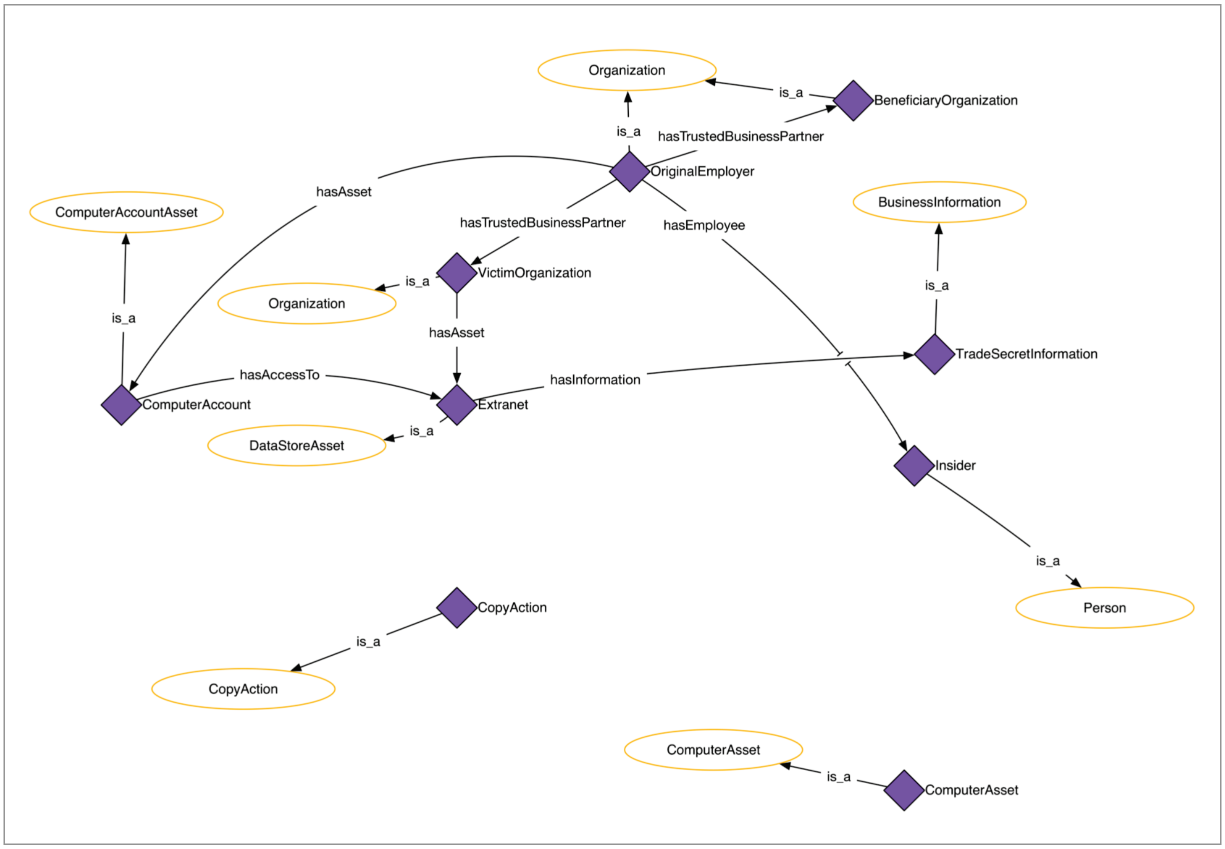


Figure 32: Addition of IT Actions

Expanding the Description of the *CopyAction*

Each *CopyAction* has an instrument (which we define as something important used to facilitate/perform the action), a source (where the object started), a destination (where the object ended up), an object (of the copy action), and an actor (the person doing the copying).

We may not have a complete set of values or answers for every copy action, but when we do, we can model them using the above structure. When we do not have complete information, the model can still represent a copy action even if it does not know who did the copying or even what was copied. This design takes advantage of the open world assumption, which is one of the assumptions upon which the ontology model system is based.[2]

To model the copy action, we take a close look at its particulars. From this part of the sentence *…the insider used this access to download the victim organization's trade secret files…*, we can identify the following

- The instrument of the *CopyAction* is the *ComputerAccount*.
- The source of the *CopyAction* is the *Extranet*.
- The destination of the *CopyAction* is a *ComputerAsset*.
- The object of the *CopyAction* is *TradeSecretInformation*.
- The actor who performed the *CopyAction* is the Insider.

Figure 33 depicts the diagram after the *CopyAction* properties (relationships) are modeled.

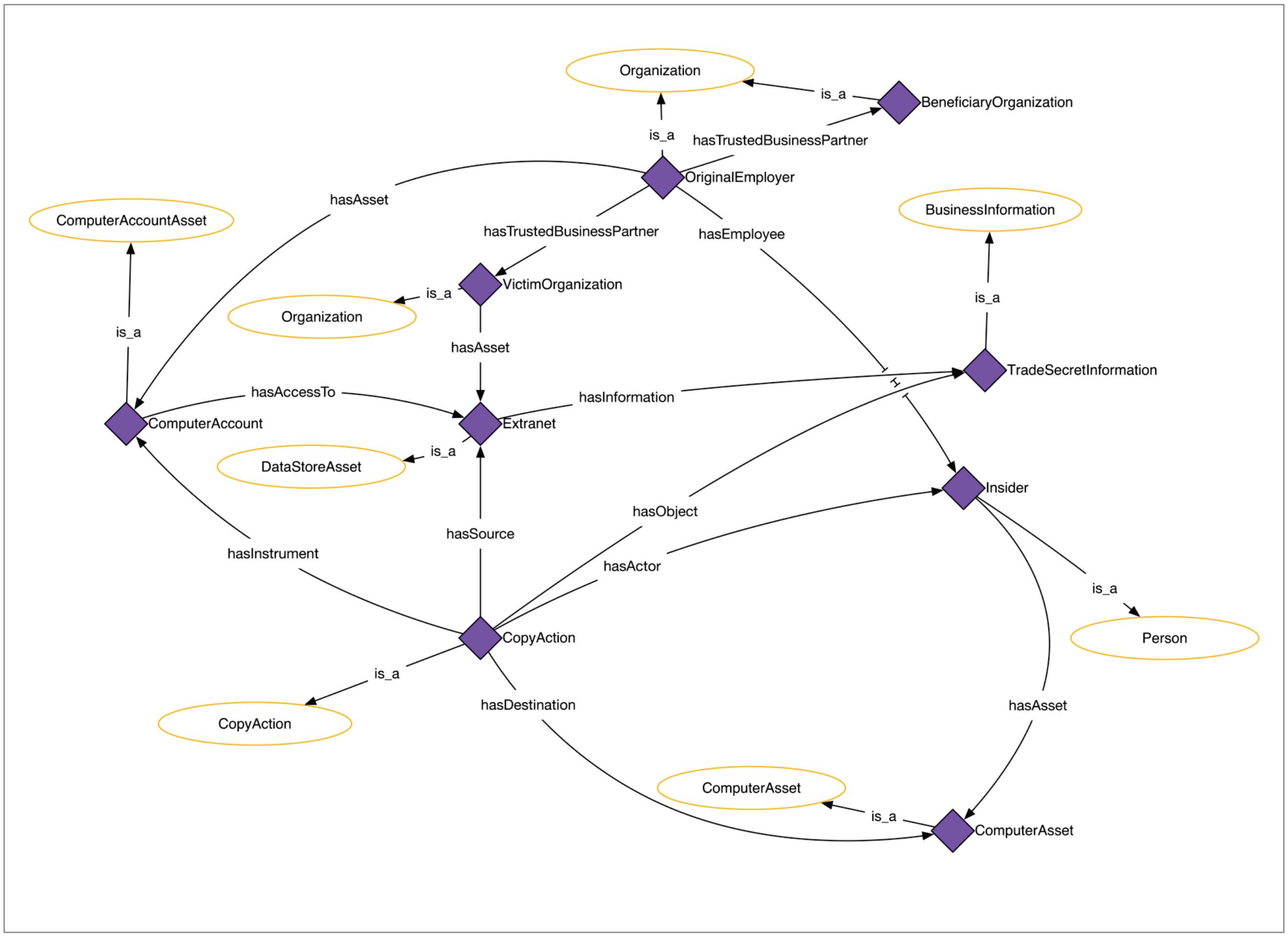


Figure 33: Describing the CopyAction

[2] See the Dataversity website for more information (http://semanticweb.com/introduction-to-open-world-assumption-vs-closed-world-assumption_b33688).

