TU Delft Delft University of Technology

Attack Pattern Ontology – A Common Language for Attack Information Sharing Between Organisations

Yiwen Zhu

September 2015

Graduation Committee:

Chairman: Prof.dr.ir. Marijn Janssen
Section Information and Communication Technology, Faculty TPM, TUDelft

First supervisor: Dr.ir. Wolter Pieters
Section Information and Communication Technology, Faculty TPM, TUDelft

Second supervisor: Dr. Michel Oey
Section System Engineering, Faculty TPM, TUDelft

Third supervisor: Dr.ir. Dina Hadžiosmanović
Section Information and Communication Technology, Faculty TPM, TUDelft

Master thesis by Yiwen Zhu
At Delft University of Technology
MSc Program: System Engineering, Policy Analysis and Management
Faculty: Technology, Policy and Management

Executive summary

Cyber attack nowadays is increasingly being reported. Organisations must protect every potential vulnerability to secure a system; yet, to attack a system, attackers only need to find a single vulnerability. Therefore, defenders need a good understanding of attacker's perspective in order to accurately anticipate threats and effectively mitigate attacks. They can gain such understanding through sharing attack information with other organisations.

According to the current situation and trends, we propose sharing attack pattern as a means to enhance cyber security, which has 3 advantages over other ways of information sharing: attack pattern captures attacker's perspective that helps defender to accurately anticipate threats; attack pattern is generic thus sharing it decreases the possibility of disclosing vulnerabilities of single organisation; attack pattern excludes redundant details that may be inapplicable for most of the organisations.

However, 2 knowledge gaps exist on the topic of sharing attack pattern (a) there is no shared understanding about the attack pattern concept and (b) information sharing is not considered as one usage of attack pattern. In order to fill in these knowledge gaps, the main research question of this thesis is:

> How to build a uniform way to represent the entities, their properties and relations of the attack pattern system to improve interaction with the attack information sharing system?

The thesis will deliver an attack pattern ontology to answer this question. The research is conducted in three main phases: analysis, design and evaluation. In the analysis phase, we collect information about attack pattern and ontology through desk research. Five requirements on both the ontology contents and usages were generated to guide the design phase. Then, in the design phase, the elements of the attack pattern ontology were defined and described; classes, object properties, data properties and annotation properties were listed and defined. Four attack pattern instances were produced to show how the ontology works. Different to the existing researches, our attack pattern ontology emphasises the following features of attack pattern:

- Different to the normal way of pattern one attack step into one attack pattern, we pattern all the attack steps of complete attack incidents into one attack pattern.
- Information producer remains anonymous that the ontology does not provide any content relate to the information source in any form.

Different to the practice of the attack pattern concept in CAPEC, our attack pattern ontology has the following advantages:

- The ontology supports information consuming with a single clear structure and defined relations.
- Attack pattern contents are reusable that two different attack patterns can share one attack method, vulnerability, consequence, etc.
- Data reserved based on the ontology is 'smart' that auto-classification and auto consistency checking are possible.

At last in the evaluation phase, the ontology is assessed both qualitatively and quantitatively; five master students were invited to evaluate six criteria of the ontology. From their feedbacks, we concluded the following two ways of helping users to easier understand and use the ontology:

- Adding annotations to define the classes and explain their usages;
- Adding more subclasses, i.e. increase the depth of the hierarchy, to provide sense of what kind of events or objects belongs to one class

During the process of finishing this project, we found that generating requirements for ontology may be effective and helpful in framing the proposed ontology design, especially when the topic is complex and unfamiliar to the designer. In addition to this practical lesson learned, readers may also utilize the information from this thesis on 3 aspects (a) What is attack pattern? (b) What are the considerations on the contents of shareable attack information? (c) What are the potential usages of an ontology in the domain of cyber security?

This research has limitations on both the methodology of developing ontology and the delivered product. Firstly, the scope was limited to between organisations, which means information is shared between different organisations in the same country. Other levels of information sharing can also be relevant: within an organization, between legal bodies. Next, the lower level classes were developed based on existing attack pattern instances. But existing attack pattern concept present something different to what this research intended to capture: one step of one type of attack vs. all steps of one type of attack. Third, the questionnaire was only answered by students but not professionals. The ontology is limited to well-educated user group and users must make agreements and rules of how to use the ontology beforehand, which means the ontology will be used differently among different information sharing communities.

This thesis can be further developed in several directions: (a) the sharing level can be extended to inter-department and international; (b) the characteristics of each industry can be integrated into the design of attack pattern ontology, i.e. develop different ontologies for different information sharing communities; (c) the attack information can be combined with attacker information, which will lead to a more powerful attack model.

***Key words**: Information sharing; Cyber security; Cyber attack; Attack pattern; Ontology*

Acknowledgements

This document represents the final report of the Master Thesis Project that concludes my MSc. Education in the Master program SEPAM (System Engineering, Policy Analysis and Management) at Delft University of Technology. The project has been conducted for the ICT (Information and Communication Technology) department in the faculty of TPM (Technology Policy and Management).

This thesis would not have been possible without the help and support of a group of people. This is the place where I express my gratefulness to everyone that has made this thesis research possible.

I would like to thank my graduation committee members - Marijn Janssen, Wolter Pieters, Michel Oey and Dina Hadžiosmanović. Thank you for guiding and helping me through the whole process of finishing this thesis project. Marijn provided many keen suggestions from an overall view. Wolter not just provided academic support but also additional learning tunnels. We have a master thesis group of five students where we can learn from each other. I am also glad to have participated in the annual workshop on the economics of information security, which is held in our campus this year. Michel gave support from different perspective that he can always point out the weak point in the thesis. A special thank to Dina, who has taken so much time to discuss the thesis with me and help improving my report.

I would also like to express my gratitude to my classmates who participated in the questionnaire. Thank you for your time and suggestions. Your opinions formed the foundation for the evaluation stage of my research. Moreover, many thanks to my thesis group members. Thank you for all your help in the regular meetings and your comments in peer reviews.

Last but not least, I am grateful for all the support I received from my family and friends. Thank you for your encouragement and faith in me.

Table of contents

Table of figures

Table of tables

Glossary of terms

A list of terms related to the attack pattern topic is given below. Meaning and explanation of the terminologies are given in the second column.

Term	Meaning
(Cyber) Attack	A malicious attempt to gain unauthorized access to system services, resources, or information for the purpose of using, altering, exposing, stealing, disabling, or destroying an asset
Pattern	Something that happens in a regular and repeated way
Attack pattern	A blueprint or generic representation describes how to perform and execute attack from the point of view of an attacker. It presents the critical features of the exploited vulnerability, the knowledge required for an attacker to perform the attack, the steps to perform the attack and the ways to counteract the development of the attack.
Attacker's point of view	Opposite to protecting and defending, from the point of view of attackers, people try to solve the problems they may face when attacking a target
Ontology	A semantic web model to provide a common language of a domain of knowledge that is exchanged and shared; it gives a description of entities and their properties, relationships, constraints.

Source: adapted from Uschold & Gruninger, 1996; Noy & McGuinness, 2001; Kuhn, 2001; Moore *et al.,* 2001; Hoglund & McGraw, 2004; Barnum & Sethi, 2007; Fernandez *et al.,* 2007; Gegick & Williams, 2007; Pauli & Engebretson, 2008; Uzunov & Fernandez, 2014; Bayley, 2014.

1. Introduction

1.1 Emerging cyber security problems

Cyber security is the act of protecting information and communication technology (ICT) systems and their contents (Fischer, 2014). It is not only a technical issue but also a societal one. Individuals, organisations and nations are given incredible power from the constantly developing Internet and networking technology. As a consequence, all political and military conflicts now have a cyber dimension; battles taking place in cyberspace could be more important than the ones on the ground (Geers, 2011). Cyber attack is not an end in itself, but a powerful means that can increase the speed, scale and impact of an attack to a wide variety of ends (Geers, 2011). These attacks are the price of the convenience brought by the Internet.

Cyber attack nowadays is increasingly being reported describing security breaches in both governments and large corporations. Attack is not growing only in frequency but also in scale scope and complexity (Johnson *et al.,* 2014). The complexity and size of system increase while the number and the skill level of attackers continues to grow (Barnum & Sethi, 2007). As a result, securing cyberspace has become more challenging; if cyber attacks were just simple bicycles 25 years ago and cars 10-15 years ago, they are space shuttles now (Miller, 2015). In its Norton report, Semantec (2013) reported that cybercrime cause victims worldwide lose around 290 billion euros every year. Furthermore, it is likely that many incidents are not available to public or even remain undetected, like cyber espionage (Johnson, Badger, & Waltermire, 2014). Therefore the number of attacks that actually took place is likely even higher than reported; the security problem could be more severe than people's cognition.

Cyber defence is suffering from the fact of technical expertise shortage, little moral inhibition to attack, and the traditional security skills are of little help (Geers, 2011). Furthermore, economic features are often more focused than security in ICT design; Cyber security can be expansive with unsure economic returns on investments but on the contrary, cyber attack is cheap and profitable (Fischer, 2014). Therefore, the current cyber security environment is in favour of the attacker (Geers, 2011).

1.2 Arising concerns of stakeholders

Cyber security is not only the responsibility of organisations or government; it is intertwined with everyone's daily life. Citizens, businesses and government bodies are using the Internet for interactions, collaboration and communication (The Minister of Security and Justice, 2013). Many new applications are emerging including big data, cloud computing and the Internet of things, which complicate the threat environment (Fischer, 2014).

Some concerns about cyber security has emerged, which refers to different things for different stakeholders (Fischer, 2014). These concerns could be conflicting as a result of different interests of stakeholders. For example, Internet service providers want to reduce the cost of security measures and loss of reputation (Bauer & van Eeten, 2009). However, customers want to increase the cybercrime exposure and be aware of security risks (Bauer & van Eeten, 2009). Obviously, reducing reputation loss and increasing awareness of security risks are conflicting.

One of the major concerns shared by all stakeholders is insufficient expertise. The number of personnel with expertise is limited and far from enough for the overall needs (The Minister of Security and Justice, 2013). Because of expertise shortage, governments cannot keep pace with threats and attacks; talented computer scientists often prefer better income positions (Geers, 2011). Many small and medium enterprises (SMEs) are not able to employ security specialists, thus they do not have sufficient resources to defend against sophisticated cyber attacks (Bauer & van Eeten, 2009). It is also common for individual businesses and citizens that they underestimated their risk exposure (Bauer & van Eeten, 2009).

Governance is another challenge of government. The cyber security domain is complex, governance cannot be done by only one stakeholder; governments must cooperate with other organisations to solve problems such as security standards (The Minister of Security and Justice, 2013).

For businesses, another main concern comes from the conflicts between profits and brand reputation. These organisations need to maintain their public reputation as well as economic profit (Bauer & van Eeten, 2009). So even if they can correct the errors that caused attacks and are capable to block attacks, organisations still resist revealing the attack information for fear of losing public reputation (Moore *et al.*, 2001). Furthermore, they need to balance the cyber security investment with the potential damage caused by insufficient investment. However, they are not incentivised to invest in cyber security because of the unsure or low economic returns (Fischer, 2014).

Individuals and citizens often have wrong perception of low risk exposure, which caused them suffer from cyber attacks (Bauer & van Eeten, 2009). Although a growing number of individuals have realised the potential risks of cyber security breaches, most individuals have not yet and do not invest enough in security; they do not purchase security services or use it even offered free, and turn off their firewalls and virus scanners to achieve faster using of computers such as gaming (Bauer & van Eeten, 2009).

1.3 Research subject: sharing cyber attack information

Facing the situation of cyber security environment that favours the attacker, organisations feel the emergency of effectively enhancing cyber security. Specifically, the concern of expertise shortage urges people to share and reuse the limited expertise. Moreover, governance cannot be done without collaboration among organisations. Just because ordinary Internet users always underestimate their risk exposure, the organisations that provide products and services should work harder to decrease the risks. Therefore, organisations should work together to enhance cyber security instead of depending on sole knowledge source - themselves.

Sharing attack information is an important way to defend attacks. It is especially useful and necessary now because of the increasing number of targeted attacks (Micro, 2015). Targeted attack is much more effective and damaging than opportunistic attack that attackers tailor actions for the single target and they are willing to spend extra effort until succeed (GFI, 2009; Microsoft, 2012). Thus targeted attack is more sophisticated and hard to be defended. Through information sharing, people can obtain data about detection and patching vulnerabilities, which is hard to be done within a single organisation (Johnson *et al.*, 2014). As a consequence, the defence capability of the overall information sharing community can be increased.

Unfortunately in cyber security area, people are hesitating and irresolute about sharing previous failures. Both public and private organisations have concerns of the negative consequence of exposing their vulnerabilities; potential attackers could exploit the same or similar flaw of their systems (Moore, Ellison, & Linger, 2001); moreover, the private organisations fear of the public organisations to use these vulnerabilities against them later (Gal-Or & Ghose, 2004).

So enhancing the capability against cyber attacks is in a dilemma situation that sharing attack information is both wanted and rejected.

1.4 Research motivation: trends and constraints

Nevertheless, attack information still need to be shared. Facing the sophisticated attack techniques and easy-to-use attack tools (National Cyber Security Centre, 2013), people cannot abandon this cost-effective way of enhancing security (ENISA, 2013). European Union Agency for Network and Information Security (ENISA) is working with the member states, the European Commission and the private sectors to enhance information sharing on good security practices and lessons learnt (ENISA, 2015). Moreover, it planned 200,000 Euro budget for 2016 on facilitating voluntary information sharing techniques and establishing mutual interactions with stakeholders (ENISA, 2015). The goal is to enhance the quality of information collection, assessment and validation in the area of information sharing and threat analysis (ENISA, 2015).

Different kinds of attack information can have different effectiveness in sharing. People need sharable and reusable attack information. Attack information, i.e. the past experience on security failures, is used to mitigate and prevent future attacks. Organisations must protect every potential vulnerability to secure a system; yet, to attack a system, attackers only need to find a single vulnerability (Barnum & Sethi, 2007). Therefore, in order to accurately anticipate threats and effectively mitigate attacks, people must have a good understanding of attacker's perspective and their approaches (Barnum & Sethi, 2007; Hoglund & McGraw, 2004). Attack pattern is such a means that it captures attacker's perspective and facilitates early mitigation of potential attacks (Moore et al., 2001; Barnum & Sethi, 2007; Fernandez et al., 2007; Uzunov & Fernandez, 2014).

Therefore, attack pattern is a useful type of attack information for information sharing. Many attacks occur in similar ways in different contexts or environments, implying that the vulnerabilities or flaws are also the same or similar (Kumar & Spafford, 1994). Such repeated vulnerabilities and corresponding countermeasures can be expressed as patterns that will guide secure design and evaluation to prevent a variety of attacks (Fernandez, VanHilst, Petrie, &

Huang, 2006; Schumacher *et al.,* 2013). Attack pattern is such a way to represent commonly occurred attacks and reuse attack information (Schaeffer-Filho & Hutchison, 2014).

Attack pattern is also a sharable type of attack information that it addresses organisations' concerns on revealing their vulnerabilities. The sensitive nature of attack data obstructs organisations from sharing specific details of the incidents, which could leak the vulnerabilities of them; they want to decrease the risk of both attackers and competitors making use of such information. Attack pattern is a generic way to represent attack information thus organisations will not take the risks of disclosing weaknesses easily. Besides, some details that could be seen in single attack incidents are highly dependent on the specific condition of one organisation, which is not applicable for other organisations. Instead, these details could mislead other organisations and obstruct them from seeing the nature of the attack. Attack pattern excludes such unnecessary and redundant details, which makes the shared information more effective and straightforward.

To summarise, sharing attack information in the form of attack pattern has 3 advantages over other types of information: attack pattern captures attacker's perspective that helps defender to accurately anticipate threats; attack pattern is generic that it decrease the possibility of disclosing vulnerabilities of single organisation; attack pattern excludes redundant details that may be inapplicable for most of the organisations.

1.5 Conclusion

This chapter introduces the context of sharing cyber security information and the current situation of the cyber security domain. It points out the severe condition for defenders and the concerns of different stakeholders. Although people are hesitating and irresolute about sharing previous failures, sharing attack information is an inevitable trend for defenders. Attack pattern is introduced as an appropriate type of attack information to be shared.

2. Research description

The previous chapter has given the context of this research; in this chapter, we are going to specify how we are going to contribute to solving one problem within this context. We provide an overview of this research through the knowledge gap, research objective, deliverable, scope, contribution, question and method.

2.1 Knowledge gap

To the best of our knowledge, there is no shared understanding about the attack pattern concept yet. Although many researches employed the term 'attack pattern' in their works, they use different interpretations of this concept; some differences exist on the scope, form and content of 'attack pattern'. Many researches bounded the scope of attack pattern's application to *software attacks* (Hoglund & McGraw, 2004; Barnum & Sethi, 2007; Pauli & Engebretson, 2008) but at the same time, we also see a broader scope stated as *attacks* (Moore *et al.,* 2001; Fernandez *et al.,* 2007; Gegick & Williams, 2007; Zhu, 2011; Uzunov & Fernandez, 2014). The understanding about the nature of attack pattern is also diverse; an attack pattern can refer to a time signature of a specific attack (Thonnard & Dacier, 2008), it can also be a sequence of attacks that correlated to the security breach (Zhu, 2011). The approaches to pattern attacks show even more variety; it can be presented in a tree structure (Robiah et al., 2010), a series of events (Gegick & Williams, 2007) or a literal description (Barnum & Sethi, 2007; Fernandez, Pelaez, & Larrondo-Petrie, 2007). In the researches that adopted literal descriptions, we found various templates of presenting attack pattern.

Furthermore, existing studies did not consider the issue of information sharing outside an organisation. Neither the data source of attack pattern nor the beneficiaries of the attack pattern were stated clearly. Hence these researches assume reusing attack pattern in the same organisation, i.e. the data source of attack pattern and the beneficiary are the same (Moore et al., 2001; Barnum & Sethi, 2007; Fernandez et al., 2007). However, sharing information before, during and after an attack can alert other people of potential attacks and provide critical information to enhance each organisation's own defences. Therefore it is beneficial to involve the usage of information sharing in the concept of attack pattern. Unfortunately in existing studies, attack pattern is a means of presenting knowledge that can be later reused internally, the shareability of attack pattern or the way to enable attack pattern sharable remains unknown.

The knowledge gaps are summarised below:

- No common understanding about attack pattern
  Different studies define different scopes, purposes and contents of the term attack pattern. Furthermore, they propose various approaches to express this concept. A common understanding about the attack pattern concept would enable information sharing between organisations and facilitate effective communications.
- Lack of consideration about sharing

> Currently it is unclear how the attack pattern concept (the scope and definition, the building process and its interrelationship with other terms) can be shaped for the purpose of information sharing.

2.2 Research objective and deliverable

In order to fill in the knowledge gaps listed above, this research aims at developing a common language that uses attack pattern as the carrier of data in information sharing, rather than treat it only as an approach to record public knowledge or present particular types of attack. A common language means that there should not be differences of understanding about attack pattern; using attack pattern to share information means that any obstacles of information sharing should be solved and objectives of information sharing should be fulfilled. Thus the objective of this thesis is:

> To support a new usage of attack pattern - an attack data carrier for information sharing and enable consistent comprehension between participants about this attack data carrier during the process of information sharing and decision making.

The objective will be reached by adding semantics to terms and explicitly specify these terms in classes, properties, facets of these properties and ways how classes relate to each other, which will end up with the deliverable of this thesis: a shareable attack pattern ontology. This ontology can be the solution for both of the two knowledge gaps. For the common understanding knowledge gap, an ontology is to be shared by all information-sharing participants that it is the base knowledge and the common language. For the consideration of sharing knowledge gap, one of the main contributions of ontology is to support information sharing.

The role of this ontology is to unify and formalise attack pattern knowledge that is exchanged and shared. It ensures all users speak a common language that both humans and machines comprehend the shared data consistently. Although the machines do not truly understand the information shared, they can effectively manipulate the terms according to the rules and relations defined by the ontology (Berners-Lee, Hendler, & Lassila, 2001). The ontology can be used as a general vocabulary, roadmap and extensible dictionary of the domain of attack pattern. This ontology is supposed to be comprehensive enough that it can be fitted and extended for all information sharing communities whatever the characteristics of the communities are; so one single organisation does not need to learn multiple 'languages' to be able to understand and use attack pattern information from multiple information sharing communities.

2.3 Research scope

This section describes the scope of the thesis to support cybersecurity information sharing between organisations. In Figure 1, sharing cybersecurity information among organisations is broken down into 4 aspects; the entities highlighted by orange lines show the research purpose and scope.

- Cyberattack information can be shared in multiple approaches: within an organisation, between different organisations or between different legal bodies (The TRESPASS Project D5.3.1., 2013). As our focus is information sharing, it is mainly in the 'between organisations' level.
- The geographical scope is mainly on EU, especially for the Netherlands.
- No matter for which approach, the shared content can include threat information, attack information to other kinds of cybersecurity information. Based on different abstraction level, attack information can be further categorized into attack incident and attack pattern. This thesis aims at sharing information between organisations on the basis of attack pattern.
- Our focus is the problem of *how* to share, so it deals mainly the issues during sharing; other related problems will not be discussed in depth, for example why organisations want to share cybersecurity information, and what benefit will organisations gain after sharing.

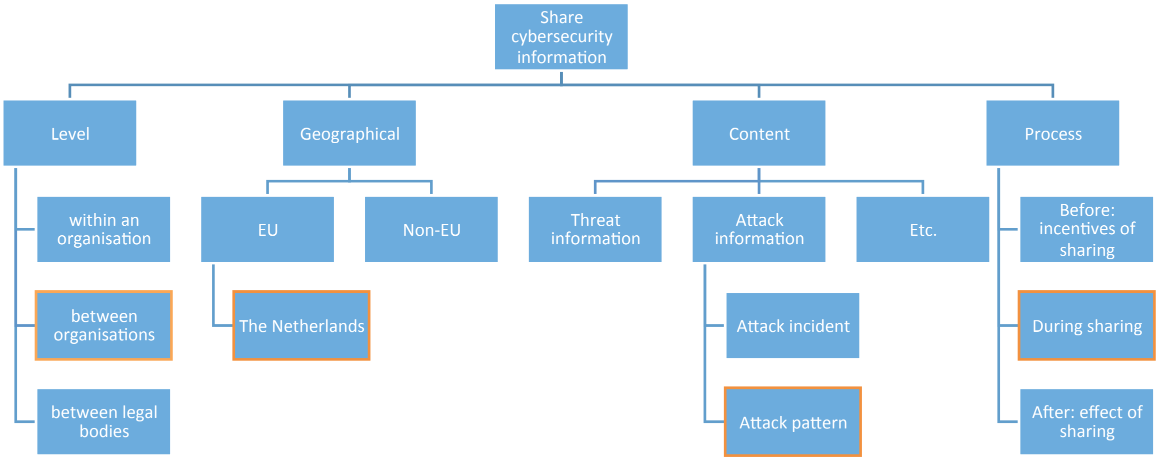


Figure 1 Scope of this research

2.4 Study's relevance

In this section, study's relevance is analysed from two different perspectives: the scientific perspective and the social perspective.

The scientific relevance

This research proposed a new way of security information sharing – sharing attack pattern. It addresses the knowledge gaps of 1. Lack of common understanding of the attack pattern concept and 2. Lack of consideration of sharing.

To support this new way of information sharing, this thesis introduces new ontology based on existing researches and attack pattern enumeration to structure attack pattern information. This ontology explicitly specifies the attack pattern concept; it also defines the scope and contents of attack pattern that are consistent with most of the existing researches. The ontology provides a basic semantic web with proper hierarchical depth and abundant relations between entities. According to each user's different purposes and requirements, the ontology can be personalised in various ways that is not capable when adopting relational or hierarchical data models. In addition, this ontology is also applicable for representing single cyber attack. The only unfitness is that presenting a single attack incident in the form of attack pattern will lose the feature of attack pattern: objectiveness and independency from the victim's context. Comparing with existing cyber attack ontologies, the ontology delivered by this research does not focus on presenting one or few aspects of the cyber attacks, instead it balances the emphasis on all relevant aspects.

Furthermore, this thesis involves shareability in the attack pattern concept that it targets at sharing attack pattern between organisations. The concerns about sharing and reusing were integrated into the attack pattern information structure.

The social relevance

The ontology helps the communication between people with different viewpoints and translating between systems with different paradigms and languages. Therefore people got the shared understanding as background knowledge to facilitate information sharing. As a consequence of that, organisations can benefit from information sharing and gain advantages such as: gaining better understanding of the security environment, learn from other organisations' experience, prepared for possible attacks to avoid them or to reduce the harm of them.

With a shared language, people gain more accurate cyber attack information from more sources, which in turn support better decision making such as security assessment and improvement, cyber security budget and developing new strategy.

2.5 Research question

Main question:
How to use attack pattern to present attack information in an ontological model for the purpose of unifying and formalising data exchanged and shared?

We want to explore how to use attack pattern to share attack information between different organisations and help these organisations to gain the advantages of reusing attack pattern. Attack pattern is an effective way to reuse previous failure experiences that it captures attacker's perspective and it is generic thus can exclude redundant details. In order to maintain a consistent understanding of the shared information, these organisations should share a common language of the attack pattern concept and the shared attack information. The common language can be built through an ontological model that both humans and machines can comprehend. Therefore by answering this question, we will deliver an attack pattern ontology.

Sub-questions:
1. What are the requirements of sharing attack information that could influence the decision-making on classes, properties and instances of the ontology?

This sub-question tries to define the main users and their requirements in the information sharing activity. It will analyse the various purposes and activities of users in the attack information sharing process.

2. How to create attack pattern ontology in the domain of sharing attack pattern?
This sub-question deals with all the details of building an ontology: the necessary types of elements an ontology should contain, how to define these elements from what resources based on what rules, etc.

3. How effective is attack information sharing using attack pattern ontology?
The last sub-question plays the evaluation role that the ontology should be tested to make sure it can satisfy all the requirements of the users.

Research phase	Sub-questions	Research method	Output
□ Analysis	□ Ontology scope & purpose	□ Desk research; use case model	□ Ontology requirements
□ Design	□ Ontology contents and structure	□ Ontology development methodology	□ OWL ontology built with Protege
□ Evaluation	□ Value of the ontology	□ Interviews and questionnaires with workshop participants	□ Assessment of the ontology

Figure 2 Correspondence between research phase, research question, research method and outputs

2.6 Research method

In order to build the ontology, we will follow the ontology development methodology introduced by Noy & McGuinness (2001) and Uschold & Gruninger (1996); the former one provides more details by giving an example while the later one is more comprehensive in the methodology procedure. After combining these two methodologies, we came up with this 7-step ontology development process:

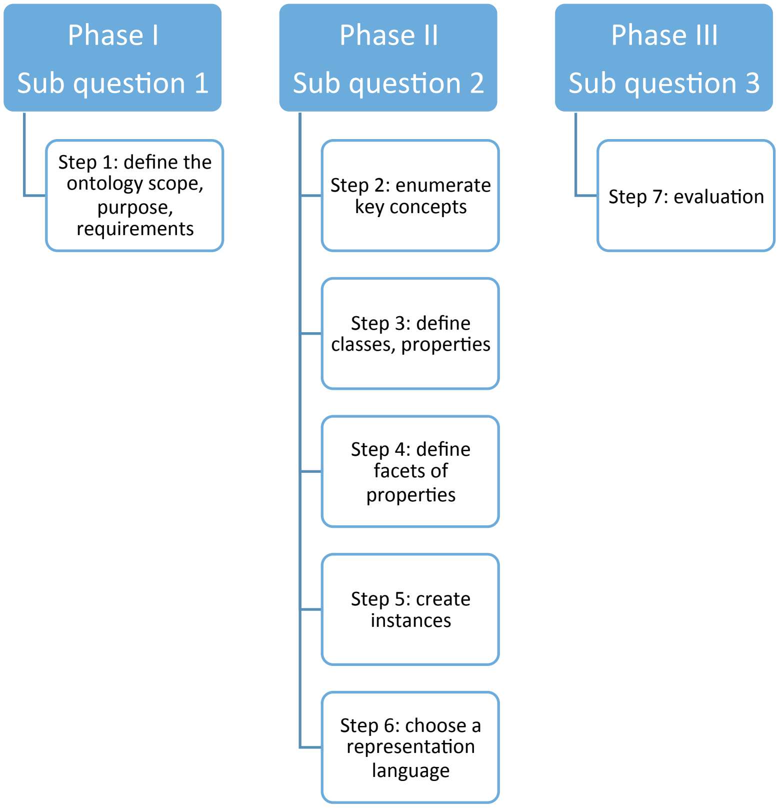

Phase I
Sub question 1
Step 1: define the ontology scope, purpose, requirements
Phase II
Sub question 2
Step 2: enumerate key concepts
Step 3: define classes, properties
Step 4: define facets of properties
Step 5: create instances
Step 6: choose a representation language
Phase III
Sub question 3
Step 7: evaluation

Figure 3 Correspondence between research phases and ontology generation steps

1. Identify the purpose and scope of the ontology

It is accomplished by the first sub-question. However, purpose and scope are answered in this chapter and we add the outputs of this step to ontology requirements. The first sub research question expresses the users' needs and expectation of the attack information sharing system. The ontology shall be built based on this expectation as a solution to satisfy users' needs. We will generate requirements for the ontology through desk research; one important literature is *Guide to Cyber Threat Information Sharing* (Johnson, Badger, & Waltermire, 2014).

