

การวิเคราะห์ความสำคัญของจุดอ่อนโดยมาตรวัดของข้อมูลสาธารณะ



นางสาวรัศมีทิพย์ วิตา

ศูนย์วิทยพัทยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2553

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

RELEVANCY ANALYSIS OF VULNERABILITY USING METRICS BASED ON GLOBAL PUBLIC INFORMATION



Miss Ratsameetip Wita

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering


Chulalongkorn University

Academic Year 2010

Copyright of Chulalongkorn University


Thesis Title RELEVANCY ANALYSIS OF VULNERABILITY USING
METRICS BASED ON GLOBAL PUBLIC INFORMATION
By Miss Ratsameetip Wita
Field of Study Computer Engineering
Thesis Advisor Yunyong Teng-amnuay, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial
Fulfillment of the Requirements for the Doctoral Degree

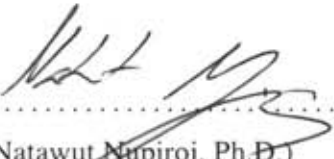

..... Dean of the Faculty of Engineering
(Associate Professor Boonsom Lerthirunwong, Dr.Eng.)


THESIS COMMITTEE


..... Chairman
(Professor Prabhas Chongstitvattana, Ph.D.)


..... Thesis Advisor
(Yunyong Teng-amnuay, Ph.D.)


..... Examiner
(Assistant Professor Kerk Piromsopa, Ph.D.)


..... Examiner
(Natawut Nupiroj, Ph.D.)


..... External Examiner
(Komain Pibulyarajana, Ph.D.)

รัศมีทิพย์ วิดา: การวิเคราะห์ความสำคัญของจุดอ่อนโดยมาตรวัดของข้อมูลสาธารณะ.
(RELEVANCY ANALYSIS OF VULNERABILITY USING METRICS BASED ON
GLOBAL PUBLIC INFORMATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : อ.ดร. ยรรยง
เต็งอำนาจ, 88 หน้า.

ระบบซอฟต์แวร์ในปัจจุบันมีความซับซ้อนที่ทำให้ไม่สามารถหลีกเลี่ยงการมีจุดอ่อนได้ ซึ่งจุดอ่อนดังกล่าวเป็นช่องทางที่สำคัญในการโจมตีและทำให้ระบบเกิดความเสียหายการแก้ไขอย่างตรงไปตรงมาคือการติดตั้งชุดปรับแต่งของซอฟต์แวร์อยู่เสมอแต่เนื่องจากความซับซ้อนของระบบและทรัพยากรที่จำกัดจึงเป็นปัญหาสำหรับผู้ดูแลระบบในการทำงานเพื่อเป็นการเพิ่มประสิทธิภาพของการทำงานและระดับความมั่นคงปลอดภัยของระบบการจัดลำดับความสำคัญของจุดอ่อนจึงเป็นสิ่งจำเป็น

งานวิจัยนี้ได้ทำการเสนอการวิเคราะห์ความสำคัญของจุดอ่อนโดยอาศัย มาตรวัด ข้อมูลสาธารณะมีการกำหนดความสำคัญของจุดอ่อนโดยศึกษาจากวัฏจักรชีวิตของจุดอ่อนและปัจจัยของการรวบรวมข้อมูลสาธารณะได้มีการเสนอกรอบการวัดความสำคัญของจุดอ่อนในเชิงปริมาณโดยอาศัยการทำเหมืองข้อมูลและออนโทโลยี

ในงานวิจัยนี้ได้มีการทำการทดลองกับกลุ่มข้อมูลตัวอย่างของจุดอ่อนที่มีความร้ายแรงจากหน่วยงานด้านความมั่นคงปลอดภัยพบว่ากลุ่มข้อมูลของจุดอ่อนที่มีความร้ายแรงดังกล่าวมีระดับความสำคัญที่แตกต่างกันขึ้นอยู่กับอายุและระดับข้อมูลการปรับปรุงจุดอ่อนและข้อมูลที่มีความสำคัญสูงมีความเกี่ยวข้องกับซอฟต์แวร์ในระดับผู้ใช้และไลบรารีกลางของซอฟต์แวร์

ภาควิชาวิศวกรรมคอมพิวเตอร์.....

สาขาวิชา.....วิศวกรรมคอมพิวเตอร์.....

ปีการศึกษา 2553

ลายมือชื่อนิสิต

Patsawatip W.

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

P. T. L.

จุฬาลงกรณ์มหาวิทยาลัย

4771863721: MAJOR COMPUTER ENGINEERING

KEYWORDS: SECURITY MANAGEMENT / VULNERABILITY / ONTOLOGY / SECURITY METRICS / WEB DATA MINING

RATSAMEETIP WITA : RELEVANCY ANALYSIS OF VULNERABILITY USING METRICS BASED ON GLOBAL PUBLIC INFORMATION. ADVISOR : YUNYONG TENG-AMNUAY, Ph.D., 88 pp.