Employment Change Actions

The CERT Insider Threat Ontology is focused on describing cyber indicators; however, the domain of Insider Threat is much larger. Simply emailing trade secret information to a colleague may not raise flags inside a company, but this act becomes an indicator when the recipient works for a competitor. Rather than modeling the universe of all possible actions, we focus on actions that frequently occur within our case data. Actions that involve insiders changing employers are one such example.

In this section, we show a model for job change action. We are currently working on modeling a prioritized list of other non-IT focus actions.

Summary Text

> *…The insider sought a new position with two of his employer's trusted business partner (TBP) organizations. The insider rejected an offer from one of the TBPs, the victim organization, and accepted an offer from the other TBP, the beneficiary organization. Subsequently, the insider announced his resignation.…*

Modeling Decisions

We model the insider accepting a job offer from the beneficiary organization as a *JobChangeAction*. The information the insider stole is maintained on his own *ComputerAsset*, which the insider brought to the beneficiary organization.

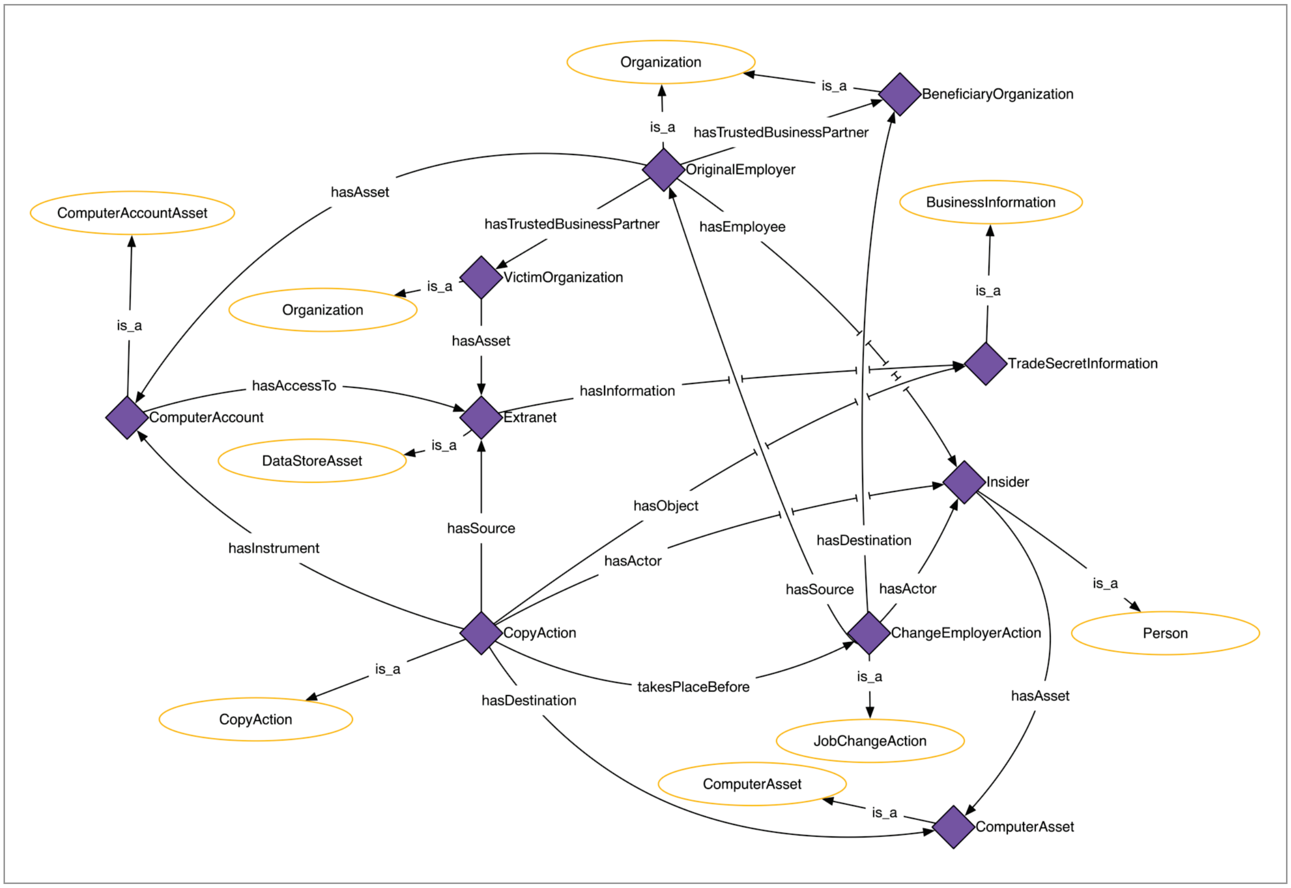


Figure 34: Job Change Action

Moving Trade Secrets

After the copy action, we move to the next insider action related to the movement of trade secrets off the company's network.

Summary Text

> ...*Two days after starting his new job with the beneficiary organization, the insider loaded the victim organization's trade secret files onto his company-assigned laptop*....

Modeling Actions

Figure 35 includes the act of loading trade secret files onto a laptop.

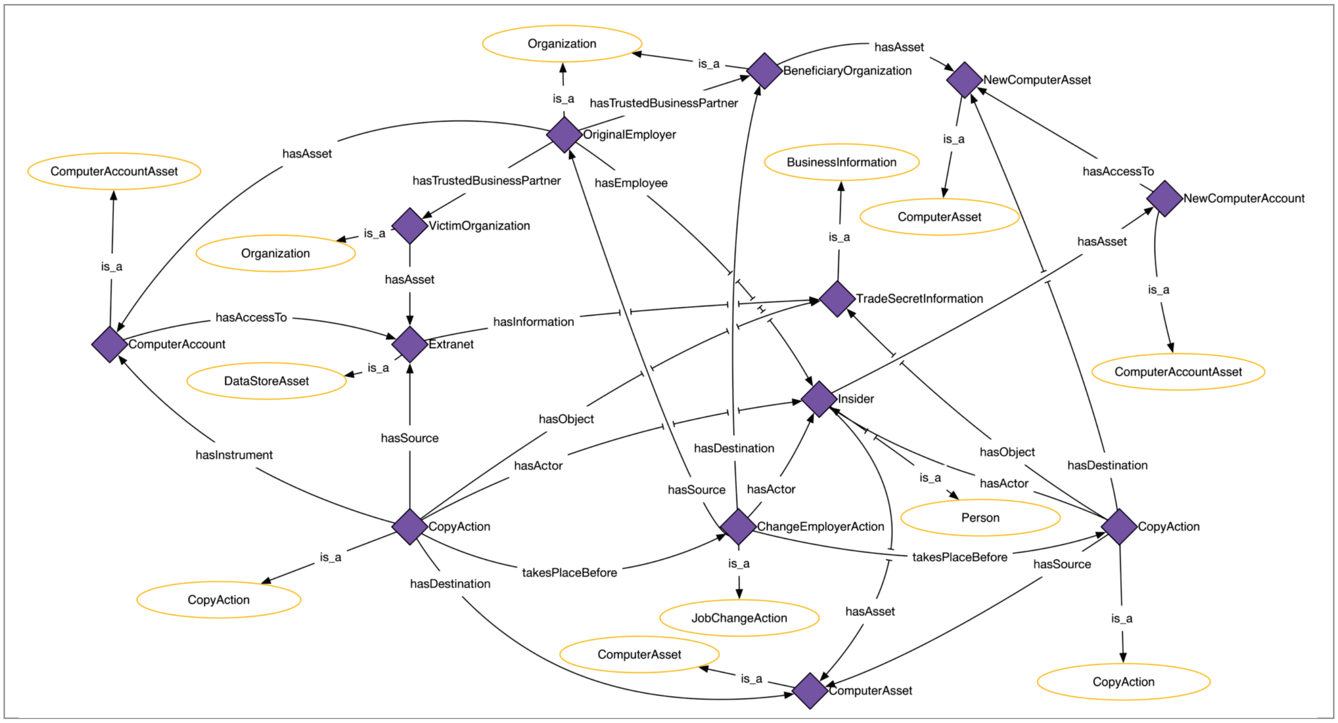


Figure 35: Moving Trade Secrets

Emailing Trade Secrets at the New Organization

Finally, we model the third insider action: emailing the trade secret information while at his new company.

Summary Text

> *…A month later, the insider emailed the trade secret files to other employees at the beneficiary organization, which led to detection of the incident.…*

Modeling Actions

Figure 36 includes an action of emailing of trade secret files to others at the new organization.