2. Enumerate key concepts

This is in the scope of the second sub-question. We will define attack pattern to filter literatures; then we will extract concepts from the definition of 'attack pattern' and the attack pattern templates (Moore et al., 2001; Barnum & Sethi, 2007; Fernandez et al., 2007; Schaeffer-Filho & Hutchison, 2014; CAPEC; etc.).

3. Define classes, properties of classes

   a. Define classes and properties

We need to determine the terminologies of the key concepts. Then based on the scope of the ontology, we decide whether each of the term is class or property.

   b. Build class hierarchy

We will use the combination approach (combination of top-down and bottom-up approach) to build the hierarchy. The method we collect concepts may results in the most general concepts. In order to break the structure down, we can find some specific concepts in existing attack pattern (Moore et al., 2001; Barnum & Sethi, 2007; Fernandez et al., 2007; Schaeffer-Filho & Hutchison;

2014; CAPEC; etc.). We can relate a top-level concept and a specific concept to a middle-level concept. Then we can generate a number of middle-level concepts as the siblings of this middle-level concept. Based on the system requirements, we can further generate more subclasses between the top-level class and the middle-level class or between the middle-level class and the specific concept (a class or an individual). To ensure a comprehensive hierarchy, we will reference multiple sources to finish the add-sibling task.

4. Define facets of the properties – property restriction

Relate the classes with properties and define the property restrictions such as data type and number of values. Property restrictions should also be determined based on the ontology scope. The property restriction choices can be found in Protégé or its tutorial (Horridge *et al.*, 2011).

5. Create instances

We can either reuse the attack pattern instances from the CAPEC (Common Attack Pattern and Enumeration Classification) platform or build a new one based on available data on cyber attack incidents.

6. Choosing a representation language and coding

The ontology editor Protégé can do coding for us; we only need to make choice on language.

7. Evaluation

It is the third research phase and the last step of developing ontology. We will finish it in the third sub-question. To verify the ontology quality, we test the ontology with evaluation criteria. We will invite several fellow students to test the ontology based on these criteria. Despite verification, we will validate the ontology qualitatively based on the criteria proposed by Gangemi et al. (2005) and Tartir, Arpinar, & Sheth (2010).

2.7 Outline of the report

This thesis is further divided into five chapters that are used to answer the research question. Chapter 3 analyses the existing body of knowledge on attack pattern. Prevalent concepts, terminologies and definitions are compared and consolidated into those selected for this study. Chapter 4 analyses the literatures on sharing cyber security information and identifies the requirements for building a sharable attack pattern ontology. The results are used to answer the first sub-question and to develop questions to test the deliverable. Chapter 5 proposes an ontology design for specifying the attack pattern concept based on the results of chapter 3 and 4. The ontology is built with the ontology editor protégé and encoded with ontology language. Chapter 3 and chapter 4 are used to answer the second sub-question and deliver the formally encoded ontology. Chapter 6 evaluates the design of attack pattern ontology for attack information sharing. It tests the design with the competency question and use cases developed in chapter 4. The last sub-question is answered in this chapter. Chapter 7 discusses the potentials of sharing attack pattern with our ontology and analyses the limitations of this design. It uses the findings of the previous chapters to answer the main question. Further more, the chapter describes the study limitation and areas for further research.

Table 1 Report structure

Research stage	Ontology	Chapter	Sub question

	generation step		
Analysis		1 Introduction	
	Step 1	2 Research description	SQ1
	Step 1	3 Background	
	Step 1	4 System specification	
Design	Step 2-3	5.1 Concept identity and structure	SQ2
	Step 4	5.2 Concept relation	
	Step 5	5.3 Instances of the ontology	
	Step 6	5.5 Choose a representation language	
Evaluation	Step 7	6 Evaluation	SQ3
		7 Discussion and conclusion	RQ

3. Background

We have already explained and described our research problem, objective as well as research methods. In this chapter, we will review literatures to present the background of our research, which will be the knowledge base to the answer of SQ 1: What are the requirements of sharing attack information that could influence the decision-making on classes, properties and instances of the ontology?

The first section introduces attack pattern. We separate the background of attack pattern into 4 parts: the concept of attack, the concept of pattern, the concept of attack pattern in practice, the concept of attack pattern in theories. The second section focuses on ontology; we mentioned four aspects of ontology: the concept of ontology and what does an ontology looks like; the reason of using ontology in information sharing and its advantages over other data models; existing ontology development methodologies and the methodologies used in this paper; the tool this research uses for developing the attack pattern ontology.

3.1 Attack pattern: definition, concepts and description

Before an attack pattern can be built, the details of attack and pattern should be explained. These two concepts construct the core concept of our attack pattern ontology. We first introduce the definitions of these two concepts as the background of our design. Then give an overview of attack pattern in practice and theories. CAPEC is the only source where a comprehensive attack pattern database can be found. In the articles that talked about attack pattern, each gives none or few attack pattern instances. However this articles provide detailed definition and description about the attack pattern concept.

The literature used here is found by doing a systematic literature research. Various different research articles were gathered by using online search databases of Scopus, Google Scholar, Springer and Science Direct. Search terms that were used were based on different combination of the following keywords: 'attack pattern, 'cyber', 'attack', and 'pattern'. Criteria for selecting proper articles out of the search query results were based on the specific relevance to the subject.

3.1.1 Attack

International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC) has defined attack as: 'any unauthorized attempt to access, use, alter, expose, steal, disable, or destroy an asset'.

National Institute of Standards and Technology (NIST) is a measurement standards laboratory, which is also an U.S. federal organisation. NIST's special publication SP 800-32 defines attack to be: 'an attempt to gain unauthorized access to system services, resources, or information, or an attempt to compromise system integrity' (Kuhn, 2001).

Both the two definitions described the malicious intention of attacks to cause harmful outcomes. Combining the two definitions, we come up with a more detailed definition that attack is:

> A malicious attempt to gain unauthorized access to system services, resources, or information for the purpose of using, altering, exposing, stealing, disabling, or destroying an asset

3.1.2 Pattern

The Merriam-Webster English Dictionary defines pattern as, 'a repeated form or design especially that is used to decorate something; the regular and repeated way in which something happens or is done; something that happens in a regular and repeated way'. From these definitions, we can extract the core concept of pattern that patterns are 'regular' and 'repeated'; what has been captured in the pattern today can happen again tomorrow. Therefore people build patterns to encapsulate and reuse knowledge (Schaeffer-Filho & Hutchison, 2014; Uzunov & Fernandez, 2014).

In computer science, patterns are general, reusable solutions to commonly occurring problems (Bayley, 2014). Pattern has been found useful in diverse areas including software engineering, where this concept has received much attention both in academia and industry (Uzunov & Fernandez, 2014). The idea of pattern was originated in architecture from Christopher Alexander's architectural patterns for architecture design (Alexander, Ishikawa, & Silverstein, 1977). Then, it was transferred to software design as design patterns in the book Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson, & Vlissides, 1995). Following design pattern, attack pattern and security pattern were also introduced to the cyber security domain (Bayley, 2014).

3.1.3 Attack pattern in practice

CAPEC (Common Attack Pattern Enumeration and Classification) is an open data resource that provides a comprehensive dictionary of known attacks (Mitre Corporation, n.d.). It aims at identifying and understanding attacks, which is more focused for academia (WASC, 2010; Mitre Corporation, n.d.). CAPEC adopts several perspectives to present attacks: by hierarchical representation, by relationship to external factors, by relationship to specific attributes. The hierarchical representation, which includes two logics (16 mechanisms of attack and 6 domains of attack), is the main navigation method to the CAPEC dictionary; it covers most of the attack pattern records (category 286 and its subclasses are excluded) and is the only perspective showed on the home page of CAPEC. However, these two logics are parallel and not connected; they can be presented as two separate attack trees with the same root of cyber attack. In CAPEC, attack patterns have 3 different completeness levels: hook, stub, complete; and 3 different abstraction levels: meta, standard, detailed (Mitre Corporation, 2014). The abstraction level shows a hierarchical structure of attack pattern that the hierarchy often starts with a category, followed by a standard or meta attack pattern and ends with a detailed attack pattern (Mitre Corporation, n.d.a).

CAPEC has the following disadvantages in presenting shared knowledge:

- The classification of attack pattern is disorganised and not mutually exclusive. For example, CAPEC-13 is the child of both the 'Exploitation of Authorization' and the 'Manipulate Resources' mechanisms of attack, but it is not the child of any of the six domains of attack.
- The abstraction level is not implemented as designed following the sequence of meta attack pattern, standard attack pattern, detailed attack pattern. For instance, attack pattern in the social engineering category and the physical security category are all in the meta level; only one detailed attack pattern exist in the supply chain category, others are all in the standard level; attack pattern in the software category are in an unorganised status that a standard pattern (CAPEC-20) can has a meta pattern child (CAPEC-97).
- A top-down approach is used to produce attack pattern rather than focusing on gathering evidence from multiple datasets to identify pattern (Schaeffer-Filho & Hutchison, 2014). Thus several classes are too broach such as 'abuse of functionality' while the next level classes would be too specific, for example 'WSDL scanning' that applies only to system based on web-services (Uzunov & Fernandez, 2014).

3.1.4 Attack pattern in theories

We reviewed twelve articles that introduced the concept of attack pattern. It is an emerging research topic that focuses on attack modelling for cyber security. In 2001, the term attack pattern was introduced in the paper Attack Modeling for Information Security and Survivability (Moore et al., 2001). Three years later, it was extended and enriched in greater detail and with a solid set of specific examples in the book Exploiting Software: How to Break Code (Hoglund & McGraw, 2004). Since then, several individuals and groups have tried to push the concept forward (Barnum & Sethi, 2007).

Some conflicts and ambiguity exist in the attack pattern definitions. As can be seen from Table 2, the works of Thonnard & Dacier (2008), Zhu (2011) and Huang et al. (2013) are describing something different to other definitions. The attack pattern defined by Thonnard & Dacier (2008) is a time signature, which is a time series (figure based) that show the 'aggregated source count for a given type of attack'. This definition describes a pattern of number of attacks on the scale of time, but not a pattern of attacks. Zhu (2011) defines attack pattern as a sequence of attacks; it presents only the steps and execution flow of an attack, which does not capture attacker's perspective or show why attacker followed such steps. Huang et al. (2013) also relate attack with time; they define attack pattern as the regularity of time intervals between attacks.

Although the rest ten definitions are similar (including the definition of the CAPEC's), some small differences exist. For example, attacker's perspective was not emphasised in all the definitions; the scope of attack types that attack pattern focus on various from software attacks (Hoglund & McGraw, 2004; Barnum & Sethi, 2007; Pauli & Engebretson, 2008) to general cyber attacks. Summarising from the rest of the definitions, for our work, we use the following definition of attack pattern:

A blueprint or generic representation describing how to perform and execute attack from the point of view of an attacker. It presents the critical features of the exploited vulnerability, the knowledge required for an attacker to perform the attack, the steps to perform the attack and the ways to counteract the development of the attack.

Table 2 Definition of attack pattern

Definition	Source
'A generic representation of a deliberate, malicious attack that commonly occurs in specific contexts'	Moore *et al.,* 2001
'An attack pattern is a blueprint for exploiting a software vulnerability. As such, an attack pattern describes several critical features of the vulnerability and arms an attacker with the knowledge required to exploit the target system.'	Hoglund & McGraw, 2004
'A mechanism to capture and communicate the attacker's perspective. Attack patterns are descriptions of common methods for exploiting software.'	Barnum & Sethi, 2007
'An attack pattern is presented from the point of view of an attacker. It specifies a generic way of performing an attack that takes advantage of specific vulnerabilities in a certain environment. The pattern also presents a way to counteract the development of the attack in the form of security patterns and to analyze the information collected at each stage of the attack. '	Fernandez *et al.,* 2007
'A model that describes how to execute an attack. Just as design patterns show the foundational details of object-oriented designs that allow one to build new systems by inheriting the information captured in the pattern, attack patterns abstract the basic properties of an attack to identify where vulnerabilities may be present.'	Gegick & Williams, 2007
'an attack time signature of a specific attack such as a worm or a type of botnet attack'	Thonnard & Dacier, 2008
'Attack Pattern is a high level blueprint that describes various types of software attacks.'	Pauli & Engebretson, 2008
'a sequence of attacks that could be of various types but correlated to the security breach'	Zhu, 2011
'An attack pattern is analogous to a design pattern, as it describes how a particular type of attack is performed as a conceptual pattern. However, attack patterns are specifided from the attacker's viewpoint and therefore represent undesirable, unintentional and unexpected operational behaviour'	Blackwell, 2012
'the association found in the time interval between two of the same type of attack on a sensor node'	Huang, Liao, Chung, & Chen, 2013
'Attack patterns capture the steps required to perform a specific security attack (exploit) in a generic fashion'	Uzunov & Fernandez, 2014

‘Attack pattern solves the problems of those wishing to compromise information security’	Bayley, 2014
‘An attack pattern is an abstraction mechanism for helping describe how an attack against vulnerable cyber-enabled capabilities is executed.’	CAPEC

Four of the review articles provided the specific information that should be captured by attack pattern (Moore *et al.,* 2001; Barnum & Sethi, 2007; Fernandez *et al.,* 2007; Blackwell, 2012). Moore *et al.* (2001) proposed the simplest template that one attack pattern captures only five attributes: name, goal, precondition, attack and postcondition. Blackwell (2012) provided the most complex template with more than ten attributes. These templates were listed and compared together with the template provided by CAPEC in Table 3. Each one of the five columns present the items from one source. The items in the same row refer to the same or similar concept (maybe not similar terms). All the attribute names are the original names from the information source; no change has been done to them. For instance, ‘attack’ (Moore et al., 2001) from defenders’ view and ‘solution’ (Fernandez et al., 2007) from the attacker’s view refers to the same attribute of attack pattern. Blank cells indicate that this information source does not have this attribute. Among these sources, the CAPEC attack pattern schema list developed by Mitre Corporate (2014) layered the attributes into required, suggested and optional. We only show the required attributes in the table below; suggested and optional attributes (over 50) are not listed here.

Table 3 Attack pattern attributes comparison

Moore et al. (2001)	Barnum & Sethi (2007)	Fernandez et al. (2007)	Blackwell (2012)	Mitre Corporate (2014)
Name	Pattern name and classification	Name	Name, Classifier	Name, ID
Goal	Attack motivation-consequences	Problem	Perpetrator (who)	Summary
			Motivation (why)	
		Intent	Intent (what)	
			Target (to what)	
Postcondition	Attack motivation-consequences	Consequences	Security or immediate impact	
			Security or ultimate impact	
	Solutions and Mitigations	Countermeasures and forensics		
Attack	Description	Solution	Execution (how)	Summary , Description
			Process diagram	
	Method of attack		Methods (with what)	
Precondition	Attack prerequisites	Context	Context or prerequisites (when)	
	Resources required		Resources (with what)	
	Attacker skill or Knowledge required		Attacker skill (internal with what)	

			Attacker knowledge (know what)	
	Related vulnerabilities or weaknesses			
	References	Known uses	Reference	
		Related patterns		Related patterns
				Pattern completeness
				Pattern abstraction
				Status

Source: Moore et al. (2001); Barnum & Sethi (2007); Fernandez et al. (2007); Blackwell (2012); Mitre Corporation (2014)

Only Barnum & Sethi (2007) described the process of generating attack pattern; other articles just simply provided attack pattern examples to show how the templates work. According to Barnum & Sethi (2007), when one particular attack is being reported many times and not matches the existing attack pattern (public knowledge), people can discover the cause of the attack and build a new attack pattern to describe such type of attacks.

Attack pattern can be used in multiple ways. However, all the articles treated attack pattern only as an approach to present knowledge, but did not think about the possible of using it as a means to share information. According to the articles, attack pattern can educate people about common attacks that make future threat modelling tasks easier (Uzunov & Fernandez, 2014). It can also be used for identifying the potential vulnerabilities and attacks that are applicable to one system (Moore *et al.,* 2001; Gegick & Williams, 2007), which then guide the design process of a system and support the judgements about possible design solutions (Barnum & Sethi, 2007; Faily *et al.,* 2012). It can be seen as data source as well to find evidence of attacks (Fernandez *et al.,* 2007).

Therefore, in addition to presenting knowledge, existing researches missed the possibility of using attack pattern as a means of sharing information outside an organisation. Articles that talked about attack pattern usages transitioning from 'what is attack pattern' directly to 'how to use attack pattern' without exception. They did not mention the concerns of reusing attack pattern outside an organisation. Maybe the ultimate usages of shared attack pattern have no difference to the usages of non-shared attack pattern, but the effectiveness of shared attack pattern can be much higher. Through information sharing, one organization can alert others of potential attacks and gain critical information to enhance their own defences. In this paper, we are going to analyse what is missed in existing researches and fill in the knowledge gaps:

- For a common language of sharing attack pattern, what should be changed to the existing attack pattern concepts. In order to answer this, we have to give answers to the next question beforehand:
- How to align the existing attack pattern concepts to a single version

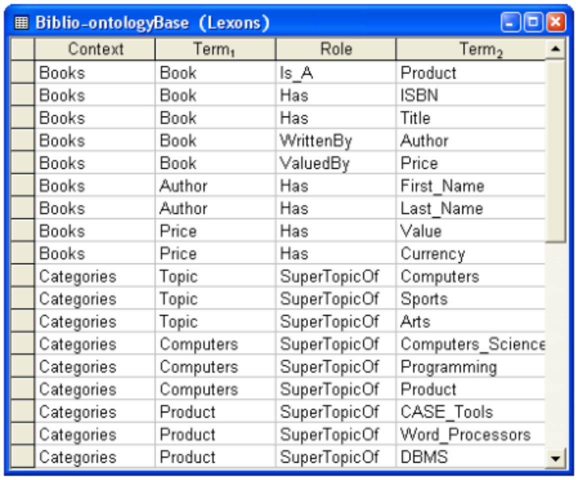
3.2 Ontology and its role in information sharing

In the first part of this chapter, we introduce ontology from several aspects, including the definition, the methodologies of developing ontology, what does ontology looks like and what are the functionalities of ontology that other data models are not capable of during information sharing.

3.2.1 Ontology: introduction

Ontology is a semantic web model to provide a common language of a domain of knowledge that is exchanged and shared (Uschold & Gruninger, 1996; Noy & McGuinness, 2001). It gives a description of entities and their properties, relationships, constraints (Gruninger & Fox, 1995). At the beginning of the 1990s, ontologies were mainly built using artificial intelligence modelling techniques based on frames and first order logic using the components introduced by Gruber (1993): concepts, relations, functions, axioms and instances (Gómez-Pérez, 2004). Now various AI-based ontology implementation languages have been created including SHOE, XOL, OIL, DAML+OIL and OWL (Gómez-Pérez, 2004). Numerous software tools are available for building ontologies such as Apelon DTS, DOME, FlexViz, Knoodl, Protégé and TopBraid Composer.

Many articles developed ontologies for various intentions. We mention three here that are relevant to our attack pattern ontology. Jarrar, Demey, & Meersman (2003) decompose an ontology into an ontology base and a set of ontological commitments to enable application independency. The ontology base holds intuitive domain knowledge whereas the ontological commitments hold application specific knowledge (Jarrar, Demey, & Meersman, 2003). An example of an ontology base is showed below in Figure 4. The two columns about term show classes, the third column about role shows relations between the instances in the classes in the second and the fourth column. What this research develops is also an ontology base that used to describe attack pattern.



Context	Term$_1$	Role	Term$_2$
Books	Book	Is_A	Product
Books	Book	Has	ISBN
Books	Book	Has	Title
Books	Book	WrittenBy	Author
Books	Book	ValuedBy	Price
Books	Author	Has	First_Name
Books	Author	Has	Last_Name
Books	Price	Has	Value
Books	Price	Has	Currency
Categories	Topic	SuperTopicOf	Computers
Categories	Topic	SuperTopicOf	Sports
Categories	Topic	SuperTopicOf	Arts
Categories	Computers	SuperTopicOf	Computers_Science
Categories	Computers	SuperTopicOf	Programming
Categories	Computers	SuperTopicOf	Product
Categories	Product	SuperTopicOf	CASE_Tools
Categories	Product	SuperTopicOf	Word_Processors
Categories	Product	SuperTopicOf	DBMS

Figure 4 An example of an ontology base

Source: Jarrar, M., Demey, J., & Meersman, R. (2003). On using conceptual data modeling for ontology engineering. In *Journal on data semantics i* (pp. 185-207). Springer Berlin Heidelberg.

Kim, Luo, & Kang (2005) create a security ontology set consists of seven related ontologies to annotate the functional aspects of resources: main security ontology, service security ontology, information object ontology, agent security ontology, security algorithms, credentials ontology, security assurance ontology. It is written in OWL language and can be applied to any electronic resource (Kim *et al.*, 2005). A graphical representation of the main security ontology is showed below:

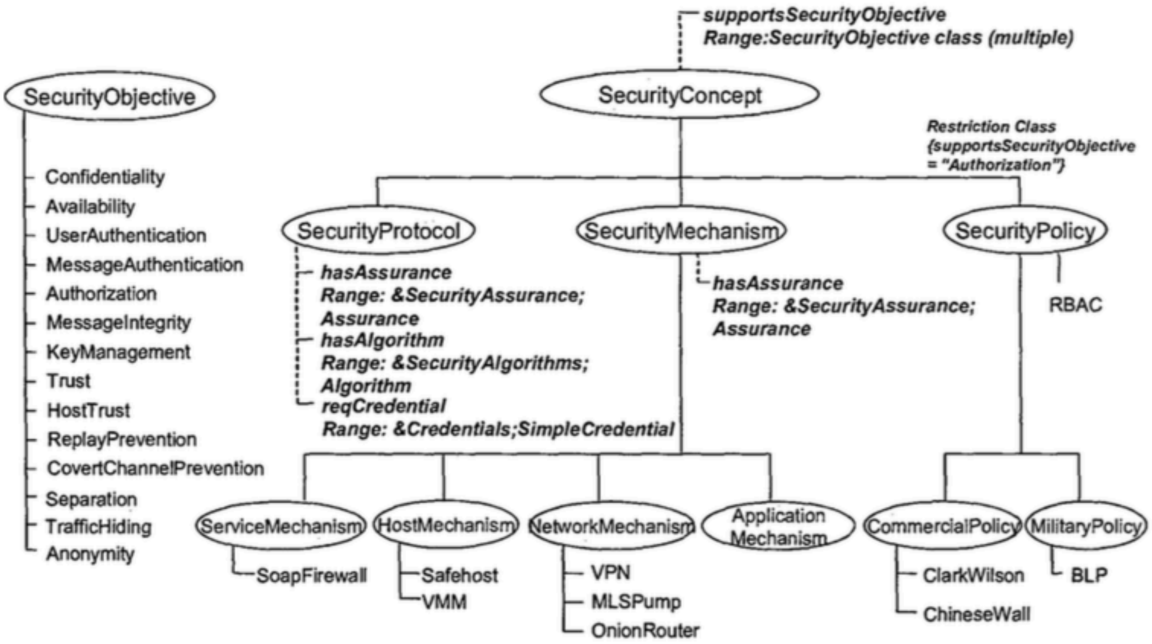


Figure 5 Main security ontology of Kim *et al.*, 2005

Source: Kim, A., Luo, J., & Kang, M. (2005). *Security ontology for annotating resources* (pp. 1483-1499). Springer Berlin Heidelberg.

Foley & Fitzgerald (2011) propose a semantic threat graph approach to manage security policy configuration. The semantic graph is used in the form of ontology that it extends threat trees with implicit concepts, individuals and relationships (Foley & Fitzgerald, 2011). This extension aims at relating semantic information about security configuration with threats, vulnerabilities and countermeasures (Foley & Fitzgerald, 2011). The ontology developing process can be seen from Figure 6, concepts and relationships were added on the basis of a threat tree.

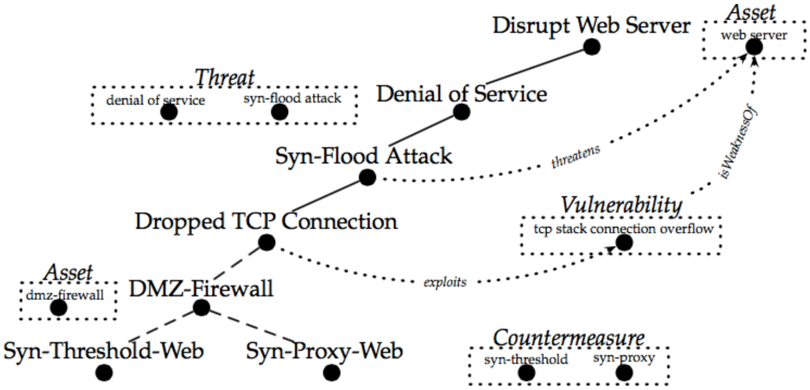


Figure 6 Extended threat trees with implicit concepts, individuals and relationships

Source: Foley, S. N., & Fitzgerald, W. M. (2011). Management of security policy configuration using a Semantic Threat Graph approach. *Journal of Computer Security*, *19*(3), 567-605.

3.2.2 Ontology-based vs. non-ontology based approach

Before illustrating the necessity of adopting semantic modelling / ontology, it is best to have a review of some popular, mainstream approaches to modelling data. In this section, we compare ontology, database schema and taxonomy to show how they contribute to information sharing.

In order to share data, all the user need to define and share a common vocabulary to describe the contextually consistent data. Thus they will share the same understanding of the content in the shared database. Compared with ontological model, taxonomy and database schema are not suitable for communication between complex systems. Taxonomy describes and classifies resources based on hierarchical relationships among entities; but it cannot provide contextual information or rich meaning of these concepts that further defines restrictions and interdependencies among concepts (Kim et al., 2005). Moreover, because hierarchical relation is the only kind of relation that connect elements in taxonomy, taxonomy is not able to provide class-based reasoning such as automatically classification. When adopting database schema, new primary keys are necessary that primary keys in two different databases cannot be synchronised. For example, two countermeasure datasets can be linked through the 'Measurement_id' primary key, but these IDs refer to different countermeasures. So only the chosen data can be shared with the new primary keys; if a third database is added, the primary keys need to be defined again ('Tutorial 3', n.d.).

An ontological model can express and interpret the meaning behind the data through detailed definition and description. Thus in our attack pattern ontology, all information about attack pattern is related; people can find information via the linked standard terminology without even knowing the existence of the information ('Tutorial 3', n.d.). Moreover, this happens without the need for transformation or mapping between the two sites; it is all settled through semantics ('Tutorial 3', n.d.). An ontology helps the integration of information from different sources with the least deviation from the origin semantics. To summarise, compared with non-ontological approaches, an ontology approach has the following features:

- Different types of relations can be added between any two elements (compared with the sole hierarchical relation in taxonomy and the table-to-table connection in database)

- Reasoning and automatically classification.
- Semantics can be added to data for further specification.

Hence, to share information that have complex relations between concepts as our attack pattern case, we choose ontology to represent the base knowledge.

3.2.3 Ontology development methodology

There are various ontology development methodologies within different disciplines. Caracciolo (2006) designs a methodology to build an ontology for logic and linguistics. This work integrates a set of hierarchical relations with two non-hierarchical relations to enable an explicit navigation (Caracciolo, 2006). Ontology Development 101 is a guide to create ontologies for beginners where 7-step process is introduced and an ontology of wine and food is developed (Noy & McGuinness, 2001). METHONTOLOGY is a chemical ontology building methodology that focuses on the reuse of ontologies (López, Gómez-Pérez, Sierra, & Sierra, 1999). According to METHONTOLOGY, most of the evaluation work of the ontology should be carried out in the conceptualisation stage to prevent errors in implementation. Uschold & Gruninger (1996) introduced the principles and methods of developing ontologies for knowledge engineering. The article intends to introduce the design and use of ontology as a shared understanding to improve communication among people, organisations and software systems (Uschold & Gruninger, 1996).

In this paper, we apply the combination of the ontology development methodology introduced by Noy & McGuinness (2001) and Uschold & Gruninger (1996). Noy & McGuinness (2001) provided detailed explanation of what to do in each step whereas it omitted some major steps mentioned by Uschold & Gruninger (1996) including the choice of ontology language and the evaluation. Therefore we mainly follow the process in Ontology Development 101 but add the steps of choosing ontology language and evaluation.

The shortcomings of existing works about attack pattern will be overcome through the seven ontology development steps:

- Gap 1 no common understanding: In step 2 and 3, summarising the existing research outcomes, we keep the necessary concepts and leave out redundant concepts
- Gap 2 no consideration about sharing: In step 1, we produce constraints and requirements for information sharing, then implement them in step 2 - 7
- Disadvantage 1 disorganised classification: In step 3, we only generate one structure to classify attack pattern and this structure has pre-defined hierarchy that all users can follow the same classification system
- Disadvantage 2 disorganised abstraction level: abandon the concept of abstraction level
- Disadvantage 3 top-down approach: In step 3, we integrate top-down and bottom-up methods to build a structure of the main concepts

3.2.4 Ontology development tool: Protégé

As mentioned earlier, many ontology development tools exist. Among these tools, we choose Protégé, because it is a free, open-source tool with intuitive user interface. Furthermore, Protégé is written in Java, thus supports running in a wide range of operating systems (Noy et al., 2003).