Because of the complexity of software, vulnerability are unavoidable. Vulnerability is a common path for attacking the system. Straightforward protections are applying system patches. Due to the complexity of the software and limited resource in administrative work, system administrator has difficulties in applying remediation. To maximize work performance and security level of the system with limited administrative resource, vulnerability needs to be prioritized.

This work proposed the analysis of vulnerability relevancy using metrics based on global public information. Vulnerability relevancy is defined based on vulnerability life-cycle and public information acquisition factors. Vulnerability Relevancy Quantification Framework is proposed using web data mining concept with ontology.

The experiments are conducted on top of severe vulnerability announcements from established security organizations. The results show that top severe vulnerabilities from the published list have different level in relevancy depending on vulnerability age and availability of remediation process. The highly ranked vulnerability is related to client-side software and common software libraries.

Department: ... Computer Engineering ...

Student's Signature ... *Ratsameetip W.*

Field of Study: ... Computer Engineering ...

Advisor's Signature ... *Y. Teng-Amnuay*

Academic Year:2010.....

Acknowledgements

This dissertation would not have been possible without wonderful academically, financially and mentally supports. First of all, It is an honor for me to have got opportunities and financial support from the Higher Education Commission of Thailand, Graduate School, Chulalongkorn University, and National Institute of Informatics, Tokyo for research scholarships and internship.

I would like to show my gratitude to my advisor, Yunyong Teng-amnuay Ph.D, for all his thoughtful guidance and support all through six years from the beginning until now. Without his patient and effort, this dissertation would not have been possible all through difficulties.

I also would like to express my thankfulness to my dissertation committees: Professor Prabhas Chongstitvattana, Ph.D., Assistant Professor Kerk Piromsopa, Ph.D., Natawut Nupiroj, Ph.D., and Komain Pibulyarajana, Ph.D. for their useful comments and advices to help me enhance on my research strength and correct the weakness.

I would like to thank the Information System Engineering Laboratory, the Ph.D. seminar group of Computer Engineering Department, Chulalongkorn University and all colleagues especially Ms. Salinda Kuapongthai and Mr. Pakorn Techaveerapong for wonderful research facilities and environment and useful comments and supportive ideas.

I would like to thank to my very best friends, Assistant Professor Pizzanu Kanongchaiyos, Ph.D. and Ms. Tida Pichitlumken for inspiring me in both working and living.

Last but not least, A Million thanks to Mom, Dad, and my beloved family. Without their love, support and encouragement, I could not have accomplished my life target.

Contents

	Page
Abstract (Thai)	iv
Abstract (English)	v
Acknowledgements	vi
Contents	vii
List of Tables	x
List of Figures	xi
Chapter	
I Introduction	1
1.1 Problem Statement	3
1.2 Objectives of Study	3
1.3 Scopes of Study	3
1.4 Expected Contribution	4
1.5 Research Methodology	4
1.6 Publications	5
1.7 Organization	5
II Related Works	6
2.1 Vulnerability Related Information	6
2.1.1 Information Sources and Standards	6
2.1.2 Vulnerability Classification	8
2.1.3 Quantification Metrics	9
2.2 Knowledge Representation using Ontology	10
2.2.1 Ontology Markup Languages	11
2.2.2 Ontology Evaluation	12
2.2.3 Ontology in Information Security	13
2.3 Managing Global Public Information	14
2.3.1 Information Retrieval	14
2.3.2 Web Mining	15
2.3.3 Expectation-Maximization Algorithm	15
2.4 Summary	16
III Vulnerability Relevancy Ranking Framework	18

Chapter	Page
3.1 Definition of Vulnerability Relevancy	18
3.1.1 Lifecycle Semantic	19
3.2 Vulnerability Relevancy in Online Document	20
3.2.1 Context Types	20
3.2.2 Information Source	21
3.2.3 Hits in Public Interest	21
3.2.4 Information Aging	21
3.3 The Framework	22
3.3.1 Knowledge Management	22
3.3.2 Lifecycle Analysis	23
3.3.3 Relevancy Quantification	23
3.4 Summary	23
IV Vulnerability Lifecycle Ontology	24
4.1 Information Source	24
4.2 Knowledge Representation	25
4.3 Creation of VLO	25
4.4 Example Usage in Document Extraction	29
4.5 Evaluation	32
4.6 Summary	33
V Ontology based Context Sensitive Profile	34
5.1 Introduction	34
5.2 Subcontext in Ontology	35
5.3 Retrieving and Preprocessing of Public Information	36
5.4 Context Sensitive Profile	37
5.4.1 Keyword Matching	37
5.4.1.1 Thesaurus Matching	38
5.4.1.2 Concept Matching	38
5.4.2 Context Matching	39
5.4.2.1 Context Richness	39
5.4.2.2 Context Availability	40

Chapter	Page
5.5 Experiments	41
5.5.1 Experiment 1 - Context Richness Evaluation	41
5.5.2 Experiment 2 - Context Availability Evaluation	41
5.6 Summary	44
VI Vulnerability Relevancy Quantification	46
6.1 Hits in Public Interest	46
6.2 Information Source	47
6.3 Information Aging	48
6.4 Subcontext Availability	48
6.4.1 Context Relevancy	50
6.5 Vulnerability Relevancy Quantification Model	52
6.5.1 Example Calculation of VRscore	55
6.5.2 Relevancy Quantification Service	58
6.6 VRscore Evaluation	58
6.7 Summary	60
VII Research Results	63
7.1 Information source	63
7.2 Hits in Public Interest	64
7.3 VRscore and Risk Rank	65
VIII Conclusions	67
8.1 Discussion and Suggestion	68
References	70
Appendix	79
Biography	88