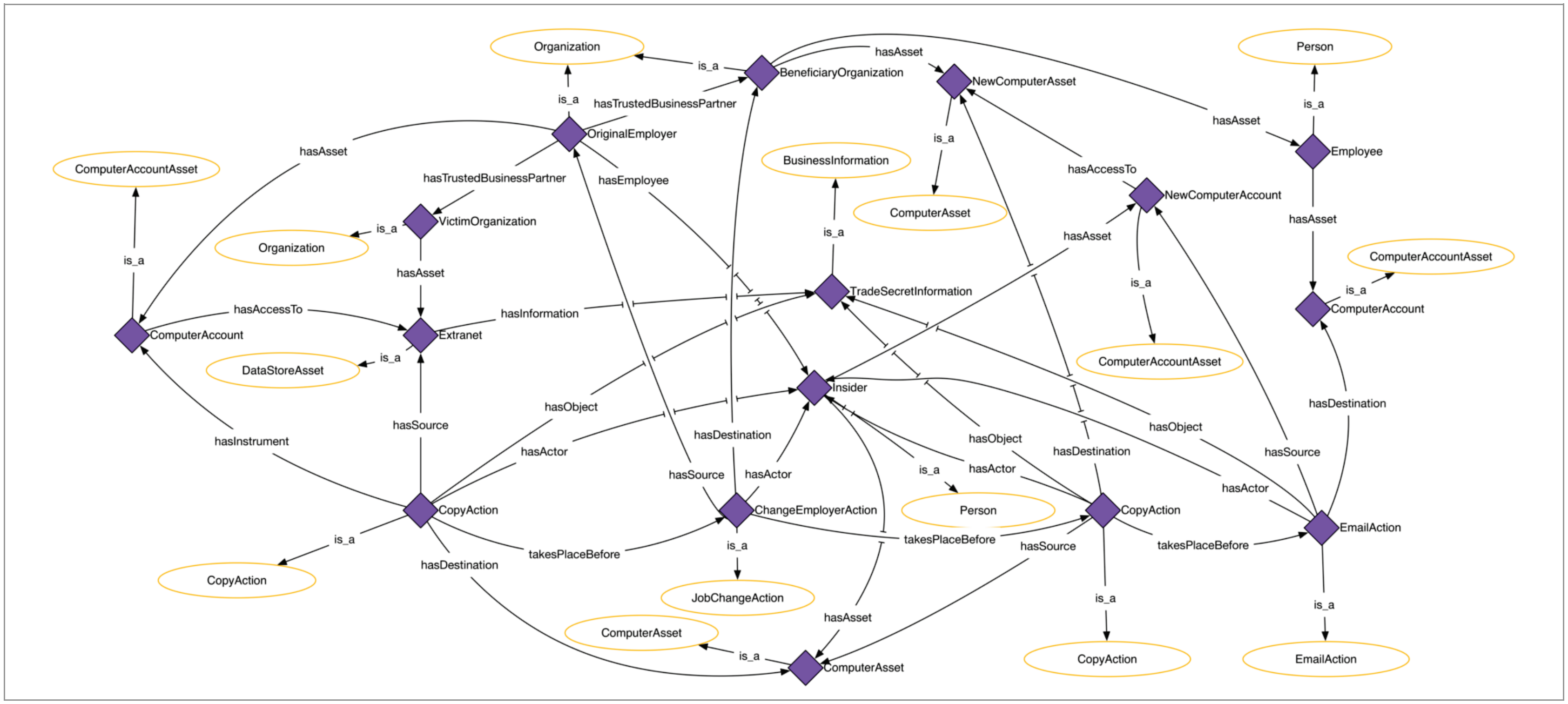


Figure 36: Emailing Trade Secrets to Beneficiary

Final Diagram and Conclusions

Figure 36, the final diagram in the process, depicts what started as a simple activity model that quickly turned into a complex series of relationships. The ontology is a way to formalize and represent these relationships in a way that can be exchanged and computed upon.

Now that we have presented the ontology using an actual insider threat case description, you should be able to

- Apply the modeling heuristics and patterns presented in this user's manual to your own organization's insider threat case data.
- Develop your own modeling patterns to apply uniformly across that data.
- Discover the benefits of semantic models and semantic reasoning against your organization's insider threat case data.
- Share your threat and case information with the insider threat community using a controlled vocabulary and standardized model.

We hope you will try the ontology, modeling techniques, and controlled vocabulary presented in this guide on your own case data. We are also interested in hearing about your experiences using this ontology. In particular, we would like to learn

- whether or not you were able to model the actions of insiders using your own case data
- what patterns you used or did not use
- whether you extended the ontology and what domains you covered
- whether you use a different term or definition than the one provided by this ontology

We welcome your feedback, questions, and comments. Contact us at insider-threat-feedback@cert.org.

Appendix B: Ontology Specification

Classes

Table 2 presents the classes of the ontology. For each class, the parent name, class definition, and source of the class definition (where applicable) are provided.

Table 2: Ontology Class Hierarchy Specification

Name	Parent Class	Definition	Definition Reference
AllTime	TemporalInterval	An instance that represents the infinite interval containing all time	None
Date	TemporalInterval	None	None
Month	TemporalInterval	None	None
TimePoint	TemporalInterval	Instances of maximally small intervals—time pixels if you will (This size of a pixel corresponds with the smallest resolvable time unit on the machine implementation in use. Time points are to be shared among all events in the data store.)	None
Year	TemporalInterval	None	None
AcceptAction	JobChangeAction	To agree to start or change to a specific job role	None
AccessAction	DigitalAction	To gain access to a system	None
AccountAuthenticationInformation	SystemInformation	Information used to identify and authenticate a person on a computer or network	None
Action	TemporalThing	A thing performed by a direct actor and indirect participants on a direct object, which may produce a result (optionally happens at a location and/or with the help of an instrument)	None
ActionModifier	Action	Modifier that describes additional subjective details about an action	None
Actor	None	The direct performer or driver of an action	http://schema.org/agent
AnomalousAction	ActionModifier	Action determined to deviate from a set baseline	None
ApplyAction	JobChangeAction	To submit an application for a job	None
Asset	None	A utility class that serves as the umbrella for a number of tangible and intangible things, such as data, hardware, personally identifiable information (PII), software, etc.	http://schema.org/Intangible
BackdoorSoftwareAsset	MalwareAsset	A computer program designed to allow an unauthorized path into the network or a system	None
BackupTapes Asset	PhysicalAsset	A reserve copy of data, stored on magnetic tape media, for use if the original becomes lost or damaged	None

Name	Parent Class	Definition	Definition Reference
BankAccountAsset	FinancialAsset	An arrangement by which an organization accepts a customer's financial assets and holds them on behalf of the customer at his or her discretion	http://www.investopedia.com/terms/a/account.asp
BankAccountInformation	FinancialInformation	The information uniquely identifying a bank account, including account numbers and balance information	None
BreachAction	JobChangeAction	To not uphold or violate the terms of a contract or agreement	http://www.shrm.org/templatestools/glossaries/hrterms/pages/b.aspx
BusinessInformation	Information	Information about how a business is run	None
BusinessPolicyInformation	BusinessInformation	Information contained in business policies (Policies provide high-level criteria for developing business processes.)	None
BusinessProcessInformation	BusinessInformation	Information on how business processes are performed	None
ClassifiedInformation	Information	Sensitive information that requires special protections	None
CompactDiskAsset	PhysicalAsset	A polycarbonate with one or more metal layers capable of storing digital information	http://www.webopedia.com/TERM/C/compact_disc.html
CompressAction	ModificationAction	The reduction in the size of data to save space or transmission time	http://searchstorage.techtarget.com/definition/compression
ComputerAccountAsset	DigitalAsset	A means of authenticating and auditing computer access to a network or domain resources	http://msdn.microsoft.com/en-us/library/cc759279
ComputerAsset	PhysicalAsset	An electronic device (or system of devices) that is used to store, manipulate, and communicate information; perform complex calculations; or control or regulate other devices or machines and is capable of receiving information (data) and of processing it in accordance with variable procedural instructions (programs or software)	http://www.oed.com/view/Entry/37975
ConnectAction	AccessAction	To establish a communications connection	http://www.merriam-webster.com/dictionary/connect
CopyAction	DigitalAction	To duplicate an original item that you can then modify, delete, or store independent of the original	http://windows.microsoft.com/en-us/windows-vista/copy-a-file-or-folder
CreateAction	ModificationAction	To cause an asset to exist	http://www.merriam-webster.com/dictionary/create
CreditCardAsset	FinancialAsset	A card issued by a financial company giving the holder an option to borrow funds, usually at a point of sale	http://www.investopedia.com/terms/c/creditcard.asp
CreditCardNumber	FinancialInformation	A unique number identifying a credit card account	None
CreditReportInformation	FinancialInformation	Information regarding an individual's history of borrowing money	None
DataDeletionEvent	Event	The logical, but not necessarily physical, erasure of data from an operating system	http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