Protégé allows users to create and edit ontologies in an application area (Noy et al., 2003). It has the building blocks that we expect in developing an ontology: classes, relations and instances. In protégé, these three basic elements are called classes, properties and individuals. For the main building blocks, protégé presents them in a series of 'tabs' where users can enter, search, edit and browse the ontology. From the ontology, the tool automatically constructs a graphical knowledge-acquisition system that support later usage of this ontology in specific applications (Noy et al., 2003). Protégé can record ontologies in various formats including RDF/XML, OWL/XML, N-Triples, N3 and Turtle RDF.

3.3 Conclusion

This chapter analyses the current body of knowledge on information sharing and attack pattern based on literature review. We first dig out the nature of the concept 'attack pattern' by separating it into 'attack' and 'pattern'. Then we introduce attack pattern concepts in practice and in theories. The limitations and disadvantages of the existing knowledge about attack pattern were pointed out for each of the two parts. In addition, in the theory part, we list attack pattern definitions and choose the ones that are corresponding to 'attack' and 'pattern'. We also list and compare 5 different templates for the contents of attack pattern. After that we introduce ontology and discuss its advantages to explain why we will build an ontological model in this paper. At last we mentioned several ontology development methodologies and present the ontology development tool used in this paper. The attack pattern definition and the summarised contents will be used as information input to answer SQ 1 in the next chapter.

4. System specification

The previous chapter introduced ontology and attack pattern. The disadvantages and gaps of the current researches on the topic of information sharing were pointed out. In this chapter, we build a specification for the system of sharing attack pattern where the ontology is used as a common language; the ontology is expected to avoid the disadvantages and fill in the gaps that we mentioned earlier. This chapter answers the first sub-question:

> SQ1: What are the requirements of sharing attack information that could influence the decision-making on classes, properties and instances of the ontology?

To present a comprehensive view of the attack pattern sharing system, we describe not only the specific attack pattern ontology but also the information sharing system as a whole. As showed in Figure 7, we introduce the topics from the outside circles to the inside circles and from higher level to lower level. Only the first point in the inner circle will be discussed in this chapter and the following two topics will be introduced in the next two chapters.

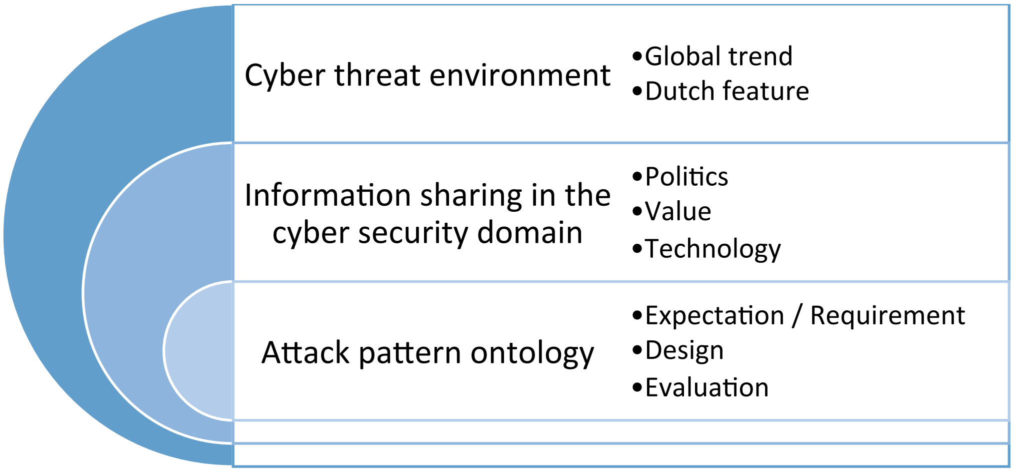


Figure 7 System of attack information sharing

We first introduce the overall threat environment of today's cyber security domain and emphasis the features of the Dutch circumstance. This content has been discussed enough in the introduction chapter thus the cyber threat environment section will be brief and simplified. Then we analyse the information sharing activity from 3 perspectives: politics, values of stakeholders and technology. After that we prosed 4 scenarios where the ontology is needed and introduce some questions that the attack pattern ontology should be able to give answer to. We generate requirements for the attack pattern ontology from two sources: 1) chapter 3. Background, in which we got the definition and content of attack pattern and 2) literature review, from which we collect requirements on cyber security information sharing.

4.1 Cyber threat environment

Cyber security is intertwined with everyone's daily life; citizens, businesses and government bodies are using the Internet for interactions, collaboration and communication (The Minister of Security and Justice, 2013). However, as a consequence of the convenience brought by connectivity, Internet and Internet users are vulnerable to cyber attack (Geers, 2011). Cyber attack nowadays is growing not only in frequency but also in scale scope and complexity (Johnson *et al.,* 2014). The complexity and size of system increase while the number and the skill level of attackers continues to grow (Barnum & Sethi, 2007).

In the Netherlands, Dutch citizens, the government and the businesses are becoming more and more dependent on the Internet and IT (National Cyber Security Centre, 2014). Thus cyber attacks and disruptions have an increasing impact on the lives of people (National Cyber Security Centre, 2014). Compare with the rest of the Europe, the Dutch use the Internet extensively through smartphones and tablets and have a great deal of trust over it (National Cyber Security Centre, 2014). Therefore securing the cyberspace is an important task for the whole Dutch society.

4.2 Information sharing in the cyber security domain

Under the current circumstance of the global and the Dutch cyber security domain, we discuss information sharing for enhancing cyber security from three perspectives: politics, value and technology. In the politics part, we introduce the Dutch politics attitude over sharing cyber security information. Then in the value part, we classify the main roles of stakeholders and analyse their interests and values over sharing cyber security information. Based on the value analysis, we propose an expected process of sharing information that using our attack pattern ontology and explain how the stakeholders interact with each other. At last in the technology part, we present the technical system of sharing attack information that how should information flow between the system and different stakeholders.

4.2.1 Politics

In order to safeguard digital security and freedom and to maintain an open and innovative digital domain, The Minister of Security and Justice of the Netherlands published National Cyber Security Strategy, where cyber security is not viewed isolatable but rather correlated with human rights, privacy, social economics and innovation (The Minister of Security and Justice, 2013). Ten central elements were introduced to reach the objectives of the Dutch government, 3 of which are about information sharing (The Minister of Security and Justice, 2013):

- For the critical infrastructure sectors, the government shall work with vital parties to enhance risk analysis and information sharing
- For the civil and military domains, knowledge and expertise shall be effectively shared between civil parties and the Netherlands Defence organization
- The position of the National Cyber Security Centre (NCSC) shall be enhanced with stronger structure for confidential information sharing and analysis

As can be seen from the National Cyber Security Strategy, the Dutch government treats information sharing as a powerful tool to enhance the national cyber security. Therefore sharing attack pattern follows the trend of the Dutch cyber security development. In the following two sections, we will analyse the attack pattern information sharing from two perspectives: the actor perspective and the engineering perspective.

4.2.2 Value

The prevention of damages caused by cyber attacks is in the interests of individual citizens, businesses and government organisations, therefore in the interests of Dutch society as a whole (National Cyber Security Centre, 2014). In this section, we will analyse each main stakeholder's interests and suggest a way of sharing attack information among organisations.

As showed in Table 4, NCSC clusters actors into the group of victim, attacker or researcher. Victims are those who suffered from cyber attacks and attackers are those who gain from attacking victims. Researcher does not involve in attack incidents directly; they help enhance cyber security by seeking vulnerabilities and exposing weaknesses of ICT environments (National Cyber Security Centre, 2012). The first column lists the main actors; the other 3 columns mark the roles of the actors with 'v'. For example, both government organisations and private organisation can play the role of victim, attacker or researcher. The attacker's role for these two types of organisations is caused by the existence of insiders who have malicious intentions. The last three actors are all attackers: professional criminal, hackvist and activist, script kiddie. The differences between the three are the motivation and knowledge; professional criminal mainly motivated by financial gain while the other two are not, script kiddie has limited knowledge of information security whereas the other two have sufficient knowledge to achieve their goal (National Cyber Security Centre, 2012).

Table 4 Actors and their roles

	Victim	Attacker	Researcher
Government organisation	v	v	v
Private organisation	v	v	v
Individual citizen	v		v
Professional criminal		v	
Hacktivist and activist		v	
Script kiddie		v	

Source: adapted from National Cyber Security Centre (2012)

We further analyse the interdependencies between different actors that in Table 4, the first 3 types of actors are against the last 3 types of actors. When sharing attack pattern, the relationship between the actors are: government organisations are in cooperation with private organisations to defend against cyber attacks from professional criminals, hacktivists and activists and script kiddies so as to protect the information security of organisations and individual citizens.

Government organisations will initiate the attack pattern information sharing and invite private organisations to participate. The choice of the initiator is the consequence of two reasons. First, government organisations have more comprehensive view of the national condition than private organisations, thus they know better the importance of information sharing for the cyber security domain. Second, the private organisations fear of the government organisations to use their vulnerabilities against them (Gal-Or & Ghose, 2004). For government organisations, the intention of sharing information is not only enhancing cyber security, but also gaining information for policy making. An important Dutch government organisation is the National Cyber Security Centre (NCSC). NCSC is responsible to propagate knowledge of ICT vulnerabilities to the government and vital sectors (National Cyber Security Centre, 2013). It brings cyber security expertise of government, industry and academia. The members of the NCSC include several ministries including the Minister of Security and Justice, Economic Affairs, Agriculture and Innovation, the Interior and Kingdom Relations, Defence and Foreign Affairs ('National Cyber Security Centre', n.d.).

The action of sharing attack pattern will be performed around industry sector or some other shared characteristics (Johnson *et al.*, 2014). This is relevant to the strategy of The Minister of Security and Justice that information should be shared within critical infrastructures sectors. The reason is that some organisations are potential victims to targeted attacks, where hackers tailored the attack just for the victim (GFI, 2009); attackers are likely to attack the same target repeatedly until they succeed (Microsoft, 2012). These organisations often face similar adversaries who use common tactics, techniques and procedures that target the same types of systems and information (Johnson *et al.*, 2014). For example, the 2014 data breach investigation report shows that denial of service and POS intrusion are the top two threats for retail organisations; these kinds of attacks occupies around 2/3 of the total incidents from retail organisations (Verizon, 2014). Through sharing threat information, organisations get prepared to be able to act decisively throughout the cyber attack life cycle, to enhance their capabilities and protect themselves as well as their customers (Johnson *et al.*, 2014; Retail Industry Leaders Association, n.d.).

The information sharing will be first performed in a small range between big companies, then introduced to SME (Small and Medium-sized Enterprise) and SMB (Small and Medium-sized Business). As mentioned above, SME and SMB are afraid of disclosing vulnerabilities to both the government organisations and attackers. Moreover, they have severe situation of cyber security expertise shortage. So the first trial will be taken between more mature and experienced private organisations and public organisations ('Terms of Reference', n.d.). The trial result will be used to improve the cooperation process and to prove the effectiveness of the approach to SME and SMB. In the second and following trials, SME and SME can be invited or volunteer to participate.

Figure 8 presents how the process of sharing attack pattern can be formed. It is an overview of the discussion above.

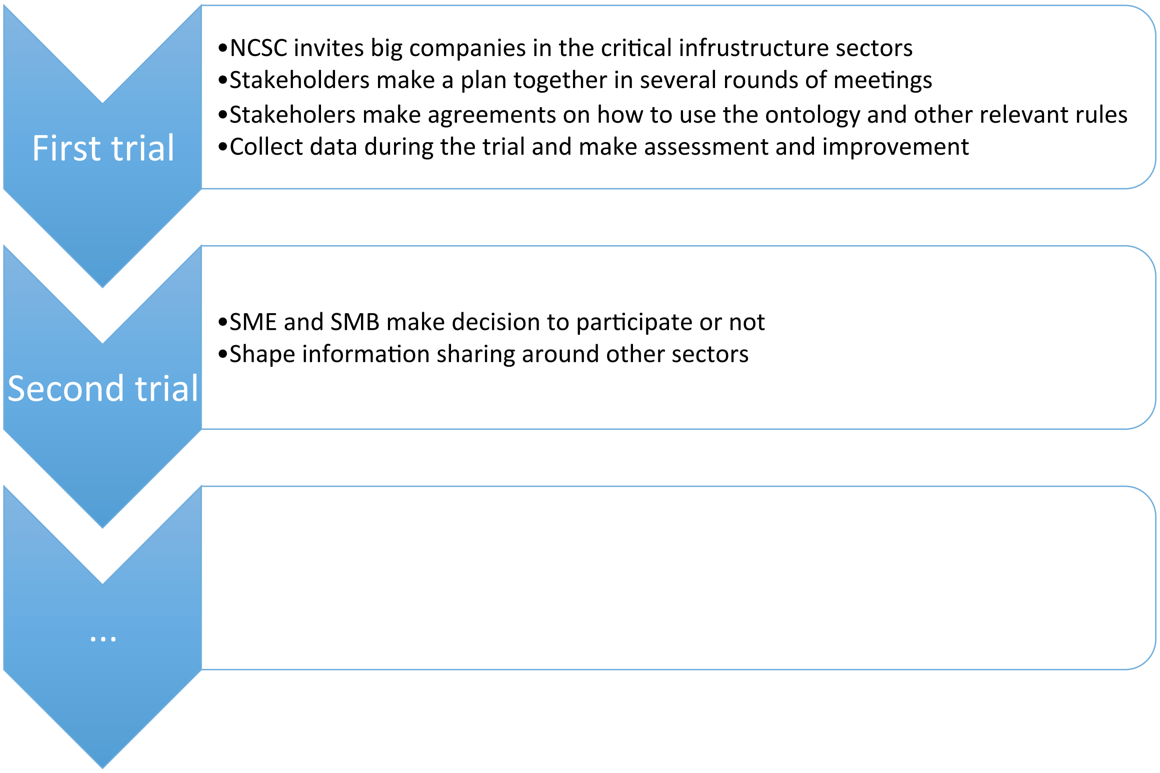

First trial
•NCSC invites big companies in the critical infrustructure sectors
•Stakeholders make a plan together in several rounds of meetings
•Stakeholers make agreements on how to use the ontology and other relevant rules
•Collect data during the trial and make assessment and improvement
Second trial
•SME and SMB make decision to participate or not
•Shape information sharing around other sectors
...

Figure 8 Process of sharing attack pattern

In Figure 8, the first trial will be initiated by government organisations such as NCSC that it invites large companies in the critical infrastructure sector. The first trial will be hold only in the critical infrastructure sectors, which includes the most important sectors for citizens' daily life such as energy, healthcare and food. An overall plan of how to share information in each trial should be made by all the participants as a whole. More specifically, all participants should agree on how to use the ontology and some other relevant rules. For example how to make decision on adding new classes to the ontology. At the end of the trial, all aspects including the attack pattern ontology will be improved based on the experience during the trial. Then in the second trial, SME and SME can be invited. Furthermore, the information sharing community can be formed around organisations that share other kinds of characteristics in addition to in the same industry sector. For instance, instead of sharing attack pattern between organisations in the retailing industry, information can be shared between organisations in the same supply chain, i.e. from the raw material provider to the manufacturer and carrier, at last to the supplier and retailer. These organisations are business partners that may connect through the Internet. Thus it is in the interests of them to share attack pattern and maintain a more secure cooperation environment.

4.2.3 Technology

In this section, we explain how the information sharing system operates once stakeholders reach an agreement on sharing attack information. Figure 9 presents the information flow of the system of sharing attack pattern. Several roles can be recognised: information producer, information provider and information consumer (Vázquez et al., 2012). Information producer is the information source while information provider publishes the information produced.

When sharing attack pattern, information provider is often also the information producer, because companies want to be anonymous in the community and they will not provide their data to another entity that is the information producer. In Figure 9, private organisations are the information provider as well as the information producer. They have multiple ways to produce and provide attack pattern:

- Based on one single organisation's experience, build attack pattern from repeated and regular attack incidents
- Multiple organisations exchange detailed attack data, and based on multiple organisations' experience, build attack pattern from repeated and regular attack incidents
- Based on one single organisation's experience, build attack pattern from single attack incident

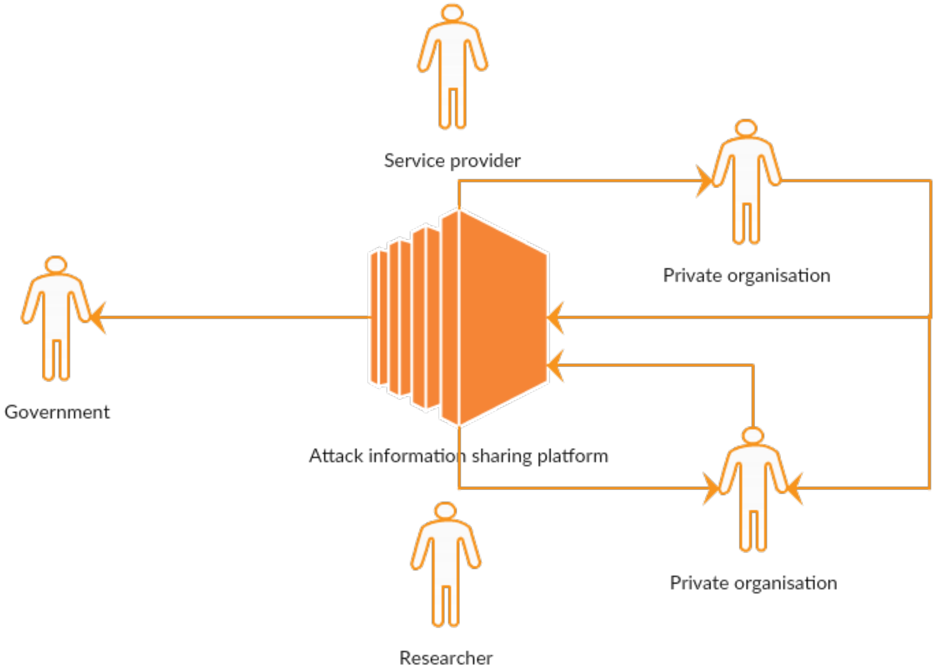


Figure 9 Attack pattern information sharing overview

Both government organisations and private organisations are the information consumer. Although not the member of any industry sector, government organisations can obtain a good overview of each information sharing community of the current situation. Private organisations can learn from each other about the newest attack techniques and corresponding countermeasures.

In addition to the information provider, producer and consumer, there is also service provider who does not participate in information sharing. Instead, service provider operates and maintains the system where the information sharing happens. Another exception is researchers. According to Table 4, researchers come from government organisations, private organisations or individual citizens. They do not contribute to the information sharing process directly, but it may be in their interests to use the shared data. Further details such as who are the researchers and whether they can get access to the shared data or not, will be decided in the stakeholder meeting in the first trial (Figure 8).

4.3 Expectation and requirements of the attack pattern ontology

In this section, we propose expected capabilities and develop requirements of the attack pattern ontology. We first introduce examples of using the ontology in various circumstances in four scenarios. Then from these scenarios, we extract expected capabilities of the ontology and develop questions that the ontology should give answers to – competency questions. At last we convert the questions into statements and produce five requirements for the development of the ontology from the content perspective and the usage perspective.

The requirements are from two sources: 1) chapter 3. Background, in which we got the definition and contents of attack pattern and 2) literature review, from which we collect requirements on cyber security information sharing. The first 2 requirements (content) are based on source 1 – chapter 3 and describe the contents of the ontology; the other 3 requirements (usage) are based on source 2 - literature review and describe the usages of the ontology.

4.3.1 Ontology usage scenarios

In this section, we provide four scenarios to indicate how the attack pattern ontology provides convenience in problem solving and meets people's needs. These scenarios show that an attack pattern ontology can be applied in different industrial sectors for different usages. All these different usages help organisations to more accurately make decisions. From these scenarios, we can see some expected capabilities of the ontology that is helpful for generating competency questions.

Scenario 1 Security assessment through simulation
A hospital just updated its security system. It wants to assess its current security level and compare the current system with the old one. The ability of successfully defend against cyber attacks is an indicator to the security level. The hospital will simulate if the current security system would do better than the old one facing the same or similar cyber attacks happened in the last five years. The simulations should run without manual assistance, thus the computer should be able to understand the preconditions of the attacks, the situation of the current security system and the old security system. Therefore, the simulation needs 2 ontologies: one ontology describes the artefact being assessed and the other ontology describes how the system can be attacked. To decrease redundant work and save time spent on simulation, they apply an attack pattern ontology to describe former attacks. As a consequence of that, the total number of simulation will be significantly decreased compared with simulating all former attack incidents one by one. Based on the simulation results, the security system can be assessed and further improved.

Scenario 2 Decision-making on security investment
A car manufacturing company wants to optimise its information security budget. An important consideration is staying up to date with competing companies. This car manufacturing company

uses SCADA system to manage parts inventory for just-in-time manufacturing, which is commonly used in the manufacturing industry. In order to make an objective decision on the future security investment in the SCADA system, the car manufacturing company needs to know the regular malicious attacks and compare its security countermeasures with other manufacturing companies. Hence they need two types of information: regularly happened attacks and countermeasures for SCADA systems used in the manufacturing industry, countermeasures that achievable for their own company. The first type of information can be described by the attack pattern ontology, the second type can be described by another ontology. Computers can understand the meanings described in ontology, thus the comparison can be done with much less manual work.

Scenario 3 Employee training on cyber security
A restaurant wants to increase its staff's consciousness on cyber security through education. According to its experiences, improper behaviour of staff could lead to information security vulnerabilities. To prevent the same or similar attacks happen again, including the ones that happened in this restaurant as well as in other restaurants, this restaurant wants to show its employees misbehaviours the severity of misbehaviour. Therefore it needs to know all the misbehaviour related attacks and consequences in the restaurant industry. An easy way to extract the needed information is to make all the information recorded readable by computer. An ontology that records the attack consequences and exploited vulnerabilities in the restaurant industry will do the work.

Scenario 4 Policy and standard generation
A software development company wants to generate security standards and policies. It is easier to describe how software can be abused than explaining how to build secure software. Thus, this company develops policy and standard based on cyber attack information that targeted on its software products. The company needs to know the vulnerabilities and weaknesses of the products and the possible mitigations. Therefore either adequate attack data or attack pattern is the basis of generating secure standards and policies. For each software product, numerous attack patterns could be applicable. Hence a smart way of documenting attack or attack pattern should be employed that the computer can understand the information recorded - ontology.

From the four scenarios above, we can see some similarities of the ontology: 1) no matter how different the industry sectors are, it is able to present attack information for that sector; 2) it should present some important information of an attack such as countermeasure, attack consequence and vulnerability; 3) the ontology supports query.

4.3.2 Competency question

In this section, we define the ontology requirements in the form of specific questions that people expect the ontology to answer (Grüninger & Fox, 1995; Oberst et al., 2007). These questions are called 'competency questions' (CQ) because they show the competency of the ontology (Grüninger & Fox, 1995). Corresponding to the inner circle of Figure 7, competency question is the beginning of developing the ontology. These competency questions will serve as test criteria for ontology evaluation later (Noy & McGuinness, 2001; Gómez-Pérez, 2004; Oberst et al., 2007).

Now we convert the abstract expectation of the ontology into concrete questions that the attack pattern ontology should be able to give answers to. These questions do not cluster all the potential requirements of the ontology but rather a sketch of the ontology (Noy & McGuinness, 2001). For the first expectation, applicable for different industries, a possible competency question is:

> CQ1: What are all the types of countermeasure, attack consequence, vulnerability, etc. for cyber attack?
>
> For the second expectation, present important information of attack, a possible competency question is:
>
> CQ2: What should be filled for an attack pattern?
>
> For the third expectation, query, a possible competency question is:
>
> CQ3: How are the important attributes (countermeasure, attack consequence, vulnerability, etc.) of attack pattern related?

4.3.3 Content requirements

Corresponding to CQ2 What should be filled for an attack pattern, we produce the first content requirement, **Requirement 1: The attack pattern ontology should contain enough but not redundant concepts that can describe attack pattern.**

This requirement fills in the second gap of existing researches that we mentioned earlier in 3.1.4 Attack pattern in theories: How to align the existing attack pattern concepts to a single version. We aim at integrating existing versions of series of concepts that used to describe attack pattern and build one single concept series. Requirement 1 expresses the way to build such single version.

Another requirement on attack pattern content is **Requirement 2: In the attack pattern ontology, one attack is captured in only one attack pattern.** This requirement is corresponding to the first gap of existing researches that we mentioned earlier in 3.1.4 Attack pattern in theories: For a common language of sharing attack pattern, what should be changed to the existing attack pattern concepts.

According to the current studies, one cyberattack phenomenon might corresponding to multiple correlated patterns and one attack pattern only captures one specific part of an attack (Mitre Corporation, n.d.). For example, an attacker exploit attack pattern A for reconnaissance, attack pattern B for accessing the system, attack pattern C for causing damage. Attack pattern A, B and C are different but correlated because of both A and B help to satisfy the preconditions of C (Moore et al., 2001). We also found evidence about this attack pattern – attack relationship from the CAPEC (Common Attack Pattern and Enumeration Classification) effort. CAPEC aims at identifying and understanding attacks, which is more focused for academia (WASC, 2010; Mitre Corporation, n.d.); separate one attack into multiple patterns will definitely decrease the complexity of attack pattern and thus easier for identifying and understanding attacks. However, this attack pattern-attack relation is not clarified in the existing literatures, because many of them provide attack pattern examples on exploiting software vulnerabilities (Moore et al., 2001; Barum & Sethi, 2007; Fernandez *et al.,* 2007), which is attack pattern in the narrow sense – attack pattern C that causing damage. These articles suppose that the preconditions of their software attack pattern are already been met; they can thus only focusing on the last step.

However, identifying and understanding attacks is not the main goal for companies and government to share attack information. They want to alert others of potential attacks and gain critical information to enhance their cyber security (Johnson *et al.,* 2014). So the information shared is not only the attack mechanisms but also the attacker motivation and attacker's choices on vulnerabilities and attack methods. Thus they need the complete attack description to present these choices and the reasons behind the choices. However, when we fragment one attack into multiple attack patterns, little value remains for people trying to reach their goals such as gaining better understanding about the threat environment and increasing the situational awareness of the community. Both from the attacker's view or the defender's view, an attack description is meaningful when it shows the overall attack path. There is no necessity to separate one attack into several patterns in this case.

Therefore divergent from the attack pattern concept of current studies, in our attack information sharing system, an attack pattern captures all the parts of an attack, which is equivalent to the combination of multiple attack patterns defined by existing studies. But the corresponding relation from attack to attack pattern is the same to the current studies; each pattern encapsulates the features of various datasets (Fernandez et al., 2007; Schaeffer-Filho & Hutchison, 2014).

4.3.4 Usage requirements

Harrison & White (2012) proposed 5 requirements for attack information sharing: privacy preserving, trust, real time, useful and compatible. Among these requirements, privacy preserving, useful and compatible are modified into requirements on the ontology whereas trust and real time cannot be realised by the ontology. Trust need to be built within the information sharing community and be maintained between users (Harrison & White, 2012; Johnson *et al.,* 2014). The requirement on trust cannot be reflected in the user-system interaction use cases of the ontology. Building attack pattern require time on data collection, analyse and processing thus it is difficult to update information of attack pattern in real or near real time.

For the privacy preserving requirement, we adapt it for out ontology in **Requirement 3: The attack pattern ontology will not provide attribution in annotation or any other forms.**

This requirement also fills in the second gap of existing researches mentioned in 3.1.4 Attack pattern in theories: For a common language of sharing attack pattern, what should be changed to the existing attack pattern concepts. Some attack pattern templates point out previous attack incidents as examples of certain attack pattern, which will be prohibited in our ontology.

Generally, companies resist on providing attack information to government (consumer) that could reveal their vulnerabilities; government (regulator) could use such information to regulate against the companies (Gal-Or & Ghose, 2004). Even if the government (regulator) introduces security breach laws and corresponding sanctions to enforce companies to share information, the outcome may not be socially beneficial; we do not know yet how to set criteria to differentiate malicious concealment and benign nescience (Laube & Böhme, 2015). Hence the information producers preserve their privacy by requiring anonymous contribution (Johnson *et al.,* 2014). Actually, guarantee the information provider anonymity is the best way to protect the privacy of users (Harrison & White, 2012).

Corresponding to CQ1 and CQ3, we generate **Requirement 4: The attack pattern ontology should**

- **Have a clear and comprehendible skeleton structure**
- **Connect all related entities according to their relations**
- **Provide necessary contents for users to query**

This is corresponding to the useful requirement proposed by Harrison & White (2012). This requirement also deals with the disadvantages of CAPEC that discussed earlier in 3.1.3 Attack pattern in practice. Requirement 4 asks the ontology to be easily understandable that a main structure should be carefully developed; it also asks the ontology to be easy to use that it should contain necessary contents that are arranged in order. If our ontology satisfy this requirement, it will avoid the orderless and chaotic situation of CAPEC.

In order to enable users to gain better understanding about the threat environment, increase the situational awareness of the community, aggregate and update knowledge, make decisions with greater speed and confidence (Johnson *et al.,* 2014). The ontology should provide a low learning barrier with clear structure and definitions (Heflin, 2009). For an ontology, structure not only refers to a hierarchical structure that taxonomies have, but also the relations between entities. Clear relations can present clear definitions of entities. In addition to clear structure and definitions, the ontology should realise its advantage over other data models: reasoning. In other words, ontology return query results to users.

For the compatible requirement, we applied it for our ontology in **Requirement 5: The attack pattern ontology should be expressive.**

The information sharing system should be capable of accommodating a growing amount of participating community members (Harrison & White, 2012). This is a factor to be considered when designing the information sharing architecture (Johnson *et al.,* 2014). For the ontology, scalability refers to the ability of expressing a wide variety of knowledge (Heflin, 2009). When the amount of information shared grows, the ontology grows as well; to support a large ontology, we need to choose a proper language that scale well and at the same time is as expressive as possible.

4.4 Conclusion

This chapter explained how the ontology is going to be used and the supports and obstacles of sharing attack information in real life. It gives answer to the first sub-question:

> What are the requirements of sharing attack information that could influence the decision-making on classes, properties and instances of the ontology?

Gathered and modified for our attack pattern ontology, we produced 5 requirements from literature review and chapter 3. Background. The requirements and their development processes are showed in the table below:

Table 5 overview of requirement development process

Requirement	Development process
1 The attack pattern ontology should contain enough but not redundant concepts that can describe attack pattern	Scenario → CQ → Requirement
2 In the attack pattern ontology, one attack is captured in only one attack pattern.	Information sharing objectives → requirement
3 The attack pattern ontology will not provide attribution in annotation or any other forms.	Information sharing objectives → Requirement
4 The attack pattern ontology should o Have a clear and comprehendible skeleton structure o Connect all related entities according to their relations o Provide necessary contents for users to query	Scenario → CQ → Requirement
5 The attack pattern ontology should be expressive	Information sharing objectives → Requirement

5. Shareable attack pattern ontology

In the previous chapters, we introduced the existing knowledge about ontology and attack pattern and developed requirements for the attack pattern ontology. In this chapter, we will develop a shareable attack pattern ontology based on these requirements. The second sub-question will be answered:

SQ2 How to create attack pattern ontology in the domain of sharing attack pattern?

We follow the sequence of developing ontology, which is explained in 2.6 Research method. Firstly, we assign identities (class, property and relation, constraint, etc.) to the key concepts stated by requirement 1. A class hierarchy is introduced using the combination method of top-down and bottom-up methods and reuse some existing taxonomies. The process of building this hierarchy is showed in Appendix A. Then we define the 3 types of properties: object properties, data properties and annotation properties. The procedure is to find what properties exist and restrict properties with property restrictions. At last, we created four attack pattern instances to show how it works; the instances can be found in Appendix B. All of these are implemented in Protégé; the codes can be found in Appendix C.

5.1 Concept identity and structure

In this section, we build a class hierarchy and define which concepts are classes, which are properties. Explanation and interpretation about the relationship, constraint and characteristics of properties are showed in the next section.

5.1.1 Key concepts

From the definition of attack pattern (see 3.1.4 Attack pattern), a draft version of key concepts of the ontology is presented in the table below.

Table 6 Key concepts version 1: from definition

Concept	Relation with attack pattern
Attack	Attack pattern describes attacks that commonly occur in specific context
Vulnerability	Attack pattern describes critical features of the vulnerabilities exploited by attackers
Method	Attack pattern describes how a type of attack is executed
Attacker	An attack pattern is specified from the point of view of an attacker
Knowledge required	Attack pattern describes knowledge required for attackers to exploit the target system
Target system	
Counteract the development of the attack	Attack pattern presents a way to counteract the development of the attack

Source: adapted from Moore *et al.,* 2001; Hoglund & McGraw, 2004; Barnum & Sethi, 2007; Fernandez *et al.,* 2007; Gegick & Williams, 2007; Pauli & Engebretson, 2008; Uzunov & Fernandez, 2014

All the key concepts from attack pattern definition (Table 6) are included in the concepts from attack pattern attributes (Table 3). Their correspondence relation is showed in the first two columns of Table 7 and Table 8. Table 7 summarised concepts that are not keeping in the ontology and Table 8 shows the 2nd version of attack pattern key concepts, which includes both the concepts from attack pattern definitions and the concepts from attack pattern templates. The interpretations of the concepts are adapted from the same articles of the data source of Table 3; there is only one exception that Mitre Corporation (2014) does not define the attack pattern template items. The reason is that Mitre Corporation (2014) is version 2.6 of the attack pattern template, we found interpretations from a former version produced in 2008 (Barnum, 2008), and we use the definitions from that version.

Table 7 Concepts that will not be kept

Concept from definitions	Concept from attributes	Reason for not keeping the concept
Attack	Known uses	According to user requirements, users require anonymous contribution and attribution is not allowed
Attacker	Perpetrator (who)	Attack pattern captures attackers' view but not who the attacker is
N/A	Attack motivation	Overlapping with attack target and attack consequence
N/A	Pattern completeness	According to our ontology requirements, all attack patterns are supposed to be 'complete' for the gather statistical information use case
N/A	Pattern abstraction	The class hierarchy already presented it

- As discussed in the introduction chapter, sharing attack pattern does not disclose the vulnerabilities of single organisations. To avoid information consumer knowing the identity of information producer, single attack incidents or known uses are rejected.
- Attacker's identity or who penetrated the victims is not of interests for extracting the similarities and regularities from attacks; it might be of interest if people want to arrest the attackers or produce an attacker profile.
- Attack motivation overlaps with attack target and attack consequence. Attack consequence is caused by attack motivation; so attack consequence actually presents the successfully achieved attack motivation. There may be unachieved attack motivation, but we cannot know that from the defender's side. Attack motivation sometimes also overlapping with attack target, for example when an attack target is user, the motivation is often stealing personal information.
- Pattern completeness is a concept from CAPEC where one attack pattern can have only one item, summary; one attack pattern can also have over 20 items. This concept is an indication to the number of items an attack pattern has whereas all attack patterns based on our ontology have the same number of items.

- Similarly, pattern abstraction is also from CAPEC to show one attack pattern's position in the hierarchy; how far is the current node away from the top node. When using an ontology, users can always see an attack pattern's position as well as all the other branches that this attack pattern does not belongs to.

Table 8 Key concepts version 2: from definition and attributes

Concept from definitions	Concept from attributes	Interpretation
N/A	Name ID	Brief descriptive name of the pattern Unique integer identifier of the pattern
N/A	Attack prerequisites	The condition or characteristics of the target system much has or behaves in order for such type of attacks to happen
N/A	Resources required	Resources required by an attacker to execute this type of attack, such as CPU cycles, IP addresses, tools, etc.
Knowledge required	Attacker skill or Knowledge required	Level of skill or knowledge required
		Skill or knowledge required by an attacker to execute this type of attack
Vulnerability	Related vulnerabilities or weaknesses	The mistake or improperness that can be used to perform an attack
Method	Method of attack	The mechanism used by this type of attack
Target system	Attack target	The targeted information asset
Counteract the development of the attack	Solutions and Mitigations	Approaches that can prevent or mitigate the attack
Attack steps	Description/ Execution/ Process diagram	The steps for an attacker to execute the typical flow of the attack
N/A	Attack consequence	The technical result achieved by the attack
N/A	Related patterns	Other attack patterns relate to, dependent on, chained together, etc. with this pattern
N/A	Status	The progress, version of the pattern
N/A	N/A	Number of occurred attacks: The happened times of this type of attack within a fixed time period (per month, per year, etc.)
N/A	N/A	Typical severity: The severity of the impact to the target system if the attack occurs

Source: Adapted from Moore et al. (2001); Barnum & Sethi (2007); Fernandez et al. (2007); Gegick & Williams, 2007; Pauli & Engebretson, 2008; Barnum (2008); Blackwell (2012); Mitre Corporation (2014); Uzunov & Fernandez, 2014

After deleting the concepts in Table 7, the concepts left are showed in Table 8. In addition to these concepts from the attack pattern definition and attributes, we added 3 concepts to the key concept list (the last 3): number of incidents and typical severity. From the example scenarios, we know that users seek comparable information from the attack pattern ontology. Therefore we added these 3 concepts to help information consumers to sense the quality and quantity of attack pattern, at the same time persuade information producers to scale attack pattern:

- Number of incidents, the quantity of the attacks incidents that converted into the current attack pattern, which can make up the shortage of known uses
- Typical severity, the quality of the attack presented in the way that how much impact can be caused by this kind of attack

5.1.2 Class and property

Before producing a class hierarchy, we have to define top level classes. In this section, we decide the identity of the key concepts based on the requirements generated in the previous chapter.

In the ontology editor Protégé, the basic components are individuals, properties and classes (Horridge et al., 2011). Their meanings are given in Table 9 (Horridge et al., 2011); their relations are presented in Figure 9.

Table 9 Main components in protégé

Components in Protégé	Meaning
Individual	Instance; entities that contain the knowledge to be shared
Property	Binary relationship between individuals or between individual and value
Class	A group of entities that share specific characteristics

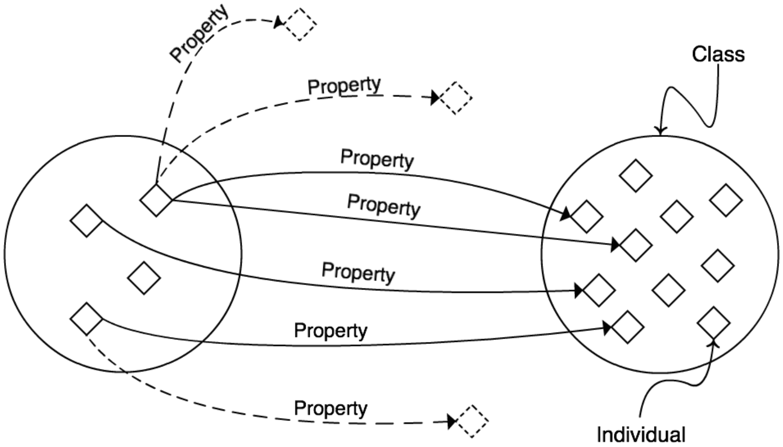


Figure 10 Main components in protégé

As showed in Figure 18, each kind of components has its own tab; in each tab, we can define and describe these components through the annotation view and the description view. In addition to individual, property and class, protégé also has ‘Active ontology’ and ‘Entities’ tabs to annotate the ontology and to show all components in the same tab.

Now, we assign the three identities in Table 9 to the key concepts based on the purpose of our ontology - share attack pattern. Table 10 gives an overview of all the concepts on the left and their identities on the right. To unify terminologies, we abbreviate the concept names into shorter terms. Two types of identities exist: class and property. A concept is a class if it is important for the described domain – it makes a distinction between objects or has particular implications to the relation between objects; a concept is a property if it only has marginal importance – it makes little or no difference for the representation of objects (Noy & McGuinness, 2001). Protégé allows three types of properties: object property, data property, annotation property (Horridge et al., 2011). Object property links an individual to another individual; data property links an individual to a data value; annotation property is annotation types such as ‘editor’, ‘version’ and ‘date’. Annotation property links not only an class but also object property, data property, individual and ontology to an annotation; one annotation can even be further annotated. Individual is not a choice of the identity type because we are defining key concepts here, which is *metadata*; individuals are the objects or the instances defined by the key concepts, which is *data*. Metadata is ‘data about data’; it can include the content, context and structure of the resource object (Kim *et al.*, 2005).

Table 10 The identity of key concepts

Key concepts	Terminology	Identity type			
		Class	Property		
			Object	Data	Annotation
Name	Name			v	
ID	ID			v	
Attack pattern	Attack pattern	v			
Attack prerequisites	Prerequisites	v			
Resources required	Resources required	v			
Attacker skill or knowledge required	Skill or knowledge required	v			
Skill or knowledge level	Skill or knowledge level	v			
Related vulnerabilities or weaknesses	Vulnerabilities	v			
Method of attack	Method	v			
Attack target	Target	v			
Counteract the attack	Countermeasures	v			
Description/ Execution flow/ Process diagram	Execution				v
Attack consequence	Consequence	v			
Related patterns	Related patterns		v		

Status	Status				v
Version	Version				v
Typical Severity	Typical Severity	v			
Number of occurred attacks	Number of occurred attacks			v	

Now we explain in detail about the choices in the table above. The classes are what we believe distinct one object from another. On the contrary, the properties are less important in making distinctions. Either a property is not comparable among objects or not interesting to be compared. For example, the *name* and *ID* are identifiers and thus unique and not comparable; the *description* is hard to be compared as well as not interesting to be compared because it can be seen as an interpretation of the *attack method*, *attack consequence*, *attacker skill or knowledge required*; *related patterns* is an relation and not comparable; *status* and *version* are annotations attached to the individual of little interest to be compared; *number of occurred attacks* is also comparable, but as a consequence of anonymous contribution, the context information cannot be provided for each of the occurred attacks, people will gain little value from comparing the occurred attack numbers by assuming similar attack context or impact.

5.1.3 Class hierarchy

In this section, we build a class hierarchy as a structure to breakdown the key concepts. We first use a bottom up method to connect the bottom level concepts with the high level concepts, then apply a top down method to complete the hierarchy and add siblings to the middle level and bottom level concepts. The significance of using a bottom up method is not only to assure the breadth of the structure, but also tailor the structure for the attack pattern instances in case the best structure is not a strict tree hierarchy. The bottom level concepts are distilled from 15 CAPEC (Common Attack Pattern Enumeration and Classification) attack pattern instances, which are from different attack domain and different mechanisms of attacks. CAPEC attack patterns are built by experts and ‘far more comprehensive than anything online’ (WASC, 2010), therefore we believe in the quality and the comprehensiveness of these attack pattern instances. The process of building this structure is showed in Appendix A.

In order to satisfy the first point of **Requirement 4** (a clear skeleton structure), we referenced some existing attack taxonomies (Simmons et al., 2014; Scarfone, Souppaya, Cody, & Orebaugh, 2008; Ye, Newman, & Farley, 2006; Hansman & Hunt, 2003). All of these taxonomies have detailed hierarchical structures for attack methods, many of them also have a list of attack impacts and attack targets. However we did not find classifications about attack prerequisites, resources required and attacker skill or knowledge required. Depends on the relevant taxonomies these articles provide, we reuse these taxonomies for these branches of our hierarchy as showed in Table 11:

Table 11 Reused taxonomies from articles

Source	Reuse taxonomy to add siblings for the branch(es)

Simmons et al., 2014	Vulnerability, Method, Consequence, Target
Scarfone et al. 2008	Vulnerability
Ye et al. 2006	Target
Hansman & Hunt, 2003	Method, Target

The criteria of reusing taxonomies are:

- High correlation with the middle level and low level concepts extracted from the 15 attack pattern instances (showed in Appendix A)
- Great hierarchical depth, which is necessary for adding siblings
- When two taxonomies from different sources are conflicting for the same branch
    - On the same hierarchical level, we take the union of the concepts instead of the intersection
    - When the same concept is put in different hierarchical levels, we follow the one where the concept is closer to the top level concept to simplify the hierarchy

With the help of existing attack taxonomies (as mentioned prior), we add siblings to complete the mid-level classes. The hierarchy is presented in both Table 12, the grey shading cells present the added siblings, and Figure 11. The '…' in Table 12 are where extra classes can be added.

Table 12 Class hierarchy

Top-level concepts	Middle level concepts	
Prerequisites	Target performs specific function	
	Existence of a specific target	
	Access to the target	Physical access
		Remote access
	No specific prerequisites	
	…	
Resource required	Material resource required	
	Financial resource required	
	Human resource required	
	Time resource required	
	No specific resource required	
	…	
Skill or Knowledge required	Skill of investigating system feature	
	Knowledge and skill of specific attack method	Knowledge of SQL
		Send HTTP requires, run the scan tool
		Social engineering technique
		…
	Knowledge and skill of specific software	
	Knowledge of specific hardware	
	No specific knowledge and skill required	

	…					
Target	Network	Application				
		Presentation layer				
		Session layer				
		Transport layer				
		Network layer				
		…				
	Software	Operating system	Windows		Name	Versio n
			Unix			
			MaxOS			
			…			
		Application	Server	Database		
				Email		
				Web		
			Client			
	Hardware	Computer				
		Network equipment				
		Peripheral devices				
	User					
Vulnerabilities	Kernel flaws	Unrestricted Consumption				
		…				
	Buffer overflow					
	Insufficient input validation	Injection	SQL, LDAP, Xpath query injection			
			Cross-site Scripting (XSS)			
			OS command injection			
			…			
		…				
	Insufficient authentication validation	Broken authentication				
		Cross site requires forgery				
		Unvalidated redirects and forwards				
		Missing Function Level Access Control				
		…				
	Misconfiguration	Default settings				
		Unused entities				
		Unprotected files and directories				
		…				
	Incorrect File and directory permissions					
	Social engineering					
	Weak physical protection					
	Symbolic links					
	File descriptor attacks					

	Race conditions		
Method	Denial of service	Network based	Flooding
		Host based	
		Distributed	
	Password attack	Guessing	Brute Force
			Dictionary attack
		Exploiting Implementation	
	Network attack	Web compromise	Database attack
			Cross site scripting
			Parameter tempering
			Cookie poisoning
			Hidden field manipulation
		Spoofing	
		Session Hijacking	
		Wireless attack	
	Physical attack		
	Misuse of resources	API Abuse	
		Protocol manipulation	
	Installed malware	Virus	
		Worms	
		Trojans	
		Spyware	
Countermeasure	Reduce the negative effect or probability of the attack		
	Avoid the attack		
Countermeasure	Design		
	Implementation		
	Configuration		
Consequence	Resource consumption		
	Gain privileges		
	Information disclosure		
	Modification		
Skill or Knowledge Level	High Skill or Knowledge Level		
	Medium Skill or Knowledge Level		
	Low Skill or Knowledge Level		
Typical Severity	High Typical Severity		
	Medium Typical Severity		
	Low Typical Severity		

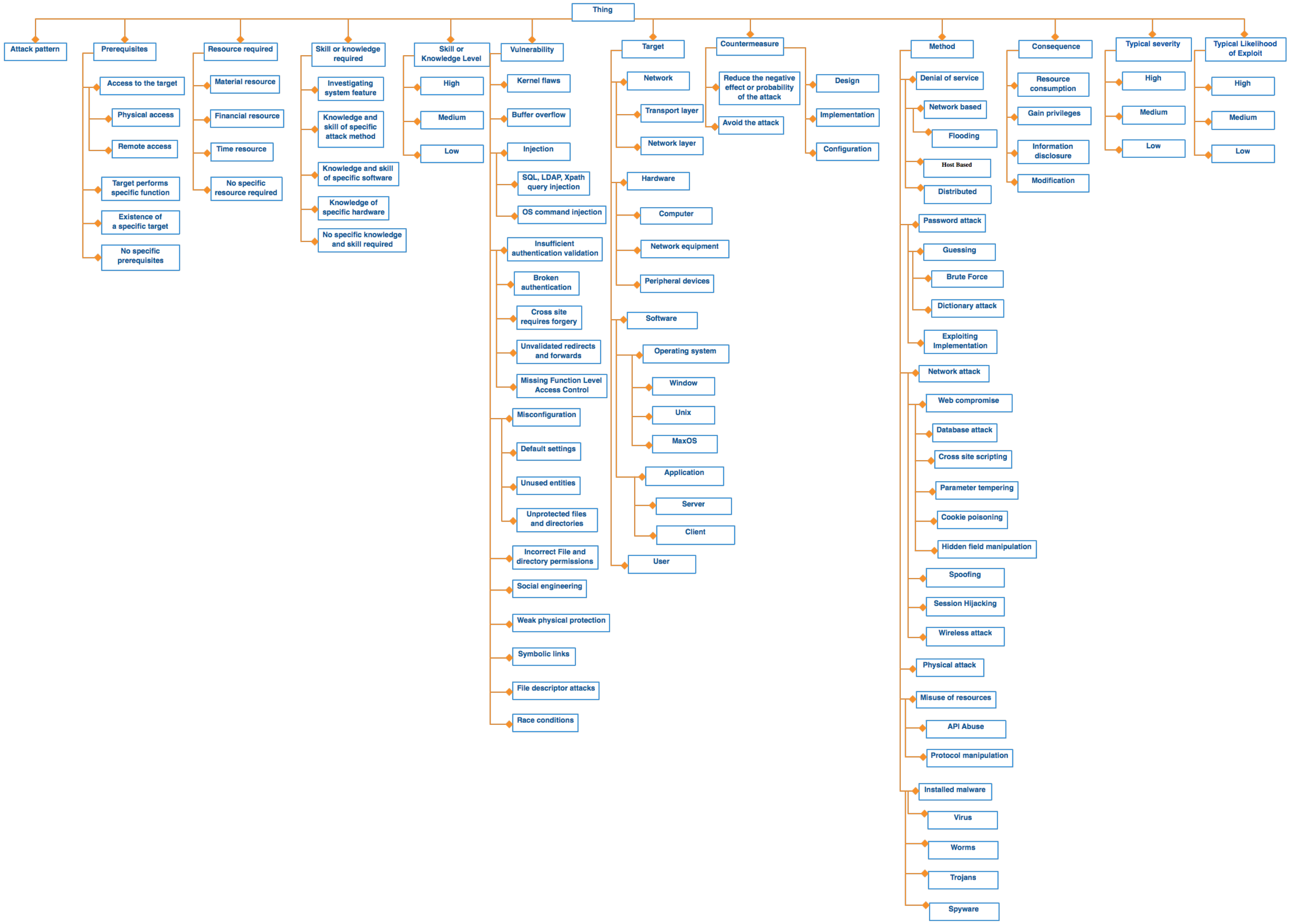


Figure 11 Class hierarchy

5.2 Concept relation

In this section, we are going to define how instances of each class relate to each other or relate to data values by defining properties and property restrictions. This task further defines the ontology that auto-classification and reasoning are done based on it.

5.2.1 Add properties

We cannot define property restriction without defining properties. So in line with the second point of **Requirement 4** (connect related entities), we need to find all the relations between the individuals of each class; the classes are linked with the relations of the individuals belonging to them (Horridge et al., 2011). We already know that all the individuals are related to attack pattern individuals; besides, we find these additional relations from literatures:

- Attack pattern encapsulates attack (Uzunov & Fernandez, 2014)
- Attack exploits vulnerabilities (Meier, 2003)
- Attack is made by employing action (Miede *et al.,* 2010)
- Countermeasures work against attacks and vulnerabilities (Miede *et al.,* 2010; Vorobiev & Bekmamedova, 2010)
- Assets have vulnerabilities (Miede *et al.,* 2010; Vorobiev & Bekmamedova, 2010). The relations can be described in a reverse way, one such example could be: vulnerability influences asset (Vorobiev & Bekmamedova, 2010)
- Attack affect an asset and result in a consequence (Vorobiev & Bekmamedova, 2010)

As a result of these additional relations, we add eight new relations: two between countermeasure and vulnerability, two between vulnerability and countermeasure, two between target and vulnerability, two between vulnerability and target. After adding these new properties, we got a preliminary version of our properties.

As showed in Table 13, we use the format hasRelation and isRelationOf to name the properties that relate instances in the domain class to instances in the range class. Property names are recommended to start with a lower case letter, have no spaces and prefixed with the word ‘has’ or ‘is’ (Horridge et al., 2011). Properties link individuals from the domain to individuals from the range. In Table 13, the first column is linked to the last column with the property in the middle column. Each object property may have a corresponding inverse property (Horridge et al., 2011). For example, hasTarget is the inverse property of isTargetof.

Table 13 Properties

Domain	Property name	Range
Attack pattern	hasRelatedPattern	Attack pattern
Attack pattern	hasTarget	Target
Target	isTargetOf	Attack pattern

Target	hasVulnerabilities	Vulnerabilities
Vulnerabilities	isVulnerabilitiesOf	Target
Attack pattern	exploit	Vulnerabilities
Vulnerabilities	isExploitedBy	Attack pattern
Attack pattern	hasPrerequisites	Prerequisites
Prerequisites	isPrerequisitesOf	Attack pattern
Attack pattern	hasResourceRequired	Resources required
Resources required	isResourceRequiredOf	Attack pattern
Attack pattern	hasSkillOrKnowledgeRequired	Skill or knowledge required
Skill or knowledge required	isSkillOrKnowledgeRequiredOf	Attack pattern
Attack pattern	hasSkillOrKnowledgeLevel	Skill or knowledge level
Skill or knowledge level	isSkillOrKnowledgeLevelOf	Attack pattern
Attack pattern	employ	Method
Method	isEmployedBy	Attack pattern
Attack pattern	hasConsequences	Consequence
Consequence	isConsequenceOf	Attack pattern
Attack pattern	isWorkedAgainst	Countermeasures
Countermeasures	workAgainst	Attack pattern
Vulnerability	isWorkedAgainst	Countermeasures
Countermeasures	workAgainst	Vulnerability
Attack pattern	hasTypicalSeverity	Typical Severity
Typical Severity	isTypicalSeverityOf	Attack pattern
Attack pattern	hasName	N/A
Attack pattern	hasID	N/A
Attack pattern	hadNumberOfOccurredAttacks	N/A
Attack pattern	Execution	N/A
Attack pattern	Status	N/A
Attack pattern	Version	N/A

5.2.2 Restrict properties

The third point of **Requirement 4** (support query) asks for adequate properties and property restrictions to support the activity of adding instances automatically. These properties and property restrictions define the relation between individuals strictly; they can narrow down the variability of the relationships. For example, a property can relate multiple individuals or values to one individual; we can set this property to be *functional* to restrict that through this property, at most one individual or value can relate to an individual. Note that property restrictions define relationships that satisfy some restrictions, when we apply such property restrictions to class description, we are actually describing an anonymous class this class belongs to. We try to add all the restrictions and relations to our ontology. Here are some common restrictions of properties (Noy & McGuinness, 2001; Horridge et al., 2011):

- The number of the values - cardinality restrictions

    - Min: the cardinality restrictions describe the class of individuals that have at least a specified number of relationships with other individuals or data values
    - Max
    - Exactly
- The relationships an individual participate in- quantifier restrictions
    - Only: the universal restriction, also known as ‘allValuesFrom’, means the set of individuals only (∀) has relationships with an individual of a specified class; it is important to point out that the universal restrictions do not guarantee the existence of a relationship for a given property
    - Some: the existential restriction, also known as ‘someValuesFrom’, means the set of individuals has at least one (∃) relationship with an individual of a specified class
- The relationship an individual link to another specific individual - hasValue restrictions (∋)
- Data property value type
    - *String* is the simplest value type, which stores a sequence of elements. It is used for slots such as name.
    - *Number* (Float, Integer, etc.) describes properties with numeric values.
    - *Boolean* properties are simple yes or no (‘true’ or ‘false’) flags.
    - *Enumerated* properties specify a list of values allowed.

Note that, all the properties link individuals to individuals, not class to class. But attaching properties to each single individual would be ineffective and not necessary. Many individuals can relate to a specific individual or a specific class of individuals through the same property. To specify how a property works for a group of individuals, we need the property restrictions. It might be necessary to further clarify the distinction between quantifier restrictions and hasValue restrictions, which is explained in Appendix F Property restriction comparison.

5.3 Instances of the ontology

In this section, we create instances to apply our ontology and illustrate the usages of the ontology. The data is from payment card data breach incidents of the retail industry. The ontology will be used in similar ways for other industries. We chose these incidents because they caught lots of attention and we can get access to the details of these incidents through case studies. But still, some technical details are not available even from these case studies. Note that the original information of cyber attacks is not accessible to external individuals or organisations; the cases we referred to are exception of this condition.

As stated by 4.3.4 Usage requirements, each attack pattern captures all the attack steps with single or multiple vulnerabilities exploited and single or multiple methods employed. This can be seen from the attack pattern instance below and in Appendix B Attack pattern instances. For example, in the instance below, attacker chosen social engineering to get the initial access to the victim system because they have found supportive information.

Table 14 Attack pattern instance 1 – retail industry

<table>
<tr><td colspan="4">Attack pattern: POS Intrusion (domain)</td></tr>
<tr><td colspan="4">Execution/description:
- Attacker(s) search how the victim company interact with its vendors
- An email containing malware was sent to a vendor to get the credentials to an online vendor portal.
- Get access to the victim's system via the vendor portal and further infiltrate the network
- Install malware on point of sale system. The malware gather payment card information as cards were swiped.
- Data was sent to a shared central repository; default user account name and password for an IT management software suite were used to log in to the shared drive
- Data was moved to drop locations on hacked servers via FTP and sold on black market</td></tr>
<tr><th>Property</th><th>Individual or data (range)</th><th>Class path (range)</th><th>Annotation</th></tr>
<tr><td>hasName</td><td>POS Intrusion</td><td>n/a</td><td></td></tr>
<tr><td>hasID</td><td>100</td><td>n/a</td><td></td></tr>
<tr><td>hasNumberOfOccurredAttacks</td><td>1</td><td>n/a</td><td></td></tr>
<tr><td>hasTarget</td><td>Point of sale machine</td><td>-Target
-Network
-Application
-Client</td><td></td></tr>
<tr><td>hasPrerequisites</td><td>Access to a vendor portal</td><td>-Prerequisites
-Access to the target
-Remote access</td><td></td></tr>
<tr><td>hasResourceRequired</td><td>No specific resource required</td><td>-Resource required
- No specific resource required</td><td></td></tr>
<tr><td rowspan="2">hasSkillOrKnowledgeRequired</td><td>Find vulnerability of the vendor portal</td><td>-Skill or knowledge required
-Skill of investigating system feature</td><td rowspan="2"></td></tr>
<tr><td>Create virus scanner undetectable malware</td><td>-Skill or knowledge required
- Knowledge and skill of specific software</td></tr>
<tr><td>hasSkillOrKnowledgeLevel</td><td>High</td><td>-Skill or knowledge level
-High</td><td></td></tr>
<tr><td>exploit</td><td>Email vendors</td><td>-Vulnerability
-Social engineering</td><td>Attackers may did a Google search that results in a great deal of information of how the victim interacts with vendors. A vendor portal and a list of</td></tr>
</table>

			relevant companies may be revealed. An email containing malware was sent to a vendor to get the credentials to an online vendor portal
	Vulnerability found by common network tools	- Vulnerability - Misconfiguration	Technical details unclear
employ	Software installed on POS system	- Method - Installed malware - Spyware	
isWorkedAgainst	Remove the malware from the network	- Countermeasure - Reduce the negative effect or probability of the attack	The victim was initially informed by external organisation about the suspicious activities
		-Countermeasure -Implementation	
hasConsequences	Disclose guest payment card data	-Consequence -Information disclosure	
	Disclose personal data	-Consequence -Information disclosure	Includes name, mailing address, phone number or email address
hasTypicalSeverity	High	High	

Source: adapted from Protecting Consumer Information (2014); Radichel (2014); Tipton & Choi (2014)

5.4 An informal representation of the attack pattern ontology

Based on the classes, properties and property restrictions, a basic structure of the attack pattern ontology has already been finished. However it is not a formal representation of the ontology because it is not written down in a formal ontology language in this section yet.