Name	Parent Class	Definition	Definition Reference
DataExfiltrationEvent	Event	The unauthorized transfer of data	http://whatis.techtarget.com/definition/data-exfiltration-data-extrusion
DataModificationEvent	Event	The act or process of changing parts of data	http://www.merriam-webster.com/dictionary/modification
DataRecordAsset	DigitalAsset	A unit of data that can be held in a file or datastore	None
DataStoreAsset	DigitalAsset	A collection of information that is organized so that it can easily be accessed, managed, and updated	http://searchsqlserver.techtarget.com/definition/database
DecompressAction	ModificationAction	To expand a compressed file back into its original form	http://whatis.techtarget.com/definition/uncompressing-or-decompressing
DecryptAction	ModificationAction	To cryptographically restore cipher text to the plaintext form it had before encryption	None
DeleteAction	ModificationAction	To remove something, such as words, pictures, or computer files, from a document, recording, computer, etc.	http://www.merriam-webster.com/dictionary/delete
DemoteAction	JobChangeAction	A permanent reassignment to a position with a lower pay grade, skill requirement, or level of responsibility than the employee’s current position	http://www.shrm.org/TemplatesTools/Glossaries/HRTerms/Pages/d.aspx
DepositAction	FinancialTransactionAction	To add money to a customer’s bank account	https://www.bankofamerica.com/deposits/manage/glossary.go#alp-D
DigitalAction	Action	An action involving digital assets	None
DigitalAsset	Asset	An asset in the digital realm	None
DisableAction	ModificationAction	To cause an asset to be unable to work in the normal way	http://www.merriam-webster.com/dictionary/disable
DriversLicenseNumber	UniquelyIdentifiable Information	Unique number that identifies a person's driver’s license	None
EmailAction	DigitalAction	To send an email	None
EncryptAction	ModificationAction	To cryptographically transform data to produce cipher text	None
Event	TemporalThing	Defined class that includes one or more actions	None
ExcessiveAction	ActionModifier	Action performed in excess of an organization-defined threshold for normal activity	None
FileAsset	DigitalAsset	A complete collection of data (as text or a program) treated by a computer as a unit, especially for purposes of input and output	http://www.merriam-webster.com/dictionary/file
FinancialAsset	Asset	An asset involving money	None
FinancialInformation	Information	Information about financial assets	None
FinancialTransactionAction	Action	A transaction involving the movement of money	None
FirewallAsset	SoftwareAsset	A system designed to prevent unauthorized connections to or from a private network	http://www.webopedia.com/TERM/F/firewall.html

Name	Parent Class	Definition	Definition Reference
FloppyDiskAsset	PhysicalAsset	A disk storage medium composed of a disk of thin and flexible magnetic storage medium, sealed in a rectangular plastic carrier lined with fabric that removes dust particles	http://en.wikipedia.org/wiki/Floppy_disk
FraudEvent	Event	Intentional perversion of truth to induce another to part with something of value or to surrender a legal right	http://www.merriam-webster.com/dictionary/fraud
FraudulentAction	ActionModifier	A deliberately deceptive action (Examples include forging signatures on documents, IP or MAC address spoofing, or falsifying PII.)	None
HardDriveAsset	PhysicalAsset	A high-capacity, self-contained storage device containing a read-write mechanism together with one or more hard disks inside a sealed unit	http://www.oed.com/view/Entry/84122
IPAddress	NetworkInformation	Address identifying a computer on a network	None
IllegitimateAction	ActionModifier	An action that is not performed legitimately	None
Information	None	A representation of data	None
InstallAction	DigitalAction	The act of making a program ready for execution	http://en.wikipedia.org/wiki/Installation_(computer_programs)
IntellectualProperty	BusinessInformation	Information about assets owned by the organization	None
JobChangeAction	Action	To change roles or positions at one's current employer or to begin a position with a new employer	None
JobFunctionChangeEvent	Event	Event where an individual's job function changes	None
JobOfferAction	JobChangeAction	To offer a job to a potential employee	None
JobOfferEvent	Event	Event where an individual is offered employment	None
KeyLoggerAsset	SoftwareAsset	A type of surveillance software that has the capability to record every keystroke made to a log file, which is usually encrypted	http://www.webopedia.com/TERM/K/keylogger.html
LaptopAsset	PhysicalAsset	A portable computer small enough to sit on your lap	http://www.webopedia.com/TERM/L/laptop_computer.html
LoanAction	FinancialTransactionAction	To give money, property, or other material goods to another party in exchange for future repayment of the principal amount along with interest or other finance charges	http://www.investopedia.com/terms/l/loan.asp
LogicBombAsset	MalwareAsset	A malicious program that is coded to execute when a certain set of requirements are met	None
LoginAction	AccessAction	The process of presenting an identity (typically a user ID) and authentication (a password, token, or other item) to gain access to information systems and resources	https://definedterm.com/login

Name	Parent Class	Definition	Definition Reference
MACAddress	NetworkInformation	Unique address identifying a piece of networked hardware	None
MaliciousCodeInformation	SourceCodeInformation	Source code for a piece of software that performs malicious actions	None
MalwareAsset	SoftwareAsset	A malicious piece of software	None
MasqueradingEvent	Event	Where a system entity illegitimately poses as (assumes the identity of) another entity	http://www.sans.org/security-resources/glossary-of-terms/?pass=m (adapted)
MedicalInformation	Information	Information on an individual's medical history	None
ModificationAction	DigitalAction	To change a file or system	None
MoneyAsset	FinancialAsset	An officially issued legal tender generally consisting of currency and coin (Money is the circulating medium of exchange as defined by a government.)	http://www.investopedia.com/terms/m/money.asp
NationalSecurityInformation	Information	Information classified by a government as having the potential to cause harm to national security when in the wrong hands	None
NetworkAsset	PhysicalAsset	A collection of computers and other hardware components interconnected by communication channels that allow sharing of resources and information	http://en.wikipedia.org/wiki/Computer_network
NetworkInformation	TechnologyInformation	Information identifying a computer or device on a network	None
Organization	Actor	An organized body of people with a particular purpose, such as a business, government department, charity, etc.	http://www.oed.com/view/Entry/132452
PasswordCrackerAsset	SoftwareAsset	A program that is used to identify an unknown or forgotten password to a computer or network resource	http://searchfinancialsecurity.techtarget.com/definition/password-cracker
PasswordInformation	AccountAuthenticationInformation	Secret used for authentication of a computer account	None
Person	Actor	A human being	None
PhysicalAsset	Asset	An asset in the physical realm	None
PortScannerAsset	SoftwareAsset	A software program that scans a network for systems with open ports	None
PrintAction	DigitalAction	To send a unit of work to a printer to create a physical representation of digital data on physical media, usually paper	None
PrinterAsset	PhysicalAsset	A device that accepts text and graphic output from a computer and transfers the information to paper	http://whatis.techtarget.com/definition/printer
PromoteAction	JobChangeAction	Career advancement within an organization, which includes increased authority, level of responsibility, status, and pay	http://www.shrm.org/templatestools/glossaries/hrterms/pages/p.aspx

Name	Parent Class	Definition	Definition Reference
ReassignAction	JobChangeAction	To transfer individuals to alternative positions where their talents or skills may be best utilized to their own or the organization’s benefit or where they are better able to perform the job in accordance with required standards	http://www.shrm.org/TemplatesTools/Glossaries/HRTerms/Pages/r.aspx
RecruitmentEvent	Event	To solicit and actively seek applicants to fill recently vacated or newly created positions using a variety of methods	http://www.shrm.org/TemplatesTools/Glossaries/HRTerms/Pages/r.aspx
RejectAction	JobChangeAction	To disagree to start or change to a specific job role	None
ReprimandAction	JobChangeAction	An oral or written reproach given to an employee as part of a disciplinary action	http://www.shrm.org/TemplatesTools/Glossaries/HRTerms/Pages/r.aspx
ResignationAction	JobChangeAction	To terminate one's employment	None
SDCardAsset	PhysicalAsset	A tiny memory card used to make storage portable among various devices (An SD card is about the size of a postage stamp and weighs approximately two grams.)	http://searchstorage.techtarget.com/definition/Secure-Digital-card
SabotageEvent	Event	To deliberately destroy, damage, or obstruct	http://www.oed.com/view/Entry/169373 (adapted)
SearchAction	DigitalAction	To peruse, look through, examine (writings, records) to discover whether certain things are contained there	http://www.oed.com/view/Entry/174308
ServerAsset	PhysicalAsset	A system entity that provides a service in response to requests from other system entities called clients	http://www.sans.org/security-resources/glossary-of-terms/?pass=s
ServiceAsset	SoftwareAsset	A piece of software that runs in the background on a computer	None
SocialSecurityNumber	UniquelyIdentifiableInformation	Unique number assigned by the federal government that uniquely identifies an individual	None
SoftwareAsset	DigitalAsset	The programs and procedures required to enable a computer to perform a specific task, as opposed to the physical components of the system	http://www.oed.com/view/Entry/183938
SourceCodeInformation	TechnologyInformation	The code from which a piece of software is compiled	None
SuspendAction	JobChangeAction	To prohibit an individual from holding his or her usual post or carrying out his or her usual role for a particular length of time	None
SuspiciousAction	ActionModifier	Action that falls under organization-defined criteria for being potentially malicious	None
SystemConfigurationInformation	SystemInformation	Settings that specify how a system operates	None
SystemInformation	TechnologyInformation	Information about a computer or computer account	None