For individuals, we need to define their binary relationships through object properties and their relationships with values through data properties. For classes, we need to define the disjointness, because by default classes are overlapping and any individual may be an instance of any class (Horridge et al., 2011). For properties, we need to define property restrictions and property characteristics (see below). In addition to property restrictions, we can enrich the meaning of properties through the use of property characteristics (Horridge et al., 2011). We can make the object property functional (at most one individual relate to it), inverse functional (its inverse property is functional), transitive (e.g. hasAncestor), symmetric (e.g. hasSibling), asymmetric (e.g. hasChild), reflexive (relate an individual to itself), irreflexive (the domain and the range do not refer to the same entity). For data properties, the only characteristic is functional.

In the tables below, we define the characteristics and property restrictions of data properties, annotation properties and object properties:

- Property restriction
    - Cardinality restrictions
        - Min
        - Max
        - Exactly
    - Quantifier restrictions
        - Only
        - Some
    - hasValue restrictions
- Property characteristics
    - Functional
    - Inverse functional
    - Transitive
    - Symmetric
    - Asymmetric
    - Reflexive
    - Irreflexive

Table 15 Data property characteristic

Domain	Data properties	Range	Characteristic
Attack pattern	hasName	String	Functional
Attack pattern	hasID	Integer	Functional
Attack pattern	hasNumberOfOccurredAttacks	Integer	Functional

As already stated, a property links the individual in the domain class to the individual or value in the range. The attack pattern name and ID are unique for each individual; we will add them for each single attack pattern. Thus property restriction is not needed. All of these three data properties relate only one value to each attack pattern, so they are functional.

Table 16 Annotation property restriction and characteristic

Domain	Annotation properties	Range
Attack pattern	Execution flow	Literal
Attack pattern	Status	String
Attack pattern	Version	String

Users can use annotation to further describe and add information to the class, property or individual. In general, the range of status and version is string (Horridge et al., 2011). The execution flow is a detailed description, therefore literal is assigned to the range. Users can add annotation or create annotation types easily. For example, a label, a comment, etc.

Table 17 Object property restriction and characteristic

Domain	Property restriction	Property name	Range	Characteristic
Attack pattern	Only	hasRelatedPattern	Attack pattern	Symmetric; Irreflexive
Attack pattern	Only Some	hasTarget	Target	Functional; Asymmetric; Irreflexive
Target	Some	isTargetOf	Attack pattern	Inverse functional; Asymmetric
Target	Only	hasVulnerabilities	Vulnerability	Asymmetric; Irreflexive
Vulnerability	Only	isVulnerabilitiesOf	Target	
Attack pattern	Some	hasPrerequisites	Prerequisite	
Prerequisite	Some	isPrerequisitesOf	Attack pattern	
Attack pattern	Only Some	hasResourceRequired	Resources required	
Resources required	Some	isResourceRequiredOf	Attack pattern	
Attack pattern	Only Some	hasSkillOrKnowledgeRequired	Skill or knowledge required	
Skill or knowledge required	Some	isSkillOrKnowledgeRequiredOf	Attack pattern	
Attack pattern	Only Some	hasSkillOrKnowledgeLevel	Skill or knowledge level	
Skill or knowledge level	Some	isSkillOrKnowledgeLevelOf	Attack pattern	
Attack pattern	Only Some	exploit	Vulnerability	
Vulnerability	Some	isExploitedBy	Attack pattern	
Attack pattern	Only Some	employ	Method	
Method	Some	isEmployedBy	Attack pattern	
Attack pattern	Only Some	hasConsequences	Consequence	
Consequence	Some	isConsequenceOf	Attack pattern	
Attack pattern	Only Some	isWorkedAgainst	Countermeasure	
Countermeasure	Some	workAgainst	Attack pattern	
Vulnerability	Some	isWorkedAgainst	Countermeasure	
Countermeasure	Some	workAgainst	Vulnerability	
Attack pattern	Only Some	hasTypicalSeverity	TypicalSeverity	
Typical Severity	Some	isTypicalSeverityOf	Attack Pattern	

For those inverse properties, we do not restrict ‘only’ to them because we need the ontology to be extendible to also contain attack incidents. All the properties with ‘attack pattern’ in the domain have ‘some’ restriction, because all the attack pattern individuals must relate to the individuals in the range. For instance, all attack pattern individuals must and only have target from the target class: Attack pattern ∃ ∀ hasTarget Target

To summarise the contents in this section, a high level overview of the ontology is given in Figure 12. As can be seen from the figure, it is an attack pattern centred ontology – all concepts serve for representing attack pattern.

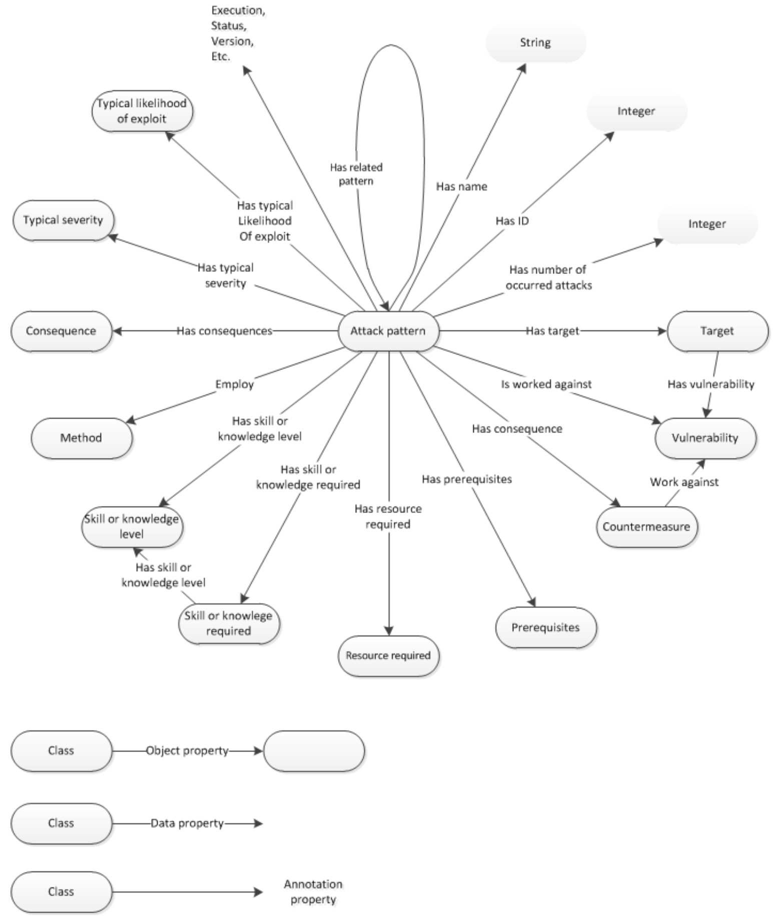


Figure 12 Attack pattern ontology

5.5 Choose a representation language

Requirement 5 asks for an expressive ontology language. We use the ontology editor Protégé to implement the ontology introduced in this chapter. Protégé helps to encode the ontology into a formal language. The code is pasted in Appendix C.

An ontology defines a common and machine-interpretable vocabulary that allows the reuse and sharing of knowledge (Gómez-Pérez, 2004; Noy & McGuinness, 2001). At the beginning of the

1990s, ontologies were mainly built using artificial intelligence (AI) modelling techniques based on frames and first order logic (Gómez-Pérez, 2004). Many AI-based ontology implementation languages have been created; nowadays people exploit the characteristics of the Web into ontology languages and produce web-based ontology languages: SHOE, XOL, OIL and DAML+OIL (Gómez-Pérez, 2004). The World Wide Web Consortium (W3C) is developing OWL as a standard and recommended ontology language, which is based on DAML+OIL (Gómez-Pérez, 2004). The Defence Advanced Research Projects Agency (DARPA) is developing DARPA Agent Markup Language (DAML) in conjunction with W3C to facilitate agent interaction on the Web (Hendler & McGuinness 2000).

In this research, we choose OWL (web ontology language) as the formal language to encode the ontology for the following reasons:

- OWL language is designed for the need of information processing that it facilitates greater machine interpretability than other languages such as XML and RDF (Heflin, 2009).
- OWL language has been widespread and used for the vast majority of ontologies (Vrandečić, 2009).
- OWL is a formal syntax for defining ontologies (‘Tutorial 3’, n.d.), which is recommended and fully supported by Protégé. Besides, the Protégé using guide is written for leading users to build OWL ontologies.

5.6 Conclusion

This chapter focuses on designing the ontology. Based on the analysis and outputs of previous chapters, it gives answer to the second sub-question:

SQ2 How to create attack pattern ontology in the domain of sharing attack pattern?

To answer this question, we defined the following four types of entities:

- Class in Table 10 and Table 12
- Property
  - Object property inTable 13 and Table 17
  - Data property in Table 13 and Table 15
  - Annotation property in Table 13 and Table 16

Attack pattern instances can be found in Appendix B and a formal representation of the ontology written in standard ontology language can be found in Appendix C.

6. Evaluation

In the previous chapter we developed the sharable attack pattern ontology based on the requirements generated earlier. In this chapter, the attack pattern ontology will be evaluated. The third sub-question will be answered:

SQ3: How effective is attack information sharing using attack pattern ontology?

We choose the evaluation criteria by literature review; eight criteria are to be evaluated. Among these eight, six are evaluated qualitatively through questions in questionnaire, one is checked by Protégé, the one left is not closely related to the design of ontology. The questionnaire can be found in Appendix D Ontology evaluation questionnaire. The results is in Appendix E Ontology evaluation result. Additionally, we also have quantitative evaluation where 3 criteria are discussed.

6.1 Evaluation criteria and methods

We evaluate the attack pattern ontology based on the criteria from the work of Vrandečić (2009). Vrandečić (2009) integrated and summarised ontology evaluation criteria from five previous articles (Gómez-Pérez, 2004; Gruber, 1995; Grüninger & Fox, 1995; Gangemi, Catenacci, Ciaramita, & Lehmann, 2005; Obrst, Ceusters, Mani, Ray, & Smith, 2007). As showed in Table 18, Vrandečić (2009) generate a concise set of 8 criteria: accuracy, adaptability, clarity, completeness, computational efficiency, conciseness, consistency, and organisational fitness. These criteria measure different aspects of ontology including scope, structure, correctness, etc.

Table 18 Ontology evaluation criteria adapted from Vrandečić (2009)

Criteria	Interpretation
Accuracy	The definitions and descriptions of classes, properties and individuals are compatible with the conceptualisations of the users.
Adaptability	The ontology offers the conceptual foundation for anticipated tasks and unexpected situations, i.e. include required meta-data.
Clarity	The ontology effectively express the intended meaning of the defined terms such as objective definitions and understandable names of elements
Completeness	The ontology covers the domain of interest; it gives to all questions that supposed to be answered by the ontology
Computational efficiency	The ability of reasoners, particularly its speed, to perform required tasks including query answering, classification and consistency checking
Conciseness	The ontology does not include irrelevant elements or redundant representations
Consistency	The ontology does not include contradictions
Organisational fitness	Decided by an organisation if an ontology should be applied or not through ontology metadata: tools, libraries, data sources, etc.

The method of evaluating each of the 8 criteria of the ontology is given in Table 19. One exception is the consistency criterion; consistency checks the logic of the ontology that can be done by running the reasoner of Protégé. A reasoner provides two main services including classification and consistency checking (Horridge et al., 2011). If any contradiction or inconsistency is reported in the error report after running reasoner, we will fix the inconsistency until no error is reported. In the remaining 7 criteria, computational efficiency asks more on the reasoner ability than on the ontology design. We will only analyse the impact of our design on this criterion through quantitative evaluation, but not the reasoner ability. Except the 2 criteria discussed above, 6 criteria are evaluated by questionnaire in order to obtain objectiveness. In addition to the qualitative evaluation, in the last column of Table 19, 3 criteria (adaptability, completeness and computational efficiency) can be evaluated quantitatively through structural metrics (Vrandečić, 2009).

Table 19 Evaluation methods for each criterion

Criteria	Method	
	Qualitative	Quantitative
Accuracy	Questionnaire	
Adaptability	Questionnaire	Structural metrics
Clarity	Questionnaire	
Completeness	Questionnaire	Structural metrics
Computational efficiency		Structural metrics
Conciseness	Questionnaire	
Consistency	Run reasoner in Protégé	
Organisational fitness	Questionnaire	

6.2 Qualitative evaluation

For qualitative evaluation, the main method is questionnaire where participants are asked to answer questions and give feedback to the design and usage of the attack pattern ontology. Taylor-Powell (1998) gave some suggestions on writing questions in a questionnaire, such as ‘be specific’, ‘use clear wording’, ‘avoid questions that are too demanding and time consuming’, ‘use mutually exclusive categories’ and ‘avoid making assumptions’. For evaluating our ontology, the hardest part is to develop questions that are neither demanding nor time consuming as there are 6 criteria to be evaluated. In order to control the time spent on answering questions, we will not ask participants to evaluated a list of of all elements and descriptions of the ontology, instead, we will ask them to give feedbacks and suggestions for each criterion. So firstly, we ask closed questions to participants that if they think the ontology are clear and easy to use. If the answer is totally positive, the evaluation for this criterion is done and the participant can answer the question for the next criterion. However, given not positive answers, the participant needs to answer the followed open question for the same criterion; for instance, why this ontology is not good enough, where needs to be improved and how can it improved.

Master students of Delft University of Technology will be invited to participate in the qualitative evaluation process. For the participants, the process of evaluation follows 3 steps:

- (Approximate 10 minutes) Listen to the introduction given by the organiser (the designer of the ontology) and ask questions to try to understand the contents of the attack pattern ontology
- (Approximate 20 minutes) Finish tasks in the questionnaire by using Protégé (with the assistance of the organiser) and try to understand the usages of the attack pattern ontology
- (Approximate 20 minutes) Answer questions in the questionnaire

6.2.1 Use cases

In this section, we describe use cases that participants will need as the use guide of the ontology and Protégé. The 'right' results of performing these use cases are given in text as well as in screenshots of the user interface of Protégé. Participants are supposed to achieve the same results after following the use cases.

Developing an ontology is often not a goal in itself; other applications need to use the set of data and their structure defined by the ontology (Noy & McGuinness, 2001). We have specified the expected attack pattern ontology in 4. System specification. In order to confirm that our attack pattern ontology can reach our expectation, we designed a series of tasks. To assist participants in finishing these tasks, descriptions and instructions are provided in the form of use cases. Use cases are a way to express how a set of actors uses a system to achieve various goals (Bittner, 2002). Use cases not only show the process of activity, but also the conditions before and after the activity. Therefore participants can follow it to finish the tasks and check their results.

Two use cases were produced for users to test the basic activities of the ontology: produce information and consume information. As can be seen from Figure 9 Attack pattern information sharing overview, there are two types of information flow that indicates the activity of producing information or consuming information. Therefore users will evaluate the ontology based on these two basic activities.

Users know what are to be evaluated and what should they pay attention to before trying the ontology based on use cases. All the criteria to be tested are given in advance to employing the use cases. Users gain the deepest impression of the ontology by using it instead of observing it; they can feel the difficulties of converting text of description into required items of attack pattern, making choices among classes and understanding the relations between entities.

The data used for the two use cases is well controlled to avoid bias in the outcome. For the first use case – produce information, each user uses one attack incident description to build one attack pattern. They build one attack pattern only based on one attack incident to get an impression of the ontology; in reality users are supposed to use multiple attack incidents to build one attack pattern but it would have spend the partcipants too much time. These attack incident descriptions are the ones we used for building the four attack pattern instances (Appendix B Attack pattern instances), which are different from the ones we used for building the ontology class hierarchy

(Appendix A Build the class hierarchy). Therefore we ensure that the data used for building the ontology is not used for evaluation again. In addition, users will be asked to choose one from the four attack incidents. So the ontology can be tested with different attack incidents. The attack pattern corresponding to the chosen attack incident will be deleted from the ontology database. After deploying the first use case, there will be still four different attack pattern incidents. For the consume information use case, users will query from all the existing four attack pattern instances.

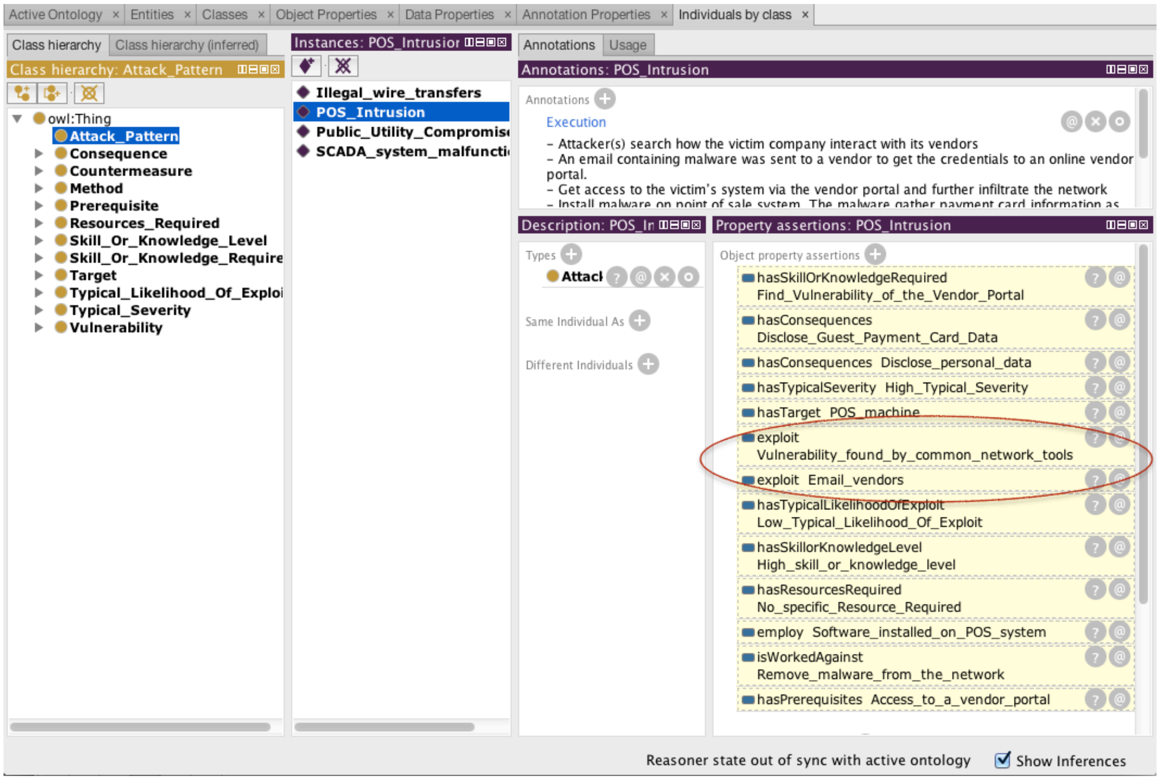
Each of the use cases (Table 20 and Table 21) below contains the following components: name, summary, actor, precondition, basic flows, alternate flows, precondition and post condition (Bittner, 2002). Actor is the role of the person who interacts with the system. Basic flows are the steps that actors take to achieve the goal of the use case; alternate flows capture the less common conditions when actors interact with the system. Precondition is the things that should be fulfilled when the use case begins; post condition is the things that must be true when the use case is complete.

Table 20 Use case 1 Produce information

Name	Create a new attack pattern instance
Summary	Add new attack pattern instance and its name, ID, corresponding consequence, countermeasure, methods, vulnerabilities, etc.
Actor	A user as an information producer
Precondition	This user has accessed into the system
Basic flow	1. Decide the class paths and annotations of each of the aspects of this attack pattern: consequence, countermeasure, method, etc. 2. In the individual tab, go to the instances window, add one instance and name it; in the description window, define its class path ‘Attack pattern’ 3. Select the added attack pattern instance, in the annotations window add its annotations such as execution flow. 4. Select the added attack pattern instance and go to the property assertions window, in the data property assertions section define the value that link to this attack pattern through data properties ‘hasID’ ‘hasName’ ‘hasNumberOfOccurredAttacks’ 5. Go to the class hierarchy window and select consequence, in the instances window, check if there are instances match the consequence of the just added attack pattern: if not, add a new one and define its class path in the description window (same as step 1); or else go to the property assertions window, in the object property assertions section relate the consequence instance(s) with the attack pattern through object properties 6. Repeat step 5 for all other instances that need to be connect with the new attack pattern through object properties 7. Go the menu reasoner → run reasoner, the instances added in step 5 should relate to the new attack pattern through the inverse object properties of the used object properties in step 5
Alternate flows	5. Users do not need to relate these instances with the attack pattern ‘public

	utility compromised', instead they can select the attack pattern 'public utility compromised' and relate it with all other relevant instances through object properties
Post condition	An new attack pattern instance is created and defined with relevant annotations, data properties and object properties

Figure 13 shows user interface of protégé after step 6 in use case 1. The area with yellow shadow shows the relations added by the reasoner. For example, we have defined that *exploit* is the inverse property of *isExploitedBy*; *isExploitedBy* is thus the inverse property of *exploit*. In step 5 of use case 1, we relate Email_vendors to attack pattern 'POS Intrusion' through *isExpoitedBy*. The reasoner can understand these relationships and relate 'POS Intrusion' to Email_vendors through *employ* (as mentioned above, this is one of the two services provided by reasoned: classification).



Execution

- Attacker(s) search how the victim company interact with its vendors
- An email containing malware was sent to a vendor to get the credentials to an online vendor portal.
- Get access to the victim's system via the vendor portal and further infiltrate the network
- Install malware on point of sale system. The malware gather payment card information as

hasSkillOrKnowledgeRequired Find_Vulnerability_of_the_Vendor_Portal
hasConsequences Disclose_Guest_Payment_Card_Data
hasConsequences Disclose_personal_data
hasTypicalSeverity High_Typical_Severity
hasTarget POS_machine
exploit Vulnerability_found_by_common_network_tools
exploit Email_vendors
hasTypicalLikelihoodOfExploit Low_Typical_Likelihood_Of_Exploit
hasSkillorKnowledgeLevel High_skill_or_knowledge_level
hasResourcesRequired No_specific_Resource_Required
employ Software_installed_on_POS_system
isWorkedAgainst Remove_malware_from_the_network
hasPrerequisites Access_to_a_vendor_portal

Reasoner state out of sync with active ontology

☑ Show Inferences

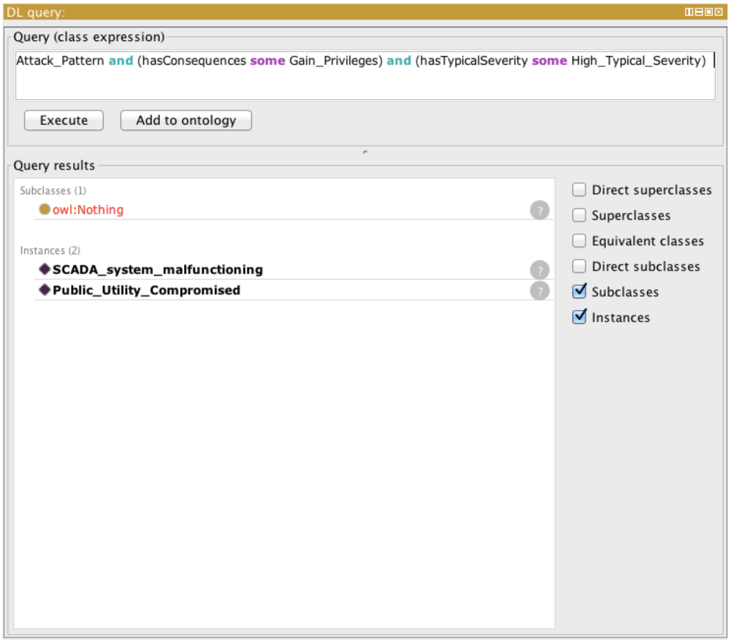
Figure 13 Step 7 of use case 1

After successfully adding an attack pattern, we can try reading the information from different classes. In Figure 13, from the instances window in the middle, we know that the selected instance is *POS_Intrusion;* in the property assertion window on the right, we can see that the selected instance is related with vulnerability instances, attacker skill and knowledge instance, attack consequence instance, etc. If users want to learn more about these aspects of the *POS_Intrusion* attack pattern, they can click on the name of the instance. For example, click on the vulnerability instance in the red cycle *Email_vendors,* the 3 windows on the right will show the annotation, description and property assertion of *Email_vendors.*

Table 21 Use case 2 Consume information

Name	Use reasoner to automatically compute entities that have consequence of gaining privileges and high typical severity
Summary	A user asks the ontology to do query that filter attack patterns with specific consequence and typical severity
Actor	A user as an information consumer
Precondition	This user has accessed into the system
Basic flow	1. Go to the Query tab 2. Entre the class expression in the query box: Attack_Pattern and (hasConsequences some Gain_Priviledges) and (hasTypicalSeverity some High_Typical_Severity) 3. On the right of the query results box, select the type of entities to filter 4. Synchronise reasoner and execute the query 5. The query results is showed in the query results box
Alternate flows	5. If no results is returned, check the query expression in step 2 and perform step 2, 3, 4, 5 again
Post condition	The information of interests is found

In Protégé, users have two ways to do query. Users can use SPARQL language to ask computer to finish various types of tasks including query certain results, check the existence of data, extract data and construct RDF figure. The returned query result will also in the form of SPARQL. Another way of query has less functions than using the SPARQL language, but it can avoid to learn a new language, in the use case users will use the same method of defining entities in Protégé to query with basic restriction words ‘some’, ‘exactly’, ‘min’, ‘and’, ‘or’ to connect object properties, classes or values. The second way of query result is showed in Figure 14.



```
Attack_Pattern and (hasConsequences some Gain_Privileges) and (hasTypicalSeverity some High_Typical_Severity)
```

Subclasses (1)
owl:Nothing

Instances (2)
SCADA_system_malfunctioning
Public_Utility_Compromised

☐ Direct superclasses
☐ Superclasses
☐ Equivalent classes
☐ Direct subclasses
☑ Subclasses
☑ Instances

Figure 14 Step 5 of Use case 3

6.2.2 Questions for qualitative evaluation

After trying the ontology in Protégé to finish the 3 uses cases, users will gain an initial impression about the ontology. Now, they can be guided by the questions in the questionnaire to criticize on the ontology. The complete questionnaire is showed in Appendix D Ontology evaluation questionnaire.

According to Table 19 Evaluation methods for each criterion, there are 6 criteria to be evaluated in this questionnaire: accuracy, adaptability, clarity, completeness, conciseness and organisational fitness. As discussed earlier, for each of the criterion, we will ask participants to point out the improvement space of the ontology:

1. Accuracy: Look into the names of the entities (classes, instances, properties, annotations) and the relationships between entities (instance – object property - instance), can you find the right entities and corresponding relationships for performing your task? If not, which names or description of relationships caused you difficulties?
2. Adaptability: Could you find all the necessary entities (classes, instances, properties, annotations) to fulfil the tasks?
3. Clarity: Can you easily understand the entities' names (classes, instances, properties, annotations) and descriptions (please go to the description window of classes, properties and instances)? If not, which names or descriptions could be improved?
4. Completeness: Does the ontology give answers to these questions? Which not and why?
    4.1: Can you convert attack steps and strategies into multiple attack methods or vulnerabilities? If not, why?
    4.2: Can you see the complete class tree?

4.3: Do you know what should be filled for an attack pattern by reading the class hierarchy? If not, what are not clarified?

4.4: Can you view information per attack pattern, per method, per prerequisites or per vulnerability?

4.5: When viewing an instance, can you see its position in the class hierarchy? When viewing a class, can you see its position in the class hierarchy?

4.6: Can you find information through query?

4.7: Can you reuse existing instances, i.e. can two attack patterns relate to the same method instance, prerequisite instance, etc.?

5. Conciseness: Do you think all the existing elements (classes, instances, properties, annotations) of the ontology are necessary? If not, which elements are redundant?

6. Organisational fitness: Imagine you are from the financial/ healthcare/ energy/ retail/ telecommunication company, if your company want to use this ontology as the base ontology of attack information sharing, what should be added, changed or deleted?

For the 4th criterion - completeness, we should generate questions that the ontology should give answers to, which are the competency questions mentioned earlier in Chapter 4. The questions here have been adapted for the realistic operations of the user thus they are not exactly the same with questions in 4.3.2 Competency question. In addition, we added some questions (4.1, 4.5, 4.6, 4.7) that are corresponding to the advantages of ontological model over other data models. These seven questions present seven expected advantages of this ontology, which will be discussed later in 6.4 Discussion.

6.2.3 Results

Five master students of Delft University of Technology gave answers to the questionnaire. All of them are second year students from the faculty of Technology, Policy and Management, i.e. with little or no cyber security expertise. Their answers for each of the 6 criteria are showed in Appendix E Ontology evaluation result. Generally, two subjective reasons caused obstacles to answer these 6 questions: short of cyber security knowledge and lack of long-term experience on using this ontology.

Use cases

According to observation and communication with the questionnaire participants, the ontology is difficult to understand but easy to use.

- After an initial introduction, the participants can understand easily what is the attack pattern for, what is the relationship between an attack and an attack pattern, what does the key concepts of the attack pattern mean
- It is easy to recognise the required elements from the attack descriptions, the attack pattern can be built very fast. Participants make choices quickly, such as which attack method the attacker used and what consequences the attack caused to the victim.
- For some glossaries of the cyber security domain, participants have difficulties to understand them. However based on the context descriptions, they can still recognise the required elements of attack pattern.

- If any suggestions would be made on the redundancy or shortage of the attack pattern attributes, a long-term experience (few months) on using the attack pattern structure is necessary

Answers to questions

For criteria 1 – accuracy, only one participant gives negative answers. All the five participants give positive answers for criteria 2 – adaptability whereas one participant provides suggestions of improvement. All the participants showed satisfaction on criterion 4 – completeness. Three out of five participants give positive answers to criteria 5 – conciseness, while the other 2 keep neutral as they are short of cyber security knowledge. The suggestions mainly concentrate on criteria 3 - clarity and criteria 6 - organizational fitness; almost all of the participants point out that

- The names of the entities (all or some) are hard to understand
- The ontology might need to be adjusted based on the condition of each industries

The suggestions for criterion 1 – accuracy and criterion 2 – adaptability are:

- Some entities under the class of vulnerability and under the class of network are unclear that the users do not know what are those entities
- Classes with levels (high, medium, low) do not give benchmark to users that which levels should be chosen
- Attack consequence could be expended and be more detailed

The feedbacks on criteria 3 - clarity clustering on the vulnerability class, where many glossaries arose that the participants have no idea about their meanings. There is no solution to it by changing the names, instead we can add detailed explanation and examples in the annotation. Add more subclasses and individuals can also help users to clarify the entity.

In the feedback on criteria 6- organizational fitness, participants agree that when applying the ontology in different industries, some adaptions are needed. However, they cannot give specific adjustment to be made. Such adjustment cannot be known without long-term experience on using this ontology, which is what the participants are lack of. It can be one of the topic for future researches.

6.3 Quantitative evaluation

Gangemi et al. (2005) and Tartir et al. (2010) provide five metrics to measure the design of ontology: depth, breath, relationship richness, inheritance richness and attribute richness. Equations of these five metrics as well as the evaluation results are given in Table 22.

- Average depth: average number of nodes for all paths; a path starts from a root node and ends at a leaf node
- Average breath: average number of nodes for all levels; a level is where a node and its siblings have the same distance from the root node(s)

- Relationship richness: the number of non-inheritance relationships divided by the total number of relationships
- Inheritance richness: average number of subclasses per root class
- Attribute richness: average number of properties per root class

Table 22 Quantitative evaluation

Evaluation Parameter	A	Value A	B	Value B	Formula	Value
Average Depth (AD)	Cardinality of paths	225	Number of paths	85	A/B	2.65
Average Breadth (AB)	Cardinality of levels	114	Number of levels	4	A/B	28.50
Relationship richness (RR)	Non-inheritance relationships	28	Inheritance relationships	102	A/(A+B)	0.27
Inheritance richness (IR)	Inheritance relationships	102	Number of root nodes	12	A/B	8.50
Attribute richness (AR)	Total number of properties	31	Inheritance relationships	102	A/B	0.30

The attack pattern ontology is developed with the aim to represent shareable cyber attack information at generic level. Various ontological metrics assessed in the quantitative evaluation process back up this claim. The results of the formulas of AB, AD and IR are real numbers representing the width, depth and average number of subclasses per class. The AB value indicates that our ontology contains wide (breadth) variety of concepts for describing attack pattern; however according to the AD value these concepts are not expanded in very much detail (depth). High IR implies the shallow (or horizontal) nature of the ontology that it represents a wide range of general knowledge with a low level of detail.

Different to the real number results of AB, AD and IR value, the other two parameter values needed to be compared with value 0 and 1. If a RR or AR value is close to 1, then the result value is high; otherwise if the value is close to 0, the result value is low (Tartir et al., 2005). Low value of RR indicates that classes are connected with each other only through fundamental relationships. Similarly, low AR indicates that classes contain more generic properties that belong to many different classes but less properties that belong to few specific classes.

We believe that such shallow but wide ontology is easy to be extended and adapted for both anticipated and unanticipated tasks without removal of existing entities and axioms. For the same reason, the ontology does not provide detail but it tends to bring completeness. In order to perform computational efficiency, the requirement on reasoner is low according to the low RR and high IR, which indicate the simple relationships between classes.

6.4 Discussion: ontology advantages, requirements and objective

In this section, we will explain the reason of asking the seven questions for criteria 4 – completeness. Each questions is corresponding to one advantage of using the ontology. We will also discuss how this ontology satisfy the five requirements and reach the research objective.

Advantages

Advantage 1. Shareable attack pattern ontology presents the multiple steps of a complete attack process. As highlighted in the circled area of Figure 13, one attack pattern can exploit more than one vulnerability, attack method, attack consequence, etc. Through these multiple vulnerabilities and methods, the complete attack process is presented. For example, one vulnerability (send a vendor email with malware) is exploited to get credentials to entre a vendor platform that relates with the victim; another vulnerability (explore misconfiguration mistakes) is exploited to move from the vendor platform to the targeted POS system. This way of presenting attack pattern is more valuable for users that they can see how attackers plan and perform a type of attack. After all, one feature of attack pattern is to capture the attacker's perspective (Barnum & Sethi, 2007; Fernandez et al., 2007; Uzunov & Fernandez, 2014). Only put these two vulnerabilities in the same attack pattern can users understand why attackers need to exploit the second vulnerability: they get access to a platform that relate to the targeted POS system but they are still out of the targeted system. The combination of the two vulnerabilities also reveals the reality that securing one organisation may be not enough to defend against cyber attacks; all the related external entities and organisations can be the start points. To prevent such attacks, both the retail company and its vendor companies have to be secure, which is more difficult than securing one single company. This in turn implicates the necessity of sharing attack information.

Advantage 2. A single hierarchy is used to contain and classify all elements of the shareable attack pattern ontology; users view all entities at one time. Compared with CAPEC (Common Attack Pattern Enumeration and Classification), this ontology is better in using only one hierarchy to classify all entities. The two taxonomies of CAPEC (by domain of attack or by mechanism of attack) exclude the CAPEC-286 and its sub attack patterns. In addition to that, CAPEC uses a top-down approach to define attack patterns (Schaeffer-Filho & Hutchison, 2014), which excludes the possibility of other types of structures in advance. This is exactly why CAPEC ends up with two none-comprehensive taxonomies. Instead, we build a single structure that all kinds of attack patterns can fit in. As showed in Figure 13, all the classes can be seen from the class hierarchy window; all the instances in the current selected class can be seen from the instances window.

Advantage 3. The top level classes unify the content of attack pattern; the lower level classes provides details and examples of the content to be filled. In CAPEC, attack pattern has description entities vary from 1 (only a summary) to over 20. As a result of a hierarchical structure, several classes are too broach such as 'abuse of functionality' while the next level classes would be too specific, for example 'WSDL scanning' that applies only to system based on web-services (Uzunov & Fernandez, 2014). On the contrary, the structure provided by our ontology not only provides comprehensive classifications but also tells users the entities need to be defined for each attack pattern: consequence, countermeasure, method, prerequisite, etc. Therefore users know what should be added (top level classes), what kind of instances exist for each of the entities to be added (lower level classes). This is clearer and more straightforward than CAPEC which does not have a uniformed template. Limited number of top level classes does not mean that the information contained by each attack pattern is also limited. As a consequence of using ontology, users can add as much information as they want by adding annotations and properties.

Advantage 4. In addition to view data by attack pattern, users can also view data by methods, prerequisites, vulnerabilities, etc. In CAPEC, attack pattern is recorded per attack pattern whereas with our ontology, attack pattern is recorded by multiple instances, multiple properties and multiple annotations. Therefore users have various ways of viewing recorded information. Basically, they can view attack pattern per attack pattern instance. They can also view one aspect of attack pattern, for example attack pattern that attackers install malware or attackers exploit physical vulnerabilities.

Advantage 5. Users can always see the instance's position in the hierarchy. As showed in Figure 13, we can see the position of the selected instance from the class hierarchy window. It is not achievable for CAPEC where users can only see the parent and children attack patterns in the adjacent level. In our ontology, as long as the class path has been defined, we do not need to do anything to get an overview of the positions of instances in the class hierarchy.

Advantage 6. Because each attack pattern is split into several related entities, users can reuse already exited entities when adding new attack pattern. In the four attack pattern instances we generated in Appendix B Attack pattern instances, two have the same resource required: 'no specific resource required'. When creating the second attack pattern, users do not need to add new instance in the resource required class, instead users just need to relate the attack pattern with 'no specific resource required' through the corresponding object property. This feature and advantage of our ontology avoids redundancy and squeeze the database size to the smallest. It also saves time of adding and editing information; users can view a database based on such ontology with shorter time.

Advantage 7. Entities can be defined in axioms or any logical expression, which is understandable for machine. Users can ask computer to filter data, to check the existence of data, to extract data or to construct RDF figure. Use case 3 is an example of filtering data, i.e. making a query. We applied three limitations that connected with 'and', which means we want to query the intersection of the 3 classes. The instances that match this expression is given in the query results box of Figure 14. This is an example of filter query, with SPARQL queries, the ontology is also capable of the following query actions: count, distinct, limit, union, joint, etc.

Requirements

The ontology was developed following the guide of the 5 requirements in chapter 4. We will test if the final product satisfy these requirements or not. The requirement sources and their constraints on the ontology are showed in Table 23.

Table 23 Requirements overview

Origin	Requirement		Ontology aspect
3. Background	1	Set the key concepts	Ontology contents
Discussion about existing studies (3. Background)	2	Set the relation between attack and attack pattern	Database contents based on the ontology / Ontology usage
Literature review about sharing attack information	3	No attribution	Ontology usage / contents
	4	Usefulness	Ontology usage
	5	Expressiveness	Ontology usage

To begin with, the ontology contents were constrained by requirement 1 and 3; requirement 1 defines the key concepts that must be presented in the ontology and requirement 3 defines the contents that must not be presented in the ontology. We built the ontology started from these two requirements and there is no doubt that they were satisfied.

Requirement 2 concentrates more on the database contents instead of the ontology contents; we can also cluster it as the ontology usage that it defines how attack pattern should be built from attack incidents. This is an advantage to be achieved if users build attack pattern following this requirement. We proved this advantage by asking question for the criteria 4 in questionnaire. We also showed how this requirement is satisfied in the attack pattern instances in Appendix B Attack pattern instances.

Requirement 3, 4 and 5 focus on ontology usage; they formulate the constraints of enabling a easy and useful information sharing process. We tried to satisfy requirement 3 – no attribution by deleting all the concepts that could disclose the identity of information producer: attack/ known uses. As for requirement 4, according to the questionnaire, users did not give feedback of not comprehendible structure and shortage of relations. Only the ‘provide necessary contents’ for query is debatable because users tried only one query. However, users need long term using experience to decide if more contents were needed for the query activity. Even if this requirement was not fully satisfied, we cannot tell from the questionnaire outcome. This can be a potential improvement space for further research. At last, we chose the OWL language to enable expressive of the ontology. Currently, this is the most popular and widespread ontology language that is accepted by most of the ontology editing tools. All the activities needed in the process of developing the ontology are fully supported by the ontology editor Protégé and are presented in the OWL code (Appendix C OWL code). Therefore requirement 5 is also satisfied.

Objective

We hope people with different background and positions can understand each other and share the same language through using our ontology. To prove that, we have to answer this question: will the outcome change if different people were invited to try and use the ontology? The outcome refers to attack pattern produced for the same attack incidents. Query is not considered as the problem of generalization here, because people will get the same query outcome as long as they know how to use the query language.

The problem here is about understanding and comprehension. The use cases and questionnaires proved that the participants could understand most of the ontology and have no problem of using it. Notice that the participants do not have much cyber security knowledge. Therefore people with equivalent or better education background should be capable of using the same ontology. People with experience or knowledge about cyber security should also have no problem of understanding or using it. However, for people who do not have enough knowledge or education background, there might be difficulties and obstacles. Therefore, the outcome will be the same if other well-educated people or people with expertise were invited for the questionnaire. But the outcome might be different for people with insufficient education or cyber security knowledge.

6.5 Conclusion

In this chapter, we evaluated our design of the attack pattern ontology. The last sub-question is answered:
SQ3: How effective does this ontology enable the attack information sharing activity?

From the qualitative evaluation, we know that our design is a shallow but wide ontology; this confirms the objective of our research that we want to build a base ontology that is generic so as to be used within different information sharing communities. From the quantitative evaluation, we received many feedbacks about the unclearness and difficulties in understanding the ontology contents. The reason is partly because of the short time given to questionnaire participants and their insufficient knowledge on the cyber security domain. However we are optimistic that after the following improvements, the problems on accuracy, adaptability and clarity will be significantly reduced:

- Adding annotations to define the classes and explain their usages
- Adding more subclasses, i.e. increase the depth of the hierarchy, to provide sense of what kind of events or objects belongs to one class

7. Discussion and conclusion

Through this report, we have seen definitions and descriptions about attack pattern and ontology; various requirements arose in developing the attack pattern ontology.

In this chapter, we return to the main research question and also look into the contribution of this research. Then we finish the report with discussion on the limitation of the research, recommendations for further research and a personal reflection on the thesis process.

7.1 Answering the main research question

Today's information technology is highly developed as a result of people's growing dependent on the internet. The impact that could be caused by cyber attacks is therefore increasing. It is not surprise to see a growing number of attacks being reported over recent years, yet to mention the attacks not being reported. Sharing attack information can enhance the security capabilities of the members of information sharing community. Our research objectives were to develop a common language that exploit a new usage of attack pattern and use it as the carrier of data in information sharing. In order to maintain a consistent understanding of the shared information, this objective was reached by developing a common language of the attack pattern concept through an ontological model. Therefore the following research questions is constructed:

> How to use attack pattern to present attack information in an ontological model for the purpose of unifying and formalising data exchanged and shared?

7.1.1 Defining the system of sharing attack pattern

We described the cyber security environment in the Netherlands where Internet is extensively being used and dependent on. We analysed the circumstance of sharing attack pattern from three perspectives. From the politics perspective, the Dutch government is very supportive in sharing cyber security information; from the value perspective, a process of sharing attack pattern is designed including how to form such cooperation and how to improve the design of the attack pattern ontology; from the technology perspective, the information flow in the system is described.

7.1.2 Developing an product for the system: attack pattern ontology

We looked at former studies on attack pattern to gain knowledge about what attack pattern is, what attack pattern delivers and how to build attack pattern. For the purpose of filling in the knowledge gap between existing studies and the needs on sharing information, we integrated requirements of cyber security information sharing into the requirements on ontology development.

Integrating the requirements into the steps of developing ontology, we defined key concepts and their roles in the ontology; we built class hierarchy as the main structure and define relations between elements; we also describe the attributes of the relations and produced four attack pattern instances; as last we presented the ontology with formal ontology language. This ontology is the base language for sharing information about attack pattern.

7.1.3 Assessment of proposed ontology

This design was evaluated by master students and based on their comments, more annotations and subclasses could be added so as to make the ontology easier to be understood and used. The evaluation results showed that this attack pattern ontology is easy to use but hard to understand. The reason is partly because of the short time given to questionnaire participants and their insufficient knowledge on the cyber security domain. Moreover, it is shallow but wide, which proved our will of using it within different information sharing communities. Two types of improvements can help users to easier and faster understand the ontology so as to use it:

- Adding annotations to define the classes and explain their usages;
- Adding more subclasses, i.e. increase the depth of the hierarchy, to provide sense of what kind of events or objects belongs to one class

7.1.4 Answering research question

For the purpose of using attack pattern as data carrier in an ontological model to seek consistency between information sharing community members, we developed a semantic web to contain attack pattern. Classes were used to cluster attack pattern data and relevant information; instances belong to each class were connected by properties; all the classes, properties and instances can be annotated by various annotation properties. The ontology has the following top level classes: Attack pattern, Prerequisites, Resources required, Skill or knowledge required, Skill or knowledge level, Vulnerabilities, Method, Target and Countermeasures. Each of these top level classes has some lower level branch classes. Attack data is recorded in instances and classified in classes. The ontology has the following data properties: name, ID and Number of occurred attacks. In addition to the default annotation properties such as comments, this ontology has the following special annotation properties that used to describe each attack pattern: execution flow, status and version. Each attack pattern is related with its prerequisites, resources required, etc. through object properties. Besides, each attack pattern can also relate to other attack pattern through an object property: related patterns.

Different to the existing researches, our attack pattern ontology emphasises the following features of attack pattern:

- Different to the normal way of pattern one attack step into one attack pattern, we pattern all the attack steps of complete attack incidents into one attack pattern.
- Information producer remains anonymous that the ontology does not provide any content relate to the information source in any form.

Different to the practice of the attack pattern concept in CAPEC, our attack pattern ontology has the following advantages:

- The ontology supports information consuming with a single clear structure and defined relations.

- Attack pattern contents are reusable that two different attack patterns can share one attack method, vulnerability, consequence, etc.
- Data reserved based on the ontology is 'smart' that auto-classification and auto consistency checking are possible.

7.2 Limitation

There are four limitations of the thesis on the methodology and three limitations on the deliverable usage. Firstly, the scope was limited to between organisations, which means information is shared between different organisations in the same country. This significantly decreased the complexity of the problem that we do not need to consider cross-border issues or departments' interdependencies. Because we are focusing more on the contents of the shared attack information, we believed that this narrow scope would not make much difference to the deliverable. However, other levels of information sharing are clearly relevant, with the actual needs of both the excluded situations. The limited scope for the thesis perhaps limits the generality of the delivered ontology. The potential influence of interactions between departments within the same organisation and international collaborations on the ontology design could be further explored.

Next, the lower level classes were developed based on existing attack pattern instances. Existing attack pattern concepts are different to the ones this research intended to capture: one step of one type of attack vs. all steps of one type of attack. Furthermore, some attack pattern instances may be captured from one or few industry sectors while we suppose the ontology is applicable in all industry sectors. These differences could have made our second, third and fourth level classes deviated from the ideal results.

Third, the use cases used for evaluation cannot cover all the aspects of the ontology but only show the basic usages. In evaluation, it is not possible to ask participants to spend more time on trying the ontology usages, for instance building more attack pattern instances using this ontology and doing more queries. Therefore the participants can only evaluate the ontology based on their limited experience on using this ontology. An ideal situation for evaluation is to invite more participants with different backgrounds and use this ontology for longer time.

Fourth, professionals were not invited in the qualitative evaluation as well. The identities of the questionnaire participants were all students, which has both advantages and disadvantages in it. Advantages are that participants can easily point out the parts that they do not understand. However, it is hard to judge the reason they do not understand is lack of certain background or not. If IT professionals and cyber security experts were also invited for the questionnaire, they might have provided some keen points on the organizational fitness criteria. However, the evaluation results and the experiences on use cases may not deviate much from the current results, because the difference between students and professionals is their experience in the cyber security area.

For the usage of the ontology, it is limited to the user group of well-educated people. The evaluation questionnaire participants could be master students and cyber security professionals. These participants are also the potentials users of the ontology; they can be clustered as well-educated people. Nevertheless, the ontology contains many glossaries of the cyber security domain, which may cause difficulties in using; compared with other people, well-educated people have better knowledge background in using it.

Then, the ontology may lead to confusion if users do not make agreement on the usage of it. Because of the generosity of the ontological model, all the classes remain at conceptual and high level classification of information, which enable different versions of interpretations. Users can have their own ways of using this ontology that different to our expectation, however they need to agree on it before they can start to use it. For example, how to align the three levels (high, medium, low or other classifications) of attack severity and attacker skill. Furthermore, users are expected to add new classes and instances into the ontology according to the circumstance of each sharing community, thus users have to build rules in advance such as how to make decision on what to add. Therefore, this ontology usage is limited that users must make their own rules before using it.

Finally, because each information sharing community formulates their own rules and standards for using this ontology, this ontology cannot be generalised among different communities. In addition to the example of levels of attack severity, different communities may also have distinct ways of calculating the number of attacks and extending the class hierarchy. As a consequence, if government want to integrate several attack pattern databases of different sharing communities, a transforming work is needed to align data that based on different rules and standards.

7.3 Recommendation for further research

During the process of framing this thesis, the continuity of research is reflected. As explained above, the current scope can be expanded. Besides, the current findings of this research can contribute to subsequent studies. Some possible areas of future research are described here.

The information sharing level can be extended to inter-department and international. This research only discussed inter-organisational level attack pattern sharing in the Netherlands. The situation is more complex in reality that information can also be shared between different departments or between organisations in different countries. It is relevant to analyse the interdependencies between IT department and non-IT department as well as to discuss different regulations on cyber attack or information sharing in different countries.

Additionally, the characteristics of each industry can be integrated into the design of attack pattern ontology, i.e. develop different ontologies for different information sharing communities. For example, in the healthcare industry, physical attack is one of the main problems; therefore all the class branches relevant to physical attacks will have more subclasses and bigger hierarchical depth. Furthermore, it is also recommended to do a research on organisations' needs that if industry sector is the best way to form information sharing communities. It is possible that organisations are more willing to share attack information with partners instead of competitors in the same industry. In many cases, business partners are in different industries to gain mutual benefits. If in reality this is preferred, then the problems is that whether sharing attack information with partners is effective or as effective as sharing attack information with competitors in the same industry.

Finally, the attack pattern ontology can be combined with attacker information ontology. This will lead to a more powerful attack model; attack pattern captures the attacker's perspective and attacker profile records the corresponding abilities of each attackers. For example, one attack pattern presents the attack skill needed and resource required, the attacker profile match each attacker with their skills and resources, people can use the combination of the two to check which attackers are capable of employing this type of attack.

7.4 Lessons learned beyond the attack pattern ontology

This thesis deals with a social-technical issue that we developed a shareable attack pattern ontology as the common language of sharing attack information. Beyond the complexity and uniqueness of this problem, some lessons are learned from the process of developing this ontology.

In general, we found that generating requirements for ontology may be effective and helpful in framing the proposed ontology design, especially when the topic is complex and unfamiliar to the designer. In the existing ontology developing methods, designers were suggested to develop several questions that the ontology should answer, which is the only requirement on the ontology. However, these questions are the initial sketch of the ontology (Noy & McGuinness, 2001), thus cannot comprehensively cover all the required aspects of the design. Furthermore, requirements in the format of questions are not easy for designers to follow and fulfill. These two disadvantages will be enlarged when the topic of the ontology is complex and unfamiliar such as the topic of our ontology. In order to prevent these advantages, we generated requirements to help decision-making during the process of designing the attack pattern ontology.

In addition to the practical lesson learned, readers may also learn from or utilize the information from this thesis:

- What is attack pattern?

Attack pattern is a method for defenders or victims to capture the perspective of attackers, thus both the defender's perspective and the attacker's perspective are employed in attack pattern. Moreover, different to many attack descriptions, attacker's method and target are necessary parts of an attack pattern that attack pattern captures attacker's perspective. The basic difference between attack pattern and attack is that attack pattern presents attacks in a more generic way; attack pattern shows the repeatedly parts of multiple similar attacks. These characteristics contribute to the analysis of cyber attacks that researchers can analyse attacker's choices with a clear overview.

- What are the considerations on the contents of shareable attack information?

Information sharing is an activity that involves multiple actors with different roles including information producer and information consumer. Therefore a successful information sharing process balances the interests and conflicts between different actors. Generally information consumer wants information as much as possible and as soon as possible. However, information producer does not want others to match them with the information they provided, especially for the case of attack information. So one of the important rules of attack information sharing is to keep information producer anonymous.

Furthermore, the shared information should be reusable for information producer and easy to gather and process for information consumer. For this purpose, we altered the correspondence relationship between attack pattern and attack in this research from many-to-one to one-to-one.

The choice of ontology itself is also a consideration of sharing useful information and easily producing information. Ontology adds semantics and relations to the system it describes, which is superior to other types of data models that these descriptions are understood by machine. No matter producing information or consuming information, users can save time and effort with the assistance of automatic classification. For instance, when we manually define attack pattern A 'employ' attack method M, it is automatically defined that attack method M 'is employed by' attack pattern A.

- What are the potential usages of an ontology in the domain of cyber security?

Some general usages of sharing attack information are achievable for an ontology, such as gaining better understanding of the security environment, learn from other organisations' experience, prepared for possible attacks to avoid them or to reduce the harm of them. In addition to these, the speciality of ontology is adding semantics and do reasoning. When sharing an ontology in the domain of cyber security, people can easily integrate or share multiple databases, query the computer to process data including filtering, counting, judging. Therefore people can use the ontology to assist decision-making on various activities.

7.5 Reflection on the graduation project

This project was conduced entirely internally. It might bring more interesting outcomes if I have conducted external interviews. It is beneficial to see how users in different organisations or industries think about this ontology. There might be conflicting opinions; how to solve these conflicts would be a main concern of the ontology design. Furthermore, their needs would be helpful for the forming of the requirements. In the evaluation stage, it would be better to go back to the same interviewee ask for their feedbacks. In addition, cyber security researchers could also be invited in the ontology evaluation. Thus the problems on understanding glossaries and terminologies would have been avoided.

As a SEPAM project, this project presents the complexity of reality that the design was developed from different angles including the actor perspective, engineering perspective and process management perspective. This can be seen as a reflection of all different events and activities; everything could have multiple facets including technical facets and non-technical facets. For the attack pattern ontology, the technical part is more objective with little variability; the non-technical part is more uncertain with psychological factors. For example, for the problem of sharing attack information, the advantages of sharing information are bigger than the disadvantages; if we only consider the technical part, it is a easy calculation that people should share attack information; however if we consider the non-technical part, there are so many difficulties about trust and power, people might decide that they would rather not taking the advantages.

Sharing attack information is a powerful means of attack defence. In the future, it would be a widespread activity. Defenders can survive without external help in the past because of small number of attackers and high threshold of becoming attacker. Nowadays, the developing attack technology and tools significantly lowered the threshold of becoming attacker that people without any IT knowledge can perform attacks. As a result, organisations and individuals are suffering from large amount and numerous types of cyber attacks. Being pushed by the intensified threat environment, organisations have to share attack information and try to overcome the impediments of information sharing. The problem of sharing attack information is not solved by this single project, but I hope its outcomes can contribute to the cyber security domain and generate interests in the subjects of attack pattern and attack information sharing through ontology.

Reference

Abrams, M., & Weiss, J. (2008). Malicious control system cyber security attack case study– Maroochy Water Services, Australia. *McLean, VA: The MITRE Corporation*.

Anderson, R. (1993, December). Why cryptosystems fail. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (pp. 215-227). ACM.

Arbaugh, W. A., Fithen, W. L., & McHugh, J. (2000). Windows of vulnerability: A case study analysis. *Computer*, *33*(12), 52-59.

Barnum S. (2008). *Common attack pattern enumeration and classification (CAPEC) schema description.* Retrieved from https://capec.mitre.org/documents/documentation/CAPEC_Schema_Description_v1.3.pdf

Barnum, S., & Sethi, A. (2007). Attack patterns as a knowledge resource for building secure software. In *OMG Software Assurance Workshop: Cigital.*

Bauer, J. M., & Van Eeten, M. J. (2009). Cybersecurity: Stakeholder incentives, externalities, and policy options. *Telecommunications Policy*, *33*(10), 706-719.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.Blackwell, C. (2012). A Strategy for Formalizing Attack Patterns. *Cyberpatterns 2012*, 35.

Bittner, K. (2002). *Use case modeling*. Addison-Wesley Longman Publishing Co., Inc..

Blackwell, C. (2012). A Strategy for Formalizing Attack Patterns. *Cyberpatterns 2012*, 35.

Caracciolo, C. (2006). Designing and implementing an ontology for logic and linguistics. *Literary and linguistic computing*, *21*(suppl 1), 29-39.

Crimmins, Falk, Fowler, Gravel, Kouremetis, Poremski, Sitarz, Sturgeon & Zhang (2014). *CERIAS Tech Report 2014-3 U.S. Bank of Cyber: An analysis of Cyber Attacks on the U.S. Financial System*. Purdue University.