Name	Parent Class	Definition	Definition Reference
SystemModificationEvent	Event	A software or hardware configuration change	None
TechnologyInformation	Information	Information about or involving technology	None
TemporalInterval	TemporalThing	None	None
TemporalThing	None	None	None
TerminateAction	JobChangeAction	Separation from employment due to a voluntary resignation, layoff, retirement, or dismissal	http://www.shrm.org/templatestools/glossaries/hrterms/pages/t.aspx
TheftEvent	Event	To take something that does not belong to you in a way that is wrong or illegal	http://www.merriam-webster.com/dictionary/theft http://www.merriam-webster.com/dictionary/stealing
TradeSecretInformation	BusinessInformation	Information that is kept secret by the organization and is intended to provide some competitive advantage	None
USBDriveAsset	PhysicalAsset	A plug-and-play portable storage device that uses flash memory and is lightweight enough to attach to a key chain	http://searchstorage.techtarget.com/definition/USB-drive
UnauthorizedAction	ActionModifier	Action that was not authorized by the organization or data/system owner	None
UniquelyIdentifiableInformation	Information	Information that is unique to an individual	None
UsernameInformation	AccountAuthenticationInformation	Identifier for a user's computer account	None
VirtualMachineAsset	SoftwareAsset	A software implementation of a computing environment in which an operating system (OS) or program can be installed and run	http://searchservervirtualization.techtarget.com/definition/virtual-machine
VirusAsset	MalwareAsset	A program that is capable of replicating itself and has malicious purposes	None
WithdrawAction	FinancialTransactionAction	To remove funds from an account	https://www.bankofamerica.com/deposits/manage/glossary.go#alp-D

Object Properties

Table 3 presents the object properties of the ontology. For each object property, the parent property, description, domain, range, and inverse are provided. For object properties whose definitions are listed as "See Inverse," please refer to the definition of the inverse property.

Table 3: Ontology Object Property Hierarchy Specification

Name	Parent Property	Definition	Domain	Range	Inverse
endMeetsBeginningOf	temporallyRelatedTo	This property links a temporal thing that follows immediately after a second temporal thing. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	http://semanic.org/OntDef/SpaceTime#startMeetsEndingOf
startMeetsEndingOf	temporallyRelatedTo	This predicate means that the TemporalThing subject starts immediately following the TemporalThing object. subject and object have no time points in common, but there is also no time point between the ending of object and the starting of subject. Derived from OpenCyc 1.0. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	None
cotemporalWith	temporallyRelatedTo	This property means that the two temporal things have precisely the same temporal extent (see temporalExtent). Derived from OpenCyc 1.0 (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	cotemporalWith
finishedBy	temporallyRelatedTo	See inverse. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	finishes
finishes	temporallyRelatedTo	This predicate means that subject and object end at the same time and that the subject starts after the object. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	None
hasAccessTo	None	This describes a computer account's access to an asset.	ComputerAccountAsset	Asset	None
hasAccomplice	hasEventRelation	This relation defines an accomplice to the insider during the event. An accomplice is a person who helps another commit a crime	Person	Person	isAccompliceOf

Name	Parent Property	Definition	Domain	Range	Inverse
hasAction	None	This property links an event to its action(s).	Event	Action	None
hasActor	None	This property links an action to its actor(s).	Action	Actor	None
hasAdministrativeAssistant	hasEmployee	This property links an organization to one of its administrative assistants. An administrative assistant is defined as an individual who provides various kinds of administrative support to people and groups in organizations.	Organization	Person	isAdministrativeAssistantOf
hasAnalyst	hasTechnicalEmployee	This links an organization to an analyst. An analyst is defined as an employee who analyzes or is skilled in analysis.	Organization	Person	isAnalystOf
hasAsset	None	This property links an asset or actor to an asset it owns.	None	Asset	isAssetOf
hasBeneficiaryOrganization	hasEventRelation	This links an event to the beneficiary organization for the event. The beneficiary organization is defined as the organization that the insider intended to provide some benefit to through their malicious actions. The beneficiary organization may or may not have been knowingly involved in the incident.	Event	Organization	isBeneficiaryOrganizationOf
hasBoyfriend	hasFriendRelation	This relates a person to a male friend with whom that person has a romantic relationship.	Person	Person	None
hasBrother	hasFamilyRelation	This relates a male to other sons and daughters of his parents.	Person	Person	None
hasChiefExecutiveOfficer	hasUppermanagementEmployee	This links an organization to a chief executive officer. A chief executive officer is defined as a top executive in an organization.	Organization	Person	isChiefExecutiveOfficerOf
hasChiefFinancialOfficer	hasUppermanagementEmployee	This links an organization to a chief financial officer. A chief financial officer is a top executive who manages the finances of an organization.	Organization	Person	isChiefFinancialOfficerOf
hasChiefTechnicalOfficer	hasUppermanagementEmployee	This links an organization to a chief technical officer. A chief technical officer is defined as a top executive who runs the technology groups within an organization.	Organization	Person	isChiefTechnicalOfficerOf
hasColleague	hasWorkRelation	This property links colleagues. A colleague is defined as a fellow worker or member of a staff, department, profession, etc.	Person	Person	None

Name	Parent Property	Definition	Domain	Range	Inverse
hasCompetitor	hasWorkRelation	This property links competing individuals or organizations.	Actor	Actor	None
hasConspirator	hasEventRelation	This relation defines a conspirator for the insider during the event. A conspirator is a person who is involved in a secret plan to do something harmful or illegal.	Person	Person	isConspiratorOf
hasConsultant	hasWorkRelation	This property links a consultant to a customer. A consultant is defined as an individual who works independently to assist and advise client organizations with various organizational functions and responsibilities on a fee-for-service basis.	Actor	Actor	None
hasContractor	hasExternalEmployee	This links an organization to a contractor. A contractor is defined as a person or company that undertakes a contract to provide materials or labor to perform a service or do a job.	Organization	Person	isContractorOf
hasCustomer	hasWorkRelation	This property links an actor to a customer. A customer is defined as a person or organization that buys goods or services from a business or an organization.	Actor	Actor	None
hasCustomerServiceRepresentative	hasRepresentative	This links an organization to a customer service representative. A customer service representative is defined as an individual who interacts with customers to provide information in response to inquiries about products and services and handles and resolves complaints.	Organization	Person	isCustomerServiceRepresentativeOf
hasDestination	hasLocation	See parent definition.	DigitalAction	Asset	None
hasEducationEmployee	hasEmployee	This property links an organization to an employee in the education system.	Organization	Person	isEducationEmployeeOf
hasEmployee	hasWorkRelation	This property links an organization to one of its employees. An employee is defined as a person working for another person or an organization for pay.	Organization	Person	isEmployeeOf
hasEventRelation	hasRelation	This property describes an actor's role in the insider event. This role can be in relation to the event itself or to another actor in the event.	None	Actor	None

Name	Parent Property	Definition	Domain	Range	Inverse
hasExternalEmployee	hasEmployee	None.	Organization	Person	isExternalEmployeeOf
hasFamilyRelation	hasRelation	This is a connection between two people associated with familial ties.	Person	Person	None
hasFather	hasFamilyRelation	This relates a male to his child or children.	None	None	None
hasFriend	hasFriendRelation	See parent definition.	Person	Person	None
hasFriendRelation	hasRelation	This defines friends of the insider who may have knowingly or unknowingly been involved in the event.	Person	Person	None
hasGirlfriend	hasFriendRelation	This relates a person to a female friend with whom that person has a romantic relationship.	Person	Person	None
hasHusband	hasSpouse	This relates a married man to his spouse.	Person	Person	None
hasInformation	None	This property links an asset to the information it contains.	Asset	Information	None
hasInstrument	None	This links an action to an asset used in the action. This fits into an action as follows: "An actor performs an action on an object with an instrument."	Action	Asset	None
hasLocation	None	This property defines the logical or physical locations of information, an asset, or an actor.	None	Asset	None
hasManager	hasEmployee	This links an organization to a manager. A manager is defined as an employee who manages a group within an organization.	Organization	Person	isManagerOf
hasMother	hasFamilyRelation	This relates a female to her child or children.	None	None	None
hasNetworkAdministrator	hasTechnicalEmployee	This links an organization to a network administrator. A network administrator is defined as an employee who is responsible for upkeep, configuration, and reliable operation of a network.	Organization	Person	isNetworkAdministratorOf
hasObject	None	This links an action to the object that was acted upon. This fits into an action as follows: "An actor performs an action on an object with an instrument."	Action	Asset	None
hasOfficeManager	hasManager	This links an organization to an office manager. An office manager is defined as an employee that runs day-to-day operations within an office.	Organization	Person	isOfficeManagerOf

Name	Parent Property	Definition	Domain	Range	Inverse
hasPoliceOfficer	hasSecurityEmployee	This links an organization to a police officer. A police officer is defined as a person who maintains order and protects life and property by enforcing local, tribal, State, or Federal laws and ordinances. The police officer performs a combination of the following duties: patrol a specific area; direct traffic; issue traffic summonses; investigate accidents; apprehend and arrest suspects, or serve legal processes of courts.	Organization	Person	isPoliceOfficerOf
hasProfessor	hasEducationEmployee	This is a teacher of the highest academic rank in a college or university.	Organization	Person	isProfessorOf
hasReferenceTo	None	This property links a piece of information to a thing that it describes, is about, references, or makes mention of.	Information	None	None
hasRelation	None	This property describes how an actor or event is related to another actor.	None	Actor	isRelationOf
hasRepresentative	hasEmployee	This links an organization to a representative. A representative is defined as an employee who is chosen or appointed to act or speak for another or others, in particular.	Organization	Person	isRepresentativeOf
hasResearcher	hasEmployee	This links an organization to a researcher. A researcher is defined as an employee who investigates new areas of study and applications of technology.	Organization	Person	isResearcherOf
hasRetailer	hasSalesEmployee	This links an organization to a retailer. A retailer is defined as a seller of goods or commodities in small quantities directly to consumers.	Organization	Person	isRetailerOf
hasSalesEmployee	hasEmployee	This links an organization to an employee that sells a product or service provided by the organization.	Organization	Person	isSalesEmployeeOf
hasSecurityEmployee	hasEmployee	This links an organization to a person who ensures the physical safety of people or assets.	Organization	Person	isSecurityEmployeeOf
hasSecurityGuard	hasSecurityEmployee	This links an organization to a security guard. A security guard is defined as an employee who guards, patrols, or monitors a premises to prevent theft, violence, or infractions of rules.	Organization	Person	isSecurityGuardOf

Name	Parent Property	Definition	Domain	Range	Inverse
hasSister	hasFamilyRelation	This relates a female to other sons and daughters of her parents.	Person	Person	None
hasSoftwareDeveloper	hasTechnicalEmployee	This links an organization to an individual who creates software.	Organization	Person	isSoftwareDeveloperOf
hasSource	hasLocation	See parent definition.	DigitalAction	Asset	None
hasSpouse	hasFamilyRelation	This relates a husband or wife to his or her partner.	Person	Person	None
hasSubcontractor	hasExternalEmployee	This links an organization to a subcontractor. A subcontractor is defined as a person or business that contracts to provide some service or material necessary for the performance of another's contract.	Organization	Person	isSubcontractorOf
hasSystemAdministrator	hasTechnicalEmployee	This links an organization to a system administrator. A system administrator is defined as an employee who is responsible for the upkeep, configuration, and reliable operation of computer systems.	Organization	Person	isSystemAdministratorOf
hasTechnicalEmployee	hasEmployee	This links an organization to an employee whose duties typically involve computers or computer networks.	Organization	Person	isTechnicalEmployeeOf
hasTechnicalManager	hasManager	This links an organization to a technical manager. A technical manager is defined as an employee that provides technical direction and leadership for the development of products and projects.	Organization	Person	isTechnicalManagerOf
hasTechnician	hasTechnicalEmployee	This links an organization to a technician. A technician is defined as a person who is trained or skilled in the technicalities of a subject.	Organization	Person	isTechnicianOf
hasTrustedBusinessPartner	hasWorkRelation	This property defines a collaborative professional relationship between organizations or people involving some level of mutual trust.	Organization	Actor	isTrustedBusinessPartnerOf
hasUppermanagementEmployee	hasEmployee	This links an organization to an employee who holds an upper management position within the organization.	Organization	Person	isUppermanagementEmployeeOf
hasVendor	hasSalesEmployee	This links an organization to a vendor. A vendor is defined as a person that sells something.	Organization	Person	isVendorOf

Name	Parent Property	Definition	Domain	Range	Inverse
hasVictimOrganization	hasEventRelation	This links an event to the victim organization for the event. A victim organization is an organization that suffers from the malicious actions of an insider.	Event	Organization	isVictimOrganizationOf
hasWife	hasSpouse	This relates a married woman to her spouse.	Person	Person	None
hasWorkRelation	hasRelation	This describes a professional relationship.	Actor	Actor	None
isAccompliceOf	isEventRelationOf	See inverse.	Person	Person	None
isActorOf	None	See inverse.	Actor	Action	hasActor
isAdministrativeAssistantOf	isEmployeeOf	See inverse.	Person	Organization	None
isAnalystOf	isTechnicalEmployeeOf	See inverse.	Person	Organization	None
isAssetOf	None	See inverse.	Asset	None	None
isBeneficiaryOrganizationOf	isEventRelationOf	See inverse.	Organization	Event	None
isBoyfriendOf	isFriendRelationOf	See inverse.	Person	Person	hasBoyfriend
isBrotherOf	isFamilyRelationOf	See inverse.	Person	Person	hasBrother
isChiefExecutiveOfficerOf	isUppermanagementEmployeeOf	See inverse.	Person	Organization	None
isChiefFinancialOfficerOf	isUppermanagementEmployeeOf	See inverse.	Person	Organization	None
isChiefTechnicalOfficerOf	isUppermanagementEmployeeOf	See inverse.	Person	Organization	None
isColleagueOf	isWorkRelationOf	See inverse.	Person	Actor	hasColleague
isConspiratorOf	isEventRelationOf	See inverse.	Person	Person	None
isConsultantOf	isWorkRelationOf	See inverse.	Actor	Actor	hasConsultant
isContractorOf	isExternalEmployeeOf	See inverse.	Person	Organization	None
isCustomerOf	isWorkRelationOf	See inverse.	Actor	Actor	hasCustomer
isCustomerServiceRepresentativeOf	isRepresentativeOf	See inverse.	Person	Organization	None

Name	Parent Property	Definition	Domain	Range	Inverse
isEducationEmployeeOf	isEmployeeOf	See inverse.	Person	Organization	None
isEmployeeOf	isWorkRelationOf	See inverse.	Person	Organization	None
isEventRelationOf	isRelationOf	See inverse.	Actor	None	hasEventRelation
isExternalEmployeeOf	isEmployeeOf	See inverse.	Person	Organization	None
isFamilyRelationOf	isRelationOf	See inverse.	Person	Person	hasFamilyRelation
isFatherOf	isFamilyRelationOf	See inverse.	Person	Person	hasFather
isFriendOf	isFriendRelationOf	See inverse.	Person	Person	hasFriend
isFriendRelationOf	isRelationOf	See inverse.	Person	Person	hasFriendRelation
isGirlfriendOf	isFriendRelationOf	See inverse.	Person	Person	hasGirlfriend
isHusbandOf	isSpouseOf	See inverse.	Person	Person	hasHusband
isManagerOf	isEmployeeOf	See inverse.	Person	Organization	None
isMotherOf	isFamilyRelationOf	See inverse.	Person	Person	hasMother
isNetworkAdministratorOf	isTechnicalEmployeeOf	See inverse.	Person	Organization	None
isOfficeManagerOf	isManagerOf	See inverse.	Person	Organization	None
isPartnerOf	isFamilyRelationOf	See inverse.	Person	Person	None
isPoliceOfficerOf	isSecurityEmployeeOf	See inverse.	Person	Organization	None
isProfessorOf	isEducationEmployeeOf	See inverse.	Person	Organization	None
isReferencedBy	http://www.w3.org/2002/07/owl#topObjectProperty	See inverse.	None	None	hasReferenceTo
isRelationOf	None	See inverse.	None	None	None
isRepresentativeOf	isEmployeeOf	See inverse.	Person	Organization	None
isResearcherOf	isEmployeeOf	See inverse.	Person	Organization	None
isRetailerOf	isSalesEmployeeOf	See inverse.	Person	Organization	None