D'Amico, A., Buchanan, L., Goodall, J., & Walczak, P. (2010, April). Mission impact of cyber events: scenarios and ontology to express the relationships between cyber assets, missions and users. In *Proceedings of 5th International Conference on Information Warfare and Security* (pp. 8-9).

ENISA (2013). ENISA Annual Report 2013. *European Union Agency for Network and Information Security (ENISA), Tech. Rep*.

ENISA (2015). ENISA Work Programme 2015. *European Union Agency for Network and Information Security (ENISA), Tech. Rep*.

Euzenat, J. (1996, November). Corporate memory through cooperative creation of knowledge bases and hyper-documents. In *Proc. 10th workshop on knowledge acquisition (KAW), Banff (CA), pages (36)* (pp. 1-18).

Fernandez, E. B., Larrondo-Petrie, M. M., Sorgente, T., & VanHilst, M. (2006). A methodology to develop secure systems using patterns. *Chapter*, *5*, 107-126.

Fernandez, E. B., VanHilst, M., Petrie, M. M. L., & Huang, S. (2006). Defining security requirements through misuse actions. In *Advanced Software Engineering: Expanding the Frontiers of Software Technology* (pp. 123-137). Springer US.

Fernandez, E., Pelaez, J., & Larrondo-Petrie, M. (2007). Attack patterns: A new forensic and design tool. In *Advances in Digital Forensics III* (pp. 345-357). Springer New York.

Fischer, E., A. (2014). *Cybersecurity Issues and Challenges: In Brief.* Retrieved from https://www.fas.org/sgp/crs/misc/R43831.pdf

Foley, S. N., & Fitzgerald, W. M. (2011). Management of security policy configuration using a Semantic Threat Graph approach. *Journal of Computer Security*, *19*(3), 567-605.

Gal-Or, E. & Ghose, A. (2004). The economic consequences of sharing security information. In Economics of information security (pp. 95-104). Springer US.

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2005). A theoretical framework for ontology evaluation and validation. In *SWAP* (Vol. 166).

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2006). *Modelling ontology evaluation and validation* (pp. 140-154). Springer Berlin Heidelberg.

Geers, K. (2011). *Strategic cyber security*. Kenneth Geers.

Gegick, M., & Williams, L. (2007). On the design of more secure software-intensive systems by use of attack patterns. *Information and Software Technology*, *49*(4), 381-397.

GFI. (2009). *Targeted cyber attacks.* Retrieved from GFI: http://www.gfi.com/whitepapers/cyber-attacks.pdf

Gomaa, H., & Olimpiew, E. M. (2005). The role of use cases in requirements and analysis modeling. In *Workshop on Use Cases in Model-Driven Software Engineering. Montego Bay, Jamaica*.

Gómez-Pérez, A. (2004). Ontology evaluation. In *Handbook on ontologies* (pp. 251-273). Springer Berlin Heidelberg.

Grüninger, M., & Fox, M. S. (1995). Methodology for the Design and Evaluation of Ontologies.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. *International journal of human-computer studies*, *43*(5), 907-928.

Hansman, S., & Hunt, R. (2003). A taxonomy of network and computer attack methodologies. *Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand*, *7*.

Harrison, K., & White, G. (2012, November). Information sharing requirements and framework needed for community cyber incident detection and response. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for* (pp. 463-469). IEEE.

Heflin, J. (2009). OWL Web Ontology Language-Use Cases and Requirements. *W3C Recommendation*, 10, 12.

Hendler, J., & McGuinness, D. L. (2000). The DARPA agent markup language. *IEEE Intelligent systems*, *15*(6), 67-73.

Hlomani, H., & Stacey, D. (2014). Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal, na (na)*, 1-5.

Hoglund, G. & McGraw, G. (2004, August). Exploiting Software: How to Break Code. In *Invited Talk, Usenix Security Symposium, San Diego*.

Horridge, M., Brandt, S., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. *The University of Manchester*.

Huang, J. Y., Liao, I. E., Chung, Y. F., & Chen, K. T. (2013). Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining. *Information Sciences*, *231*, 32-44.

Jarrar, M., Demey, J., & Meersman, R. (2003). On using conceptual data modeling for ontology engineering. In *Journal on data semantics i* (pp. 185-207). Springer Berlin Heidelberg.

Johnson, C., Badger, L., Waltermire, D. (2014). Guide to Cyber Threat Information Sharing

(Draft). *NIST special publication, 800-150*.

Kim, A., Luo, J., & Kang, M. (2005). *Security ontology for annotating resources* (pp. 1483-1499). Springer Berlin Heidelberg.

Krebs, B. (2013e, December 13). *Sources: Target investigating data breach*. Retrieved from Krebs on Security: http://krebsonsecurity.com/2013/12/sources-target- investigating-data-breach/

Krebs, B. (2014a, January 14). *A closer look at the Target malware, part II*. Retrieved from Krebs on Security: http://krebsonsecurity.com/2014/01/a-closer-look-at-the- target-malware-part-ii/#more-24401

Krebs, B. (2014b, January 14). *A first look at the Target intrusion malware*. Retrieved from Krebs on Security: http://krebsonsecurity.com/2014/01/a-first-look-at-the- target-intrusion-malware/

Krebs, B. (2014c, Febuary). *Email Attack on Vendor Set Up Breach at Target*. Retrieved from Kerbs On Security: Email Attack on Vendor Set Up Breach at Target http://krebsonsecurity.com/2014/02/email-attack-on-vendor-set-up-breach- at-target/

Krebs, B. (2014d, January 14). *New clues in the Target breach*. Retrieved from Krebs on Security: http://krebsonsecurity.com/2014/01/new-clues-in-the-target-breach/

Kumar, S., & Spafford, E. H. (1994). A pattern matching model for misuse intrusion detection

Laube, S., & Böhme, R. The Economics of Mandatory Security Breach Reporting to Authorities.

Liu, D., Ji, Y., & Mookerjee, V. (2011). Knowledge sharing and investment decisions in information security. *Decision Support Systems*, *52*(1), 95-107.

López, M. F., Gómez-Pérez, A., Sierra, J. P., & Sierra, A. P. (1999). Building a chemical ontology using methontology and the ontology design environment. *IEEE intelligent Systems*, *14*(1), 37-46.

McClure, S., Scambray, J., Kurtz, G., & Kurtz. (2009). *Hacking exposed: network security secrets and solutions*. McGraw-Hill.

Micro, T. (2015, March 4). *The importance of information sharing in beating targeted attacks.* Retrieved from http://blog.trendmicro.com/the-importance-of-information-sharing-in-beating-targeted-attacks/

Microsoft. (2012, June). Determined Adversaries and Targeted Attacks. Retrieved from Microsoft: http://www.microsoft.com/en-us/download/details.aspx?id=34793

Mitre Corporation. (n.d.). *About CAPEC.* Retrieved from https://capec.mitre.org/about/index.html

Miller, J. (2015, January 22). Cyberattacks worry Davos elites. Retrieved from http://www.bbc.com/news/30925696

Moore, A. P., Ellison, R. J., & Linger, R. C. (2001). *Attack modeling for information security and survivability* (No. CMU-SEI-2001-TN-001). Carnegie Mellon university Pittsburgh PA software engineering institute.

Mulligan, J. (2014, February 4). *Time for smartcards*. Retrieved from https://corporate.target.com/discover/article/time-for-smartcards

National Cyber Security Centre. (2012). *Cyber Security Assessment Netherlands CSAN-2.*

National Cyber Security Centre. (2013). *Policy for arriving at a practice for Responsible Disclosure.*

National Cyber Security Centre. (2014). *Cyber Security Assessment Netherlands CSAN-4.* Retrieved from https://english.nctv.nl/Images/cybersecurityassessmentnetherlands2014_tcm92-580598.pdf?cp=92&cs=65035

National Cyber Security Centre. (n.d.). Retrieved from https://www.forensicinstitute.nl/knowledge_lab/partners_in_developing_knowledge/national_cyber_security_centre/

Nicholas, J. M., & Steyn, H. (2012). *Project Management for Engineering Business and Technology*. Butterworth-Heinemann.

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). Protege-2000: an open-source ontology-development and knowledge-acquisition environment. In *AMIA Annu Symp Proc* (Vol. 953, p. 953).

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology.

Obrst, L., Ceusters, W., Mani, I., Ray, S., & Smith, B. (2007). The evaluation of ontologies. In *Semantic Web* (pp. 139-158). Springer US.

Pauli, J. J., & Engebretson, P. H. (2008, April). Towards a specification prototype for hierarchy-driven attack patterns. In *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on* (pp. 1168-1169). IEEE.

Protecting Consumer Information: Can Data Breaches Be Prevented: Testimony before the U.S. House of Representatives, Committee on Energy and Commerce, Subcommittee on Commerce, Manufacturing, and Trade*,* 113th Cong., 41 (2014a) (testimony of Michael R. Kingston).

Radichel, T. (2014). *Case Study: Critical Controls that Could Have Prevented Target Breach.* SANS Institute, InfoSec Reading Room. Available at http://www.sans.org/reading-room/whitepapers/casestudies/case-study-critical-controls-prevented-target-breach-35412

Razzaq, A., Anwar, Z., Ahmad, H. F., Latif, K., & Munir, F. (2014). Ontology for attack detection: An intelligent approach to web application security. *computers & security*, *45*, 124-146.

Robiah, Y., Rahayu, S. S., Sahib, S., Zaki, M. M., Faizal, M. A., & Marliza, R. (2010, June). An improved traditional worm attack pattern. In *Information Technology (ITSim), 2010 International Symposium in* (Vol. 2, pp. 1067-1072). IEEE.

Scarfone, K. A., Souppaya, M. P., Cody, A., & Orebaugh, A. D. (2008). SP 800-115. Technical Guide to Information Security Testing and Assessment.

Schaeffer-Filho, A., & Hutchison, D. (2014). Attack Pattern Recognition Through Correlating Cyber Situational Awareness in Computer Networks. In *Cyberpatterns* (pp. 125-134). Springer International Publishing.

Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., & Sommerlad, P. (2013). *Security Patterns: Integrating security and systems engineering*. John Wiley & Sons.

Simmons, C. B., Shiva, S. G., Bedi, H., Dasgupta, D. (2014, June). AVOIDIT: A cyber attack taxonomy. In *9th Annual Symposium on Information Assurance (ASIA'14)* (p. 2).

Simmons, C. B., Shiva, S. G., & Simmons, L. L. (2014, June). A qualitative analysis of an ontology based issue resolution system for cyber attack management. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on* (pp. 323-329). IEEE.

Slay, J., & Miller, M. (2008). *Lessons learned from the maroochy water breach* (pp. 73-82). Springer US.

Taylor-Powell, E. (1998). Questionnaire Design: Asking questions with a purpose. *University of Wisconsin Extension*.

Tartir, S., Arpinar, I. B., & Sheth, A. P. (2010). Ontological evaluation and validation. In *Theory and Applications of Ontology: Computer Applications* (pp. 115-130). Springer Netherlands.

Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., & Aleman-Meza, B. (2005). OntoQA: Metric-based ontology quality analysis.

Terms of Reference Working Group 2 – Information sharing (2013). Retrieved from https://resilience.enisa.europa.eu/nis-platform/shared-documents/wg2-documents/wg2-nis-platform-terms-of-reference/at_download/file

The 2013 information security breaches survey. (2013). Retrieved from https://www.gov.uk/government/uploads/system/uploads/attachment_data/ file/200455/bis-13-p184-2013-information-security-breaches-survey -technical-report.pdf

The Minister of Security and Justice. (2013). *National Cyber Security Strategy 2.*

The TRESPASS Project, D5.3.1. (2013). *Abstraction levels for model sharing.* (Deliverable D5.3.1)

Thomas, R. Gruber. (1993). A translation approach to portable ontology specifications, knowledge acquisition.

Thonnard, O., & Dacier, M. (2008). A framework for attack patterns' discovery in honeynet data. *digital investigation*, *5*, S128-S139.

Tipton, S., & Choi, Y. (2014). The Target Security Breach: A Case Study.

Tutorial 3: Semantic Modelling. (n.d.). Retrieved from http://www.linkeddatatools.com/semantic-modeling

US-Cert (2014). *ICS-CERT Monitor, January–April 2014*

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, *11*(02), 93-136.

Uzunov, A. V., & Fernandez, E. B. (2014). An extensible pattern-based library and taxonomy of security threats for distributed systems. *Computer Standards & Interfaces*, *36*(4), 734-747.

Vázquez, D. F., Acosta, O. P., Brown, S., Reid, E., & Spirito, C. (2012, June). Conceptual framework for cyber defense information sharing within trust relationships. In *Cyber conflict (CYCON), 2012 4th international conference on* (pp. 1-17). IEEE.

Verizon. (2014). 2014 Data Breach Investigation Report. Retrieved from http://www.verizonenterprise.com/DBIR/2014/reports/rp_dbir-2014-executive-summary_en_xg.pdf

Vrandečić, D. (2009). Ontology evaluation. Springer Berlin Heidelberg.

Web Application Security Consortium. (2010). *WASC,'Threat Classification,'*. Technical report, Version 2.00.

Ye, N., Newman, C., & Farley, T. (2006). A system-fault-risk framework for cyber attack classification. *Information, Knowledge, Systems Management*, *5*(2), 135-151.

Zhu, Y. (2011). Attack pattern discovery in forensic investigation of network attacks. *Selected Areas in Communications, IEEE Journal on*, *29*(7), 1349-1357.

Appendix A Build the class hierarchy

CAPEC (Common Attack Pattern Enumeration and Classification) is an open data resource that provides a comprehensive dictionary of known attacks (Mitre Corporation, n.d.). We use the attack patterns from CAPEC to build our attack pattern class hierarchy. Table 24 shows the 15 attack patterns we used to build the bottom concepts of the class hierarchy; these attack patterns covered all the 6 domains of attack and most (13 out of 16) of the mechanisms of attack that CAPEC provides.

We choose CAPEC because it is the only data source that contains attack patterns from different domains with various mechanisms. Some attack pattern examples are found during literature review, yet those are only examples concentrating on one or two specific types of attacks; we might lose the diversity of the bottom level concepts if we build the hierarchy based on these attack pattern examples. For example, none of the example in literatures mentioned physical attacks. As a consequence of that, we could have missed some unique features of physical attacks in our class hierarchy, such as accessing the target by stealing physical items. Then when users want to use the ontology to share information about physical attacks, which is significant for the healthcare sector (Verizon, 2014), they have to add new top level or middle level classes. We hope our ontology is applicable for most of the attack types with the least adaption; missing higher-level concepts is not the desired situation. Therefore we choose CAPEC as our data source for building the class hierarchy.

Table 24 Instances for class hierarchy bottom concepts

CAPEC	Domains of attack	Mechanisms of attack
2	Software	Abuse of functionality
50	Software	Abuse of functionality
100	Software	Manipulate data structures
147	Software	Deplete resources
498	Software	Gather information
66	Software	Injection
68	N/A	Exploitation of authorisation
25	N/A	Manipulate timing and state
28	N/A	Probabilistic Techniques
62	N/A	Exploitation of authentication
275	Communications	Manipulate resources
169	Hardware; Software	Gather information
418	Social engineering	Manipulate system users
507	Physical security	Gain physical access
521	Supply chain	N/A

In the tables below (Table 25 -Table 32), we show the bottom-up way of adding middle level concepts from bottom level attack patterns. Totally 8 top-level concepts exist. We present the bottom level concepts of each attack pattern from Table 24 (on the left); then we try to link each bottom level concept with the top-level concept with a middle level concept (on the right). The top-level concepts are stated in the table titles.

Table 25 Build the class hierarchy - top-level concept 1: Prerequisites

Bottom level concepts		Connecting concepts	
2	The system has account lockout mechanism	Target performs specific function	
50	The system allows users to recover their passwords and gain access back into the system.	Target performs specific function	
100	Targeted software performs buffer operations	Target performs specific function	
147	Attacker know and can access to endpoint of web service	Remote access	Access to the target
498	Physical access to a device	Physical access to the target	Access to the target
66	SQL queries used by the application to store, retrieve or modify data. User controllable input.	Target performs specific function	
68	Place code in the victim container	Remote access	Access to the target
25	The target host has a deadlock condition The target host exposes an API to the user	Target performs specific function	
28	No specific prerequisites	No specific prerequisites	
62	No specific prerequisites	No specific prerequisites	
169	No specific prerequisites	No specific prerequisites	
418	No specific prerequisites	No specific prerequisites	
507	It requires the existence of a physical target that an adversary believes hosts something of value	Existence of a specific target	
521	Access to the manufacturer's documentation through remote compromise or physical access	Physical access Remote access	Access to the target

Few attack patterns show the required resource, which could be unknown or not necessary for most of the attacks.

Table 26 Build the class hierarchy - top-level concept 2: Resource required

Bottom level concepts		Connecting concepts
50	For a brute force attack one would need a machine with sufficient CPU, RAM and HD	Material resource required
68	Deployed code that has been signed by its authoring vendor or a partner	Material resource required
28	Fuzzing tools	Material resource required

Table 27 Build the class hierarchy - top-level concept 3: Skill or Knowledge required

Bottom level concepts		Connecting concepts
2	An attacker must be able to reproduce behaviour that would result in an account being locked	Investigate system feature
50	Brute force attack; social engineering and more sophisticated technical attacks	Investigate system feature
		Knowledge and skill of specific attack method
100	Notice an overflow and stuff an input variable with content	Investigate system feature
	Detailed knowledge of the target system architecture and kernel	Knowledge of specific software
147	To generate a large amount of small XML based messages and send to the target service	Knowledge and skill of specific attack method
498	No specific knowledge and skill required	No specific knowledge and skill required
66	Basic SQL knowledge	Knowledge of specific computer language
	Specific knowledge of the database	Knowledge of specific software
68	Understand cryptographic operations in good detail Knowledge of the platform specific mechanisms of signing and verifying code.	Knowledge and skill of specific software
25	Need knowledge about the system's resources and APIs	Knowledge of specific software
28	No specific knowledge and skill required	No specific knowledge and skill required
62	Craft malicious web links and distribute them	Investigate system feature
169	Send HTTP requires, run the scan tool	Knowledge and skill of specific attack method

418	Social engineering technique	Knowledge and skill of specific attack method
507	No specific knowledge and skill required	No specific knowledge and skill required
521	Advanced knowledge of hardware capabilities of a manufacturer's product.	Knowledge of specific hardware
	Ability to read, interpret, and subsequently alter manufacturer's documentation to cause errors in design specifications	Knowledge of specific computer language

Table 28 Build the class hierarchy - top-level concept 4: Target

Bottom level concepts		Connecting concepts		
2	System that has account lockout mechanism	Network		
50	Application	Application		Software
100	Software memory	Application		Software
147	Web service	Network		
498	iOS Application	Application	OS	Software
66	Software that constructs SQL statements	Application		Software
68	Code signing way of virtual machine	Application		Software
25	Software	Application		Software
28	Software	Application		Software
62	Induce service users to click link	Application		Software
169	Application	Software		
	Network	Network		
418	Individuals	User		
507	System or device	Hardware		
521	Manufacturing system	Hardware		

Table 29 Build the class hierarchy - top-level concept 5: Vulnerabilities

Bottom level concepts		Connecting concepts
2	Uncontrolled resource consumption	Unrestricted Consumption
50	Weak Password Recovery Mechanism for Forgotten Password	Weak Cryptography
100	Buffer overflow	Faulty buffer access
		Insufficient input validation
147	Allocation of resources without limits or	Unrestricted Consumption

	throttling	
498		Weak physical protection
66	Improper SQL command	Insufficient input validation
68	Missing required cryptographic step	Weak Cryptography
25	Improper Synchronization	Misconfiguration
28	Improper Input Validation	Insufficient input validation
62	Cross-Site Request Forgery	Insufficient authentication validation
169	Exposure of System Data to an Unauthorized Control Sphere	Misconfiguration
	Exposure of Sensitive Data Through Data Queries	Insufficient input validation
	Missing encryption of sensitive data	Weak Cryptography
	Cleartext Storage of Sensitive Information	
	Incorrect default permission	Incorrect File permission
418	Perception of obligation	Social engineering
507		Weak physical protection
521		Weak physical protection

Table 30 Build the class hierarchy - top-level concept 6: Method

Bottom level concepts		Connecting concepts	
2	Flooding	Denial of service	
	API Abuse		
	Brute Force	Guessing	
147	Flooding	Denial of service	
50	Brute Force	Guessing	
	API Abuse		
	Injection	Database attack	Web compromise
100	Analysis		
	Injection		
498	Gaining access to the image files, or physically obtaining the device and leveraging the multitasking switcher interface	Physical access	
66	Injection		
68	Spoofing		
	API Abuse		
25	Analysis		
	API Abuse		

54	Analysis		
	Injection	Database attack	Web compromise
	Brute Force	Guessing	
62	Spoofing		
	Analysis		
169	Protocol manipulation		
	Injection		
	Analysis		
418			
507	Theft of item	Physical attack	
521		Physical attack	

Table 31 Build the class hierarchy - top-level concept 7: Countermeasures

Bottom level concepts		Connecting concepts
2	Implement intelligent password throttling mechanisms such as those which take IP address into account, in addition to the login name.	Design
		Reduce the negative effect or probability of the attack
50	Use multiple security questions	Design
		Reduce the negative effect or probability of the attack
	E-mail the temporary password to the registered e-mail address of the user	Design
		Reduce the negative effect or probability of the attack
100	Use secure functions not vulnerable to buffer overflow	Design
		Reduce the negative effect or probability of the attack
	User a language or complier that performs automatic bounds checking	Design
		Avoid the attack
147	Build throttling mechanism into the resource allocation. Provide for a timeout mechanism for allocated resources whose transaction does not complete within a specified interval.	Design
		Reduce the negative effect or probability of the attack
498	An application that may display sensitive information should clear the screen contents before a screenshot is taken	Implementation
		Reduce the negative effect or probability of the attack
66	Input validation	Implementation
		Avoid the attack

	Use of parameterized queries or stored procedures	Design
		Reduce the negative effect or probability of the attack
68	Use well-known cryptography appropriately and correctly Avoid reliance on flags or environment variables that are user-controllable	Design
		Reduce the negative effect or probability of the attack
25	Use known algorithm to avoid deadlock condition	Design
	For competing actions use well-known libraries which implement synchronization	Reduce the negative effect or probability of the attack
28	Test and uncover any assumptions or unexpected behavior	Implementation
		Avoid the attack
62	Use cryptographic tokens to associate a request with a specific action	Design
		Reduce the negative effect or probability of the attack
169	Keep patches up to date	Configuration
		Reduce the negative effect or probability of the attack
	Change default passwords	Design
		Reduce the negative effect or probability of the attack
	Place sensitive information offline	Design
		Reduce the negative effect or probability of the attack
	Curtail unexpected input	Implementation
		Avoid the attack
418	Education to the type of attacks	Other
		Avoid the attack
	Be aware of the information exposed	Other
		Reduce the negative effect or probability of the attack
507	Physical security techniques such as locks doors, alarms, and monitoring of targets	Implementation
		Reduce the negative effect or probability of the attack
521		

Table 32 Build the class hierarchy - top-level concept 8: Consequence

Bottom level concepts	Connecting concepts

2	An attacker can lock a legitimate user out of their own account	Resource consumption
50	An attacker can gain access into the system with the same privileges as the original user.	Gain privileges
100	Program crash	Resource consumption
	Redirection of execution	Gain privileges
147	Exhausting the resources of a web service	Resource consumption
498	Information exposure	Information disclosure
66	Add of modify data in the database	Modification
	Read data	Information disclosure
68	Gain privileges	Gain privileges
25	Denial of service	Resource consumption
28	Modify memory	Modification
	Read application data	Information disclosure
		Gain privileges
	Alter execution logic	Modification
62	Read application data	
	Modify application data	
	Gain privileges	
169	Get information about the system and organization	Information disclosure
418		Information disclosure
507		Information disclosure
521		Modification

Appendix B Attack pattern instances

Table 33 Attack pattern instance 2: critical infrastructure

<table>
<tr><td colspan="4">Attack pattern: Public utility compromised (domain)</td></tr>
<tr><td colspan="4">Execution/description:
- IT systems have not been rigorously audited for vulnerabilities and configuration mistakes
- Hackers can easily use search engines such as google and SHODAN (computer search engine) to find internet-connected control systems
- These systems are accessed and analysed by attackers
- The assessment on the system identified previous intrusion activity that the utility was attacked before</td></tr>
<tr><td>Property</td><td>Individual or data (range)</td><td>Class path (range)</td><td>Annotation</td></tr>
<tr><td>hasName</td><td>Public utility compromised</td><td>n/a</td><td></td></tr>
<tr><td>hasID</td><td>200</td><td>n/a</td><td></td></tr>
<tr><td>hasNumberOfOccurredAttacks</td><td>1</td><td>n/a</td><td></td></tr>
<tr><td>hasPrerequisites</td><td>Accessible through internet</td><td>- Prerequisites
- Target performs specific function</td><td>Easy for hackers using search engines such as google and SHODAN (computer search engine) to find internet-connected control systems</td></tr>
<tr><td>hasResourceRequired</td><td>No specific resource required</td><td>-Resource required
- No specific resource required</td><td></td></tr>
<tr><td>hasSkillOrKnowledgeRequired</td><td>No specific skill or knowledge requried</td><td>- Skill or knowledge required
- No specific skill or knowledge requried</td><td></td></tr>
<tr><td>hasSkillOrKnowledgeLevel</td><td>Low</td><td>-Skill or knowledge level
-Low</td><td></td></tr>
</table>

exploit	Misconfiguration mistakes	- Vulnerabilities - misconfiguration	IT systems have not been rigorously audited for vulnerabilities and configuration mistakes
employ	brute force	-Method -password attack -guessing -brute force	Standard brute force, try different combinations of passwords until the right one is found
hasTarget	A control system	Target - Network	Not identified utility
isWorkedAgainst	onsite cybersecurity assessment	- Countermeasures - avoid the attack	Conduct an onsite cybersecurity assessment to evaluate the overall security posture of the victim's infrastructure
		- Countermeasures - configuration	
isWorkedAgainst	Practical recommendations	- Countermeasures - avoid the attack	practical recommendations for re-architecting and securing the control network
		- Countermeasures - design	
hasConsequences	gain privileges	- Consequence - gain privileges	unauthorized access to its control system network
hasTypicalSeverity	High	-Typical Severity -High	

Source: adapted from US-Cert (2014)

Table 34 Attack pattern instance 3: water industry

Attack pattern: SCADA system malfunctioning (domain)
Execution/description: - Attacker accessed computers controlling the sewerage system - Drove around the area and issue radio commands to the sewage equipment - Pump station settings kept changing automatically and either does not perform or perform incorrect tasks - Technicians correct faults at affected pumping stations

- The attack was stopped until attacker was caught			
Property	Individual or data (range)	Class path (range)	Annotation
hasName	SCADA system malfunctioning	n/a	
hasID	300	n/a	
hasNumberOfOccurredAttacks	1	n/a	
hasPrerequisites	Access controlling computer	-Prerequisites -Access to target -Remote access	Access computers controlling the sewerage system
hasResourceRequired	Laptop	-Resources required -Material resource	
	radio transmitter	-Resources required -Material resource	
hasSkillOrKnowledgeRequired	Understand the IT/control system	- Skill or knowledge required - Knowledge and skill of specific software	Detailed knowledge about the victim's IT/control system, skill to disguise attack actions
hasSkillOrKnowledgeLevel	High	-Skill or knowledge level -High	
exploit	Lack of authentication activities	-Vulnerabilities -Insufficient authentication validation -Missing function level access control	
employ	Send bogus radio message	-Method -Network attack - Wireless attack	Use PDS software file to run or access the computers in the sewerage system
hasTarget	Client of SCADA system	-Target -Network -Application	The attacker probably helped install the equipments

		-Client	
isWorkedAgainst	Specialized intrusion detection system	-Countermeasures -Avoid attack	
		-Countermeasures -Design	
	Logging activities and events	-Countermeasures -Avoid attack	
		-Countermeasures -Design	
	Correct faults at affected pumping stations	-Countermeasures -Reduce negative effect and possibility of the attack	
		-Countermeasures - implementation	
hasConsequences	Gain privileges	-Consequence - Gain privileges	
	Altering electronic data	-Consequence -Modification	Altering electronic data in particular sewerage pumping stations and causing malfunctions in their operations: pumps were not running when they should have been; alarms were not reporting to the central computer; loss of communication between the central computer and various pumping stations
hasTypicalSeverity	High	-Typical Severity -High	

Source: adapted from Slay & Miller, (2008); Abrams & Weiss, (2008)

Table 35 Attack pattern instance 4: financial industry

Attack pattern: Illegal wire transfers (domain)
Execution/description:

<table>
<tr><td colspan="4">-Attackers form a group where a wire transfer clerk is in
-The wire transfer clerk provided the account numbers and credentials to the attack group
-One of the other members pretend to be a representative of a company and call the wire transfer clerk at the bank (while working) to place a wire transfer request with him
-The clerk processed the request and the money was transferred to a foreign account
-The clerk calls back to one of the member’s home number to confirm the transfer and pretend he was calling the representative at the company
-After confirming that the transfer worked, they conducted two more transfers via the same method</td></tr>
<tr><td>Property</td><td>Individual or data (range)</td><td>Class path (range)</td><td>Annotation</td></tr>
<tr><td>hasName</td><td>Illegal wire transfers</td><td>n/a</td><td></td></tr>
<tr><td>hasID</td><td>400</td><td>n/a</td><td></td></tr>
<tr><td>hasNumberOfOccurredAttacks</td><td>3</td><td>n/a</td><td></td></tr>
<tr><td>hasPrerequisites</td><td>No specific prerequisites</td><td>-Prerequisites
-No specific prerequisites</td><td></td></tr>
<tr><td rowspan="2">hasResourceRequired</td><td>Insider</td><td>Resources required - Human resource</td><td>One of the attack group member is a wire transfer clerk who can legally conduct wire transfers and provide account credentials</td></tr>
<tr><td>Outsider</td><td>Resources required - Human resource</td><td>A group of people can play different roles in the process of wire transfer</td></tr>
<tr><td>hasSkillOrKnowledgeRequired</td><td>No specific skill or knowledge required</td><td>-Skill or knowledge required
-No specific skill or knowledge required</td><td></td></tr>
<tr><td>hasSkillOrKnowledgeLevel</td><td>Low</td><td>-Skill or knowledge level
-Low</td><td></td></tr>
<tr><td rowspan="2">exploit</td><td>Find human resource</td><td>Vulnerabilities - Social engineering</td><td>The initiator contacted a bank teller, and the bank teller brought in a wire transfer clerk of the same bank</td></tr>
<tr><td>Disguise as legal service</td><td>Vulnerabilities - Insufficient</td><td>The attack cannot be detected immediately as the</td></tr>
</table>

	request	authentication validation	attack group followed the normal and legal process to transfer money
employ	Abuse privilege	Method - Physical attack	The attackers gain the bank customer account information directly from records by abusing employee’s privilege
hasTarget	Bank account numbers and credentials	Target - User	Bank account numbers and credentials
isWorkedAgainst	Frozen affected accounts	-Countermeasures -Reduce effect and possibility of the attack	
		-Countermeasures -Implementation	
hasConsequences	Bank account information disclosure	Consequence - Information disclosure	
hasTypicalSeverity	Low	-Typical Severity -Low	

Appendix C OWL code

The codes below are abbreviated from the original version; this is to show how the OWL language works.

```
<?xml version='1.0'?>


<!DOCTYPE rdf:RDF [
<!ENTITY ecorse_base 'http://ecorse.fr/simulateur/schema/ecorse_base#' >
]>



  <!--
  ///////////////////////////////////////////////////////////////////////////////////////
  //
  // Annotation properties
  //
  ///////////////////////////////////////////////////////////////////////////////////////
   -->




  <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Execution -->

  <owl:AnnotationProperty rdf:about='&ecorse_base;Execution'>
     <rdfs:comment>Description and execution flow of the attack pattern</rdfs:comment>
  </owl:AnnotationProperty>



  <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Version -->

  <owl:AnnotationProperty rdf:about='&ecorse_base;Version'/>



  <!-- http://ecorse.fr/simulateur/schema/ecorse_base#status -->
```

```
    <owl:AnnotationProperty rdf:about=‘&ecorse_base;status’/>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->




    <!-- http://ecorse.fr/simulateur/schema/ecorse_base#employ -->

    <owl:ObjectProperty rdf:about=‘&ecorse_base;employ’>
        <rdf:type rdf:resource=‘&owl;AsymmetricProperty’/>
        <rdf:type rdf:resource=‘&owl;IrreflexiveProperty’/>
    </owl:ObjectProperty>



    <!-- http://ecorse.fr/simulateur/schema/ecorse_base#exploit -->

    <owl:ObjectProperty rdf:about=‘&ecorse_base;exploit’/>



    <!-- http://ecorse.fr/simulateur/schema/ecorse_base#hasConsequences -->

    <owl:ObjectProperty rdf:about=‘&ecorse_base;hasConsequences’>
        <rdf:type rdf:resource=‘&owl;AsymmetricProperty’/>
        <rdf:type rdf:resource=‘&owl;IrreflexiveProperty’/>
    </owl:ObjectProperty>


  <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Data properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->
```

```
<!-- http://ecorse.fr/simulateur/schema/ecorse_base#hasID -->

<owl:DatatypeProperty rdf:about=‘&ecorse_base;hasID’>
   <rdf:type rdf:resource=‘&owl;FunctionalProperty’/>
</owl:DatatypeProperty>



<!-- http://ecorse.fr/simulateur/schema/ecorse_base#hasName -->

<owl:DatatypeProperty rdf:about=‘&ecorse_base;hasName’>
   <rdf:type rdf:resource=‘&owl;FunctionalProperty’/>
</owl:DatatypeProperty>



<!-- http://ecorse.fr/simulateur/schema/ecorse_base#hasNumberOfOccurredAttacks -->

<owl:DatatypeProperty rdf:about=‘&ecorse_base;hasNumberOfOccurredAttacks’/>



<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////////////
 -->




<!-- http://ecorse.fr/simulateur/schema/ecorse_base#API_Abuse -->

<owl:Class rdf:about=‘&ecorse_base;API_Abuse’>
   <rdfs:subClassOf rdf:resource=‘&ecorse_base;Misuse_of_Resources’/>
   <owl:disjointWith rdf:resource=‘&ecorse_base;Protocol_Manipulation’/>
</owl:Class>
```

```
 <!-- http://ecorse.fr/simulateur/schema/ecorse_base#AccessToTheTarget -->

 <owl:Class rdf:about=‘&ecorse_base;AccessToTheTarget’>
     <rdfs:subClassOf rdf:resource=‘&ecorse_base;Prerequisite’/>
 </owl:Class>



 <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Application. -->

 <owl:Class rdf:about=‘&ecorse_base;Application.’>
     <rdfs:subClassOf rdf:resource=‘&ecorse_base;Network’/>
 </owl:Class>



 <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Avoid_the_Attack -->

 <owl:Class rdf:about=‘&ecorse_base;Avoid_the_Attack’>
     <rdfs:subClassOf rdf:resource=‘&ecorse_base;Time’/>
     <owl:disjointWith
rdf:resource=‘&ecorse_base;Reduce_the_Negative_Effect_or_Probability_of_the_Attack’/>
 </owl:Class>



 <!--
 ///////////////////////////////////////////////////////////////////////////////////////
 //
 // Individuals
 //
 ///////////////////////////////////////////////////////////////////////////////////////
  -->



 <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Abuse_privilege -->

 <owl:NamedIndividual rdf:about=‘&ecorse_base;Abuse_privilege’>
     <rdf:type rdf:resource=‘&ecorse_base;Method’/>
     <rdf:type rdf:resource=‘&ecorse_base;Physical_Attack’/>
     <isEmployedBy rdf:resource=‘&ecorse_base;Illegal_wire_transfers’/>
 </owl:NamedIndividual>
```

```
    <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Access_controlling_computer -->

    <owl:NamedIndividual rdf:about=‘&ecorse_base;Access_controlling_computer’>
        <rdf:type rdf:resource=‘&ecorse_base;AccessToTheTarget’/>
        <rdf:type rdf:resource=‘&ecorse_base;Prerequisite’/>
        <rdf:type rdf:resource=‘&ecorse_base;RemoteAccess’/>
        <isPrerequisitesOf rdf:resource=‘&ecorse_base;SCADA_system_malfunctioning’/>
    </owl:NamedIndividual>


<!-- http://ecorse.fr/simulateur/schema/ecorse_base#Illegal_wire_transfers -->

    <owl:NamedIndividual rdf:about=‘&ecorse_base;Illegal_wire_transfers’>
        <rdf:type rdf:resource=‘&ecorse_base;Attack_Pattern’/>
        <hasNumberOfOccurredAttacks
rdf:datatype=‘&xsd;integer’>3</hasNumberOfOccurredAttacks>
        <hasID rdf:datatype=‘&xsd;integer’>400</hasID>
        <Execution>- Attacker accessed computers controlling the sewerage system
- Drove around the area and issue radio commands to the sewage equipment
- Pump station settings kept changing automatically and either does not perform or perform
incorrect tasks
- Technicians correct faults at affected pumping stations
- The attack was stopped until attacker was caught</Execution>
        <hasName>Illegal wire transfers</hasName>
    </owl:NamedIndividual>



    <!-- http://ecorse.fr/simulateur/schema/ecorse_base#Public_Utility_Compromised -->

    <owl:NamedIndividual rdf:about=‘&ecorse_base;Public_Utility_Compromised’>
        <rdf:type rdf:resource=‘&ecorse_base;Attack_Pattern’/>
        <hasNumberOfOccurredAttacks
rdf:datatype=‘&xsd;integer’>1</hasNumberOfOccurredAttacks>
        <hasID rdf:datatype=‘&xsd;integer’>200</hasID>
        <hasName>Public Utility Compromised</hasName>
        <Execution>- IT systems have not been rigorously audited for vulnerabilities and
configuration mistakes
- Hackers can easily use search engines such as google and SHODAN (computer search engine)
to find internet-connected control systems
```

```
- These systems are accessed and analysed by attackers
- The assessment on the system identified previous intrusion activity that the utility was attacked
before</Execution>
    </owl:NamedIndividual>



    <!-- http://ecorse.fr/simulateur/schema/ecorse_base#SCADA_system_malfunctioning -->

    <owl:NamedIndividual rdf:about=‘&ecorse_base;SCADA_system_malfunctioning’>
        <rdf:type rdf:resource=‘&ecorse_base;Attack_Pattern’/>
        <hasNumberOfOccurredAttacks
rdf:datatype=‘&xsd;integer’>1</hasNumberOfOccurredAttacks>
        <hasID rdf:datatype=‘&xsd;integer’>300</hasID>
        <Execution>- Attacker accessed computers controlling the sewerage system
- Drove around the area and issue radio commands to the sewage equipment
- Pump station settings kept changing automatically and either does not perform or perform
incorrect tasks
- Technicians correct faults at affected pumping stations
- The attack was stopped until attacker was caught</Execution>
        <hasName>SCADA system malfunctioning</hasName>
    </owl:NamedIndividual>



    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // General axioms
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->

    <rdf:Description>
        <rdf:type rdf:resource=‘&owl;AllDisjointProperties’/>
        <owl:members rdf:parseType=‘Collection’>
            <rdf:Description rdf:about=‘&ecorse_base;hasID’/>
            <rdf:Description rdf:about=‘&ecorse_base;hasName’/>
            <rdf:Description rdf:about=‘&ecorse_base;hasNumberOfOccurredAttacks’/>
        </owl:members>
    </rdf:Description>
    <rdf:Description>
    <rdf:type rdf:resource=‘&owl;AllDisjointClasses’/>
    <owl:members rdf:parseType=‘Collection’>
      <rdf:Description rdf:about=‘&ecorse_base;Cross-site_Scripting_XSS’/>
      <rdf:Description rdf:about=‘&ecorse_base;OS_Command_Injection’/>
      <rdf:Description rdf:about=‘&ecorse_base;SQL_LDAP_Xpath_query_injection’/>
    </owl:members>
  </rdf:Description>
  <rdf:Description>
    <rdf:type rdf:resource=‘&owl;AllDisjointClasses’/>
    <owl:members rdf:parseType=‘Collection’>
      <rdf:Description rdf:about=‘&ecorse_base;Broken_Authentication’/>
      <rdf:Description rdf:about=‘&ecorse_base;Cross_Site_Requires_Forgery’/>
      <rdf:Description rdf:about=‘&ecorse_base;Missing_Function_Level_Access_Control’/>
      <rdf:Description rdf:about=‘&ecorse_base;Unvalidated_Redirects_And_Forwards’/>
    </owl:members>
  </rdf:Description>
  <rdf:Description>
    <rdf:type rdf:resource=‘&owl;AllDisjointClasses’/>
    <owl:members rdf:parseType=‘Collection’>
      <rdf:Description rdf:about=‘&ecorse_base;Denial_Of_Service’/>
      <rdf:Description rdf:about=‘&ecorse_base;Installed_Malware’/>
      <rdf:Description rdf:about=‘&ecorse_base;Misuse_of_Resources’/>
      <rdf:Description rdf:about=‘&ecorse_base;Network_Attack’/>
      <rdf:Description rdf:about=‘&ecorse_base;Password_Attack’/>
      <rdf:Description rdf:about=‘&ecorse_base;Physical_Attack’/>
    </owl:members>
  </rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource=‘&owl;AllDisjointClasses’/>
    <owl:members rdf:parseType=‘Collection’>
      <rdf:Description rdf:about=‘&ecorse_base;Spyware’/>
      <rdf:Description rdf:about=‘&ecorse_base;Trojans’/>
      <rdf:Description rdf:about=‘&ecorse_base;Virus’/>
      <rdf:Description rdf:about=‘&ecorse_base;Worms’/>
    </owl:members>
  </rdf:Description>
</rdf:RDF>



<!-- Generated by the OWL API (version 3.5.1) http://owlapi.sourceforge.net -->
```

Appendix D Ontology evaluation questionnaire

Evaluation questionnaire of the attack pattern ontology

This questionnaire is designed for the evaluation of a master thesis. You will evaluate a metadata ontology that aims at presenting shareable attack patterns. There are 3 sections in finishing this questionnaire: listen to the introduction given by the organiser, finishing 2 tasks on computer, answering questionnaire questions. The estimated time for the whole process is less than 1 hour. There are 6 criteria to be evaluated: accuracy, clarity, conciseness, adaptability, completeness and organisational fitness. You will be asked to answer 6 questions for each of the criterion.

Thank you for your participation and input. All information will be recorded anonymous. Please do not hesitate to ask for assistance from the organiser during your evaluation process.

Step 1

In the introduction, you will be given the information about the following topics:

- What is attack pattern, how is it different from attacks
- What is ontology, what are the advantages and why do we need ontology to present attack pattern
- What are the contents and usages of the attack pattern ontology, how is it built in Protégé
- How to perform actions in Protégé such as adding, deleting, changing name, running reasoner
- What are the 6 aspects to be evaluated in this questionnaire
- Rules of asking the organiser questions, what kinds of answers will be given and what will not

You will get the following documents to assist you:

- A table of the contents of the class hierarchy
- Figure of buttons in Protégé
- A cyber attack case
- 6 questions to be answered

Step 2

In order to get more insights into the ontology and its usages, please follow the 2 use cases below and finish the tasks in Protégé installed in the organiser's laptop:

Name	Create a new attack pattern instance
Summary	Add new attack pattern instance and its name, ID, corresponding consequence, countermeasure, methods, vulnerabilities, etc.
Actor	A user as an information producer
Precondition	This user has accessed into the system

Basic flow	1. Decide the class paths and annotations of each of the aspects of this attack pattern: consequence, countermeasure, method, etc. 2. In the individual tab, go to the instances window, add one instance and name it; in the description window, define its class path ‘Attack pattern’ 3. Select the added attack pattern instance, in the annotations window add its annotations such as execution flow. 4. Select the added attack pattern instance and go to the property assertions window, in the data property assertions section define the value that link to this attack pattern through data properties ‘hasID’ ‘hasName’ ‘hasNumberOfOccurredAttacks’ 5. Go to the class hierarchy window and select consequence, in the instances window, check if there are instances match the consequence of the just added attack pattern: if not, add a new one and define its class path in the description window (same as step 1); or else go to the property assertions window, in the object property assertions section relate the consequence instance(s) with the attack pattern through object properties 6. Repeat step 5 for all other instances that need to be connect with the new attack pattern through object properties 7. Go the menu reasoner → run reasoner, the instances added in step 5 should relate to the new attack pattern through the inverse object properties of the used object properties in step 5
Alternate flows	5. Users do not need to relate these instances with the attack pattern ‘public utility compromised’, instead they can select the attack pattern ‘public utility compromised’ and relate it with all other relevant instances through object properties
Post condition	An new attack pattern instance is created and defined with relevant annotations, data properties and object properties

Name	Use reasoner to automatically compute entities that have consequence of gaining privileges and high typical severity
Summary	A user asks the ontology to do query that filter attack patterns with specific consequence and typical severity
Actor	A user as an information consumer
Precondition	This user has accessed into the system
Basic flow	1. Go to the Query tab 2. Entre the class expression in the query box: Attack_Pattern and (hasConsequences some Gain_Priviledges) and (hasTypicalSeverity some High_Typical_Severity) 3. On the right of the query results box, select the type of entities to filter 4. Synchronise reasoner and execute the query 5. The query results is showed in the query results box
Alternate flows	5. If no results is returned, check the query expression in step 2 and perform step 2, 3, 4, 5 again
Post condition	The information of interests is found

Step 3

Now, you have got an impression of the attack pattern ontology and its usages, please give answers to the following questions:

1. Look into the names of the entities (classes, instances, properties, annotations) and the relationships between entities (instance – object property - instance), can you find the right entities and corresponding relationships for performing your task? If not, which names or description of relationships caused you difficulties?

2. Could you find all the necessary entities (classes, instances, properties, annotations) to fulfil the tasks?

3. Can you easily understand the entities' names (classes, instances, properties, annotations) and descriptions (please go to the description window of classes, properties and instances)? If not, which names or descriptions could be improved?

4. Completeness: Does the ontology give answers to these questions? Which not? Why?

- Can you convert attack steps and strategies into multiple attack methods or vulnerabilities? If not, why?
- Can you see the complete class tree?

- Do you know what should be filled for an attack pattern by reading the class hierarchy? If not, what are not clarified?
- Can you view information per attack pattern, per method, per prerequisites or per vulnerability?
- When viewing an instance, can you see its position in the class hierarchy? When viewing a class, can you see its position in the class hierarchy?
- Can you find information through query?
- Can you reuse existing instances, i.e. can two attack patterns relate to the same method instance, prerequisite instance, etc.?

5. Do you think all the existing elements (classes, instances, properties, annotations) of the ontology are necessary? If not, which elements are redundant?

6. Imagine you are from the financial/ healthcare/ energy/ retail/ telecommunication company, if your company want to use this ontology as the base ontology of attack information sharing, what should be added, changed or deleted?

Appendix E Ontology evaluation result

Five master students of Delft University of Technology gave answers to the questionnaire. Their answers for each of the 6 criteria are showed in the 6 tables below (Table 36 - Table 41). The suggestions mainly concentrate on 2 criteria: clarity and organizational fitness.

Table 36 Evaluation results: accuracy

1. Accuracy: Look into the names of the entities (classes, instances, properties, annotations) and the relationships between entities (instance – object property - instance), can you find the right entities and corresponding relationships for performing your task? If not, which names or description of relationships caused you difficulties?	
1	Yes
2	Yes
3	Yes
4	I can find the required entities. However some are unclear under network and vulnerabilities
5	Classes related to level (e.g. skill or knowledge level) since there is no specific guideline/ criteria, it is subject (it can be changed by who this does)

Table 37 Evaluation results: adaptability

2. Adaptability: Could you find all the necessary entities (classes, instances, properties, annotations) to fulfil the tasks?	
1	Yes all required entities are present, yet the consequences could perhaps be expended and to be more detailed
2	I think so
3	Yes
4	Yes
5	Yes

Table 38 Evaluation results: clarity

3. Clarity: Can you easily understand the entities' names (classes, instances, properties, annotations) and descriptions (please go to the description window of classes, properties and instances)? If not, which names or descriptions could be improved?	
1	Mainly the network items are unclear, session layer, etc. It is hard for an outsider with same knowledge to understand. Perhaps some more generic terms should be applied.
2	I had difficulties since it's not my professional field.
3	Hard to get the meaning of all of the entities, for instance, race conditions.
4	No. Countermeasure; Vulnerability; Relationship between classes, e.g. 'only', 'some', Note: people with IT background probably will understand more easily.
5	Generally they are understandable. But in some section classes are not clear. The perspective of entities is confusing as well. Some classes are ambiguous because lack of

	annotation.

Table 39 Evaluation results: completeness

4. Completeness: Does the ontology give answers to these questions? (See blow, the questions from advantages) Which not and why?	
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes

Table 40 Evaluation results: conciseness

5. Conciseness: When viewing an instance, can you see its position in the class hierarchy? When viewing a class, can you see its position in the class hierarchy?	
1	Yes
2	Seems all meaningful.
3	I do not see anything that is not necessary.
4	It is a difficult question since I am not a professional in this field. I don’t know what elements are not necessary.
5	Hard to answer. Because I have no former background of this subject. Should be improved by some other professional people.

Table 41 Evaluation results: organisational fitness

6. Organisational fitness: Imagine you are from the financial/ healthcare/ energy/ retail/ telecommunication company, if your company want to use this ontology as the base ontology of attack information sharing, what should be added, changed or deleted?	
1	Current ontology is a generic one. But for specific industry, I think they have different focus of information sharing. So some classes, such as vulnerability, method, might be different.
2	Description of the classes & subclasses in a simpler way. Subclasses based on each industry characteristic. Those that are not relevant could be deleted or added, e.g. subclasses of consequence & countermeasure
3	I will categorise several special /usually attacks that my company focus, to make the ontology more specific and smaller. So workers in my company can understand and apply this ontology quickly.
4	Some are B2B but some are B2C (financial, retail, telecom) which have more impacts to society, so probably more classes (e.g. consequences) are needed
5	It currently does not give an easy to use view from an outsider perspective. While most have an IT department with people who would understand, some kind of manual and interface to easily share information should be added. Currently it seems too time consuming.

Appendix F Property restriction comparison

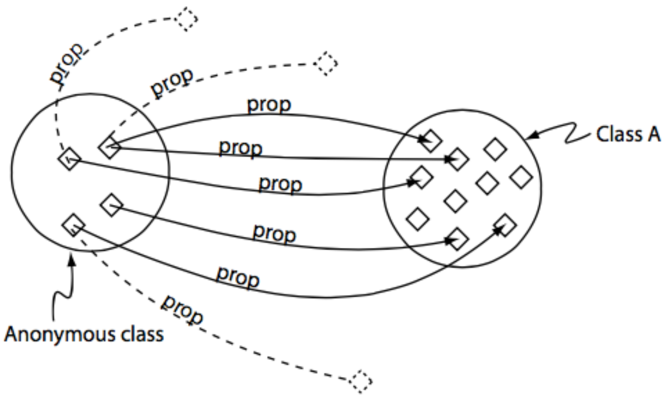


Figure 15 Quantifier restriction: Existential restriction (some ∃)

Source: Horridge, M., Brandt, S., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. *The University of Manchester*.

As presented in Figure 15, all the individuals in the anonymous class (an unnamed class) have at least one specific relationship to individuals in Class A: ∃ prop ClassA. An individual could have one or multiple relations to individuals in Class A using this property; it could also has this relation to individuals in other classes. The emphasis is on 'all individuals' in the anonymous class; it does not require all individuals in class A to relate back.

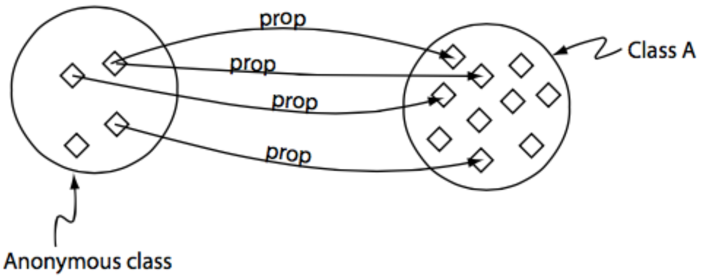


Figure 16 Quantifier restriction: Universal restriction (only ∀)

Source: Horridge, M., Brandt, S., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. *The University of Manchester*.

According to Figure 16, individuals in the anonymous class (an unnamed class) only have relationships to individuals in Class A: ∀ prop ClassA. Not all the individuals in the anonymous class have such relationship to individuals in Class A. An individual could have multiple such relations to individuals in class A; it does not have this relation to individuals in other classes. The emphasis is on Class A; this property must be with an individual of Class A. Same to the existential restriction, it does not require all individuals in class A to relate back. A drawback of this restriction is that the anonymous class also includes individuals that do not participate in this relationship at all. Therefore a common way to define the allValuesFrom relation is to combine existential and universal restrictions (Horridge et al., 2011). As a result of that, the isolate individual in Figure 16 will be excluded from the anonymous class.

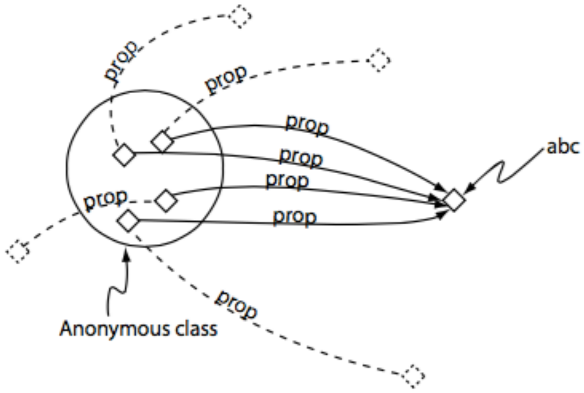


Figure 17 hasValue restriction (∋)

Source: Horridge, M., Brandt, S., Jupp, S., Moulton, G., Rector, A., Stevens, R., & Wroe, C. (2011). A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3. *The University of Manchester*.

Contrast hasValue restriction (Figure 17) with the existential restriction (Figure 15), hasValue restriction describes the relation to a *specific individual* instead of any individual from a *specific class*. The hasValue restriction relates individuals in the anonymous class to the *individual abc* whereas the existential restriction relates individuals in the anonymous class to the individuals in *Class A*. In our research, we do not need this hasValue restriction yet users might want to use it to further restrict entities.

Appendix G Protégé UI

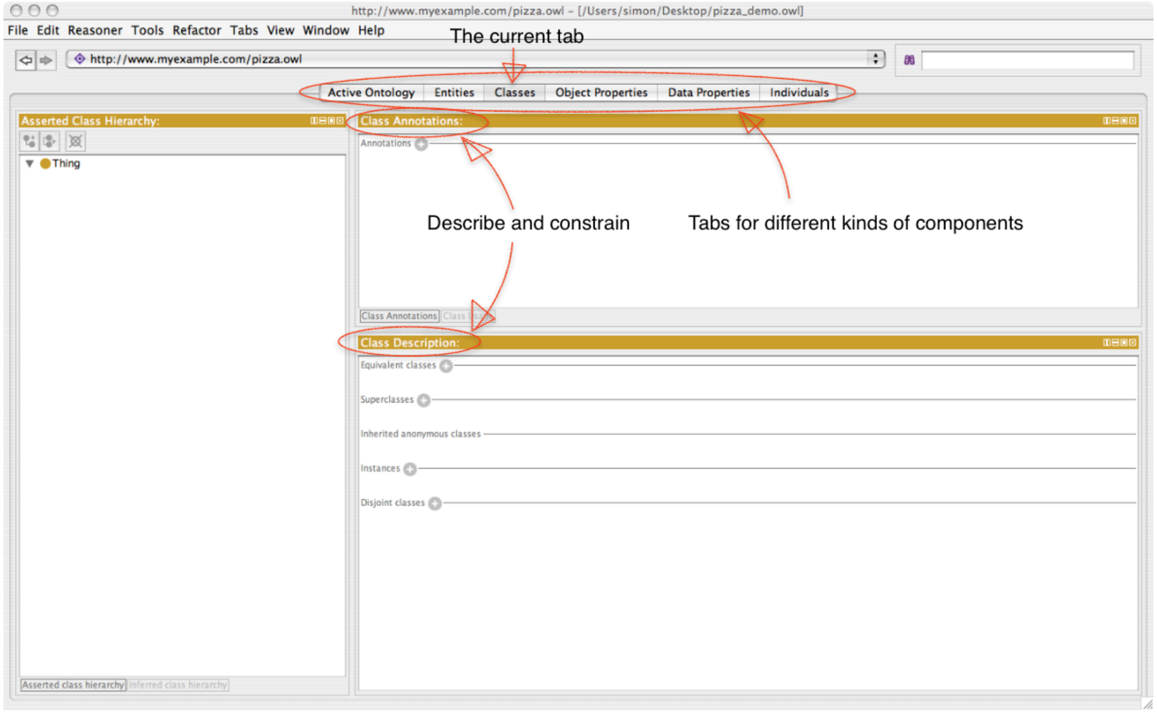


Figure 18 Protégé UI

Take the ‘Classes’ tab presented in Figure 18 as example, we go to this tab when we want to edit classes – the main building blocks of an ontology (Horridge et al., 2011). There are three yellow shading windows: asserted class hierarchy, class annotation, class description. Asserted class hierarchy present the hierarchy tree of classes; the class ‘Thing’ is the parent class of all classes and instances. Class annotation records all the annotations such as comment, versionInfo, seeAlso, isDefinedBy, priorVersion, etc. (Horridge et al., 2011). Class description provide all the relations between one class and other classes, properties and individuals: equivalent to, subclass of, general class axioms, instances, disjoint with, etc. In the class description window, we can define various relationships between classes, which is equivalent to defining the relationships of individuals belong to the classes. The situation in Figure 10 can be described in ‘subclass of’, ‘instances’ and ‘disjoint with’ section of the class description window. All the property related descriptions are defined in the ‘subclass of’ section where we use property restrictions (see 5.2.2 Restrict properties for more details) to describe how individuals are related through this property; it is called ‘subclass of’ because the defined class includes all the individuals that satisfy such description, the class we are describing is the ‘subclass of’ (or equivalent to) the defined class.

The ‘Object properties’, ‘Data properties’ and ‘Individuals’ tabs have similar layouts to the ‘Classes’ tab. The difference is that in the two properties tabs, there is an additional window for property characteristics; in the individuals tab, there is an additional window for property assertions, which shows the relation between individuals or the relation between individuals and data values.