Name	Parent Property	Definition	Domain	Range	Inverse
isSalesEmployeeOf	isEmployeeOf	See inverse.	Person	Organization	None
isSecurityEmployeeOf	isEmployeeOf	See inverse.	Person	Organization	None
isSecurityGuardOf	isSecurityEmployeeOf	See inverse.	Person	Organization	None
isSisterOf	isFamilyRelationOf	See inverse.	Person	Person	hasSister
isSoftwareDeveloperOf	isTechnicalEmployeeOf	See inverse.	Person	Organization	None
isSpouseOf	isFamilyRelationOf	See inverse.	Person	Person	hasSpouse
isSubcontractorOf	isExternalEmployeeOf	See inverse.	Person	Organization	None
isSubjectOf	None	None.	None	Information	None
isSystemAdministratorOf	isTechnicalEmployeeOf	See inverse.	Person	Organization	None
isTechnicalEmployeeOf	isEmployeeOf	See inverse.	Person	Organization	None
isTechnicalManagerOf	isManagerOf	See inverse.	Person	Organization	None
isTechnicianOf	isTechnicalEmployeeOf	See inverse.	Person	Organization	None
isTrustedBusinessPartnerOf	isWorkRelationOf	See inverse.	Actor	Organization	None
isUppermanagementEmployeeOf	isEmployeeOf	See inverse.	Person	Organization	None
isVendorOf	isSalesEmployeeOf	See inverse.	Person	Organization	None
isVictimOrganizationOf	isEventRelationOf	See inverse.	Organization	Event	None
isWifeOf	isSpouseOf	See inverse.	Person	Person	hasWife
isWorkRelationOf	isRelationOf	See inverse.	Actor	Actor	hasWorkRelation
istemporallySubsumedBy	temporallyRelatedTo	See inverse. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	temporallySubsumes

Name	Parent Property	Definition	Domain	Range	Inverse
overlapsEnd	temporallyRelatedTo	This property defines a temporal thing that starts after, partially occurs during, and ends after another temporal thing. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	overlapsStart
overlapsStart	temporallyRelatedTo	This property defines a temporal thing that starts before, partially occurs during, and ends before another temporal thing. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	None
startedBy	temporallyRelatedTo	This predicate means that subject and object start to occur or exist at the same time point (see startingPoint) and that subject ends or ceases to exist (see endingPoint) after the object ends or ceases to exist. For example, subject might be a WeddingCeremony and object might be the bride's walk down the aisle. Derived from OpenCyc 1.0 (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	None
starts	temporallyRelatedTo	See inverse. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	startedBy
takesPlaceAfter	temporallyRelatedTo	See inverse. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	takesPlaceBefore
takesPlaceBefore	temporallyRelatedTo	This property links a subject to an object such that the subject starts and ends before the object starts. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	None
temporally RelatedTo	None	This property links two entities so as to characterize their overlap in time.	TemporalThing	TemporalThing	temporallyRelatedTo
temporally Subsumes	temporallyRelatedTo	This property defines a relationship between subject and object where the subject starts after and ends before the subject. (Adapted from Eric Peterson's SpaceTime ontology: http://semanic.org/OntDef/Cur/SpaceTime.owl)	TemporalThing	None	None

Appendix C: Definitions of Top 10 Observation Groupings

Verification of Modification of Critical Data: A failure of the organization to implement controls that prevent unauthorized modification of critical data (e.g., the insider was able to remotely access the victim organization's systems, delete files, modify employee information, and change passwords).

Disgruntled Employee: An insider who is upset with the organization and desires to get back at it (e.g., the victim organization rejected a contract for the insider's own firm, and the insider plots to make the new systems administrator look bad).

Used Excessive Access Privilege—General: The insider having greater access to the organization's IT systems than is necessary for the insider's work.

Unauthorized Data Exports—Unknown: The insider removing organizational data from the organization through unknown means (e.g., the insider stole source code while working as a consultant and before announcing his/her resignation).

Compromised Passwords: The insider being able to access the organizational system due to the compromise of another employee's password (e.g., the insider copied another employee's account and password prior to being terminated).

Email/Chat with External Competitors/Conspirators: The insider communicating, through an IT system, with others related to the attack (e.g., the insider emailed source code to a personal account, then to the conspirators).

Failure to Protect Critical Files: An organizational failure to put into place sufficient protections to guard files critical to the organization (e.g., the insider had the ability to potentially wipe out all backup files with a logic bomb in a trusted script).

Violation of Need-to-Know Policy: An insider accessing organizational information that is not needed for his or her work, as defined by organizational policy (e.g., the insider downloaded personal DMV records that were not part of his or her need to know).

Unauthorized Data Download to/from Home: An unauthorized download of organizational data to or from the insider's home (e.g., at an unknown time, a former employee exceeded authorized access and obtained employee PII).

Ability of Users with System Administrator Privileges to Sabotage Systems or Data: An insider with system administrator privileges sabotaging the organization's system (e.g., insiders were able to manipulate data on the system after installing key logging software to obtain username/password).

References

URLs are valid as of the publication date of this document.

[1] G. J. Silowash, D. M. Cappelli, A. P. Moore, R. F. Trzeciak, T. Shimeall, and L. Flynn, "Common sense guide to mitigating insider threats," 2012.

[2] F. L. Greitzer and D. A. Frincke, "Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation," in *Insider Threats in Cyber Security*, ed: Springer, 2010, pp. 85-113.

[3] A. Memory, H. G. Goldberg, and T. E. Senator, "Context-Aware Insider Threat Detection," in *Proceedings of the Workshop on Activity Context System Architectures*, 2013.

[4] D. Caputo, M. Maloof, and G. Stephens, "Detecting insider theft of trade secrets," *Security & Privacy, IEEE,* vol. 7, pp. 14-21, 2009.

[5] Y. Chen and B. Malin, "Detection of anomalous insiders in collaborative environments via relational analysis of access logs," in *Proceedings of the first ACM conference on Data and application security and privacy*, 2011, pp. 63-74.

[6] M. A. Maloof and G. D. Stephens, "elicit: A system for detecting insiders who violate need-to-know," in *Recent Advances in Intrusion Detection*, 2007, pp. 146-166.

[7] D. Ha, S. Upadhyaya, H. Ngo, S. Pramanik, R. Chinchani, and S. Mathew, "Insider threat analysis using information-centric modeling," in *Advances in Digital Forensics III*, ed: Springer, 2007, pp. 55-73.

[8] W. Eberle, J. Graves, and L. Holder, "Insider threat detection using a graph-based approach," *Journal of Applied Security Research,* vol. 6, pp. 32-81, 2010.

[9] J. F. Buford, L. Lewis, and G. Jakobson, "Insider threat detection using situation-aware MAS," in *Information Fusion, 2008 11th International Conference on*, 2008, pp. 1-8.

[10] E. Santos, H. Nguyen, F. Yu, K. Kim, D. Li, J. T. Wilkinson*, et al.*, "Intent-driven insider threat detection in intelligence analyses," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, 2008, pp. 345-349.

[11] M. Kandias, V. Stavrou, N. Bozovic, and D. Gritzalis, "Proactive insider threat detection through social media: The YouTube case," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, 2013, pp. 261-266.

[12] J. S. Park and J. Giordano, "Role-based profile analysis for scalable and accurate insider-anomaly detection," in *Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International*, 2006, pp. 7 pp.-470.

[13] A. Natarajan and L. Hossain, "Towards a social network approach for monitoring insider threats to information security," in *Intelligence and Security Informatics*, ed: Springer, 2004, pp. 501-507.

[14] T. R. Gruber, "The role of common ontology in achieving sharable, reusable knowledge bases," *KR,* vol. 91, pp. 601-602, 1991.

[15] M. West, *Developing high quality data models*: Elsevier, 2011.

[16] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry*, et al.*, "Gene Ontology: tool for the unification of biology," *Nature genetics,* vol. 25, pp. 25-29, 2000.

[17] S. Schulze-Kremer, "Adding semantics to genome databases: towards an ontology for molecular biology," in *Ismb*, 1997, p. 5.

[18] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "OWL 2 web ontology language primer," *W3C recommendation,* vol. 27, p. 123, 2009.

[19] F. Baader, *The description logic handbook: theory, implementation, and applications*: Cambridge university press, 2003.

[20] P. F. Patel-Schneider, "Tutorial on the W3C OWL Web Ontology Language ENC 2004," Bell Labs Research2004.

[21] F. Baader, I. Horrocks, and U. Sattler, "Description logics," in *Handbook on ontologies*, ed: Springer, 2004, pp. 3-28.

[22] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition,* vol. 5, pp. 199-220, 1993.

[23] P. W. Hayes, Chris. (2006). *Defining N-ary Relations on the Semantic Web*. Available: http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/

[24] A. Akbik and A. Löser, "Kraken: N-ary facts in open information extraction," in *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, 2012, pp. 52-56.

[25] M. Grüninger and M. S. Fox, "The role of competency questions in enterprise engineering," in *Benchmarking—Theory and Practice*, ed: Springer, 1995, pp. 22-31.

[26] A. Gangemi, "Ontology design patterns for semantic web content," in *The Semantic Web–ISWC 2005*, ed: Springer, 2005, pp. 262-276.

[27] N. Drummond, A. L. Rector, R. Stevens, G. Moulton, M. Horridge, H. Wang*, et al.*, "Putting OWL in Order: Patterns for Sequences in OWL," in *OWLED*, 2006.

[28] U. GOVRNMENT, "Executive Order 13587-Structural Reforms To Improve the Security of Classified Networks and the Responsible Sharing and Safeguarding of Classified Information," 2011.

[29] B. Obama, "National Insider Threat Policy and Minimum Standards for Executive Branch Insider Threat Programs," T. W. House, Ed., ed: Office of the Press Secretary, 2012, p. 1.

[30] F. o. A. Scientists, "National Insider Threat Policy and Minimum Standards for Executive Branch Insider Threat Programs (Minimum Standards)," T. W. Hourse, Ed., ed. www.fas.org: Federation of American Scientists, 2012.

[31] A. J. Cañas and J. D. Novak, "What is a concept map," *Institute for human and machine cognition,* 2009.

[32] R. R. Starr and J. M. P. de Oliveira, "Conceptual maps as the first step in an ontology construction method," in *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International*, 2010, pp. 199-206.

[33] J. D. Novak and A. J. Cañas, "The theory underlying concept maps and how to construct them," *Florida Institute for Human and Machine Cognition,* vol. 1, 2006.

[34] J. J. Villalon and R. A. Calvo, "Concept Map Mining: A definition and a framework for its evaluation," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, 2008, pp. 357-360.

[35] K. Žubrinic, "Automatic creation of a concept map."

[36] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, 2002, pp. 63-70.

[37] A. J. Cañas, G. Hill, R. Carff, N. Suri, J. Lott, M. Arroyo*, et al.*, "CmapTools: A knowledge modeling and sharing environment."

[38] J.-b. Gao, B.-w. Zhang, X.-h. Chen, and Z. Luo, "Ontology-based model of network and computer attacks for security assessment," *Journal of Shanghai Jiaotong University (Science),* vol. 18, pp. 554-562, 2013.

[39] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in *Proceedings of the 4th international Symposium on information, Computer, and Communications Security*, 2009, pp. 183-194.

[40] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain," *Proceedings of Semantic Technologies for Intelligence, Defense, and Security (STIDS),* pp. 49-56, 2012.

[41] S. E. Parkin, A. van Moorsel, and R. Coles, "An information security ontology incorporating human-behavioural implications," in *Proceedings of the 2nd International Conference on Security of Information and Networks*, 2009, pp. 46-55.

[42] *schema.org*. Available: schema.org

[43] R. Tate, *The 20% Doctrine*: HarperCollins, 2012.

[44] Y. Shahar, "A framework for knowledge-based temporal abstraction," *Artificial intelligence,* vol. 90, pp. 79-133, 1997.

[45] S. K. Cha, T. Avgerinos, A. Rebert, and D. Brumley, "Unleashing mayhem on binary code," in *Security and Privacy (SP), 2012 IEEE Symposium on*, 2012, pp. 380-394.

[46] M. Bishop and C. Gates, "Defining the insider threat," in *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, 2008, p. 15.

[47] C. Huth, D. Mundie, and S. Perl, "Toward an Ontology for Insider Threat Research: Varieties of Insider Threat Definition," in *Proc. of the 3rd Workshop on Socio-Technical Aspects in Security and Trust (STAST'13), Tulane University, New Orleans, LA, USA. IEEE*, 2013.

[48] (2014). *Thing > Person*. Available: http://schema.org/Person

[49] K. Setiya, "Intention," in *The Stanford Enclopedia of Philosophy*, E. N. Zalta, Ed., Spring 2014 ed, 2014.

[50] D. Bulling, M. Scalora, R. Borum, J. Panuzio, and A. Donica, "Behavioral science guidelines for assessing insider threats," 2008.

[51] I. J. Martinez-Moyano, E. Rich, S. Conrad, D. F. Andersen, and T. R. Stewart, "A behavioral theory of insider-threat risks: A system dynamics approach," *ACM Transactions on Modeling and Computer Simulation (TOMACS),* vol. 18, p. 7, 2008.

[52] D. Brickley. (2014). *foaf project*. Available: http://www.foaf-project.org/

[53] H. Park, S. Cho, and H.-C. Kwon, "Cyber forensics ontology for cyber criminal investigation," in *Forensics in Telecommunications, Information and Multimedia*, ed: Springer, 2009, pp. 160-165.

[54] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*: Pearson Education, 2012.

[55] G. Antoniou and F. Van Harmelen, "Web ontology language: Owl," in *Handbook on ontologies*, ed: Springer, 2004, pp. 67-92.

[56] R. Trzeciak, "The CERT Insider Threat Database," in *Insider Threat Blog*, ed. cert.org, 2011.

[57] B. Martini and K.-K. R. Choo, "An integrated conceptual digital forensic framework for cloud computing," *Digital Investigation,* vol. 9, pp. 71-80, 2012.

[58] R. Lee, "SANS Digital Forensics and Incident Response Poster Released," in *Blog: SANS Digital Forensics and Incident Response Blog* vol. 2014, S. D. Faculty, Ed., ed. SANS: SANS, 2012.

[59] *CybOX - The MITRE Corporation's Structured Language for Cyber Observables*. Available: http://cybox.mitre.org/

[60] *STIX - the MITRE Corporation's Structured Threat Information Expression Standard*. Available: https://stix.mitre.org/

[61] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*: O'Reilly Media, Inc., 2009.

[62] W. N. Francis and H. Kucera, "Brown corpus manual," *Brown University Department of Linguistics,* 1979.

[63] C. Janssen, "Data Exfiltration," in *techopedia*, ed. techopedia.com, 2014.

[64] E. Peterson. *SpaceTime Ontology*. Available: http://semanic.org/Ontology/OntDef/Cur/SpaceTime.owl

[65] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM,* vol. 26, pp. 832-843, 1983.

[66] M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens, and C. Wroe, "A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition1. 2," *The University of Manchester,* 2009.

[67] W. C. O. W. Group, "{OWL} 2 Web Ontology Language Document Overview," 2009.

[68] *Office of the Director of National Intelligence: Metaphor Program*. Available: http://www.iarpa.gov/index.php/research-programs/metaphor

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. **AGENCY USE ONLY**
(Leave Blank)

2. **REPORT DATE**
May 2016

3. **REPORT TYPE AND DATES COVERED**
Final

4. **TITLE AND SUBTITLE**
An Insider Threat Indicator Ontology

5. **FUNDING NUMBERS**
FA8721-05-C-0003

6. **AUTHOR(S)**
Daniel L. Costa, Michael J. Albrethsen, Matthew L. Collins, Samuel J. Perl, George J. Silowash, & Derrick L. Spooner

7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. **PERFORMING ORGANIZATION REPORT NUMBER**
CMU/SEI-2016-TR-007

9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
AFLCMC/PZE/Hanscom
Enterprise Acquisition Division
20 Schilling Circle
Building 1305
Hanscom AFB, MA 01731-2116

10. **SPONSORING/MONITORING AGENCY REPORT NUMBER**
n/a

11. **SUPPLEMENTARY NOTES**

12A **DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified/Unlimited, DTIC, NTIS

12B **DISTRIBUTION CODE**

13. **ABSTRACT (MAXIMUM 200 WORDS)**

The insider threat community currently lacks a standardized method of expression for indicators of potential malicious insider activity. We believe that communicating potential indicators of malicious insider activity in a consistent and commonly accepted language will allow insider threat programs to implement more effective controls through an increase in collaboration and information sharing with other insider threat teams. In this report, we present an ontology for insider threat indicators. We make the case for using an ontology to fill the stated gap in the insider threat community. We also describe the semi-automated, data-driven development of the ontology, as well as the process by which the ontology was validated. In the appendices, we provide the ontology's user's manual and technical specification.

14. **SUBJECT TERMS**
Insider Threat indicators, ontology

15. **NUMBER OF PAGES**
87

16. **PRICE CODE**

17. **SECURITY CLASSIFICATION OF REPORT**
Unclassified

18. **SECURITY CLASSIFICATION OF THIS PAGE**
Unclassified

19. **SECURITY CLASSIFICATION OF ABSTRACT**
Unclassified

20. **LIMITATION OF ABSTRACT**
UL