List of Tables

Table	Page
2.1 An Example of the Content of Each CVE Item Provided by MITRE	7
4.1 Lifecycle States and Related Public Information	28
4.2 Comparison Results between Three Ontologies	33
5.1 Training Dataset for Context Richness Evaluation	41
5.2 Extraction Method of DV, CMM and CSP	42
5.3 Training Dataset for Context Availability Evaluation	43
5.4 Clustering Mode in the Experiment	43
6.1 Context-Based Relevancy Metric.	49
6.2 Clustering Result for Vulnerability Relevancy Quantification Model.	52
6.3 Vulnerability Relevancy Level from Clustering Result.	52
6.4 $R_{context}$ Value Range	54
6.5 Comparison of Relevancy Attributes from Example.	57
6.6 Comparison of CVE-2005-0344 and CVE-2007-0038.	57
6.7 Test Dataset	59
7.1 VRscores and Attributes of Sample CVE	65
7.2 Sample of Risk Ranked Vulnerability based on Severity and Relevancy	66
8.1 Risk Level of Vulnerability	68
A.1 Ranked Vulnerability Relevancy Scores	79

List of Figures

Figure	Page
2.1 Representation Dimension of Ontology	11
2.2 Example of Clustering of mixture of Gaussians	16
3.1 Public Information and Lifecycle State Relationship.	22
3.2 Vulnerability Relevancy Ranking Framework.	23
4.1 VLO Knowledge Building Process.	26
4.2 Base Concepts and Relationships of VLO.	27
4.3 OWL Definition of “Remediation” Concept in VLO	27
4.4 Thesaurus of “Tools” Concept in VLO.	29
4.5 The Structure of VLO	30
4.6 Extracted Keywords from CVE Website Describing CVE-2007-0217.	31
4.7 Extracted Keywords from iDefense Describing CVE-2007-0217.	32
5.1 Processes and Intermediate Results	34
5.2 Ontology with Subcontext Structure	35
5.3 VLO and Subcontexts of Lifecycle States	36
5.4 Example of Relevancy Evaluation for Vulnerability A	37
5.5 Evaluation Result of Context Richness	42
5.6 Evaluation Result of Context Availability	44
5.7 Context Sensitive Profile.	45
6.1 Search Result from Google Search Service	47
6.2 Vulnerability Relevancy Quantification Model.	50
6.3 Clustering Result in (a) Basic Information, (b) Technical Detail, (c) Exploit Detail, (d) Publicity, and (e) Remediation.	51
6.4 Lifecycle Context Distribution in (a) Basic Information, (b) Technical Detail, (c) Exploit Detail, (d) Publicity, and (e) Remediation	53
6.5 Variation of Vulnerability Relevancy Score from Different Attributes Used.	61
6.6 Relevancy Quantification Service.	62
6.7 Comparison of Cumulative Probability of VRScore in Test Dataset	62
7.1 Distribution of Top 20 Information Source from Search Result.	63
7.2 Cumulative Probability Distribution of Hits in Public Interest in Training Dataset and Test Tataset	64

CHAPTER I

INTRODUCTION

Software is becoming more vulnerable these days due to the increasing vulnerabilities and exposures (Frei et al., 2006; Wu and Yip, 2005). Vulnerable software and configuration are the most common weakpoints for break-ins as reported in annual SANS Top-20 Security Risks (SANS, 2007). The obvious protection is to patch or fix those vulnerabilities as soon as possible. Patching vulnerability may include an updated version of software from vendor or reconfiguration of system parameters. The process of securing the system may be slow and may take many weeks before half of the systems are patched (Qualys, 2006). When vulnerabilities are left unpatched, viruses, worm, and other types of attacks are able to exploit those vulnerabilities and be harmful to the system (Ko and Lee, 2007).

From the vulnerability life cycle, progressing from discover to correction stage needs time for vendor to analyze, workaround, and create system change, while details of vulnerability are at the same time spread among skilled hackers. Lee and Davis profiled corrective actions from various OS vendors showing solution for vulnerability (Lee and Davis, 2003). There are at least sixty days to cover all vulnerability. This delay inevitably makes system protection one step behind energetic attackers. Eventually the system patch is released. Still, there are some problems in applying patches to the system (Arbaugh, 2004). Many users decline to upgrade their system just because they are afraid of new vulnerability which may affect some function of the system, or performance, for example, Windows XP service pack 2.

Another problem is massive administrative workload (Longstaff, 2003). With 10,000 vulnerabilities reported in 2004, if it takes 10 minutes to understand a particular vulnerability description, this will take approximately $10,000 \text{ vulnerabilities} * 10 \text{ minutes to read each} = 167 \text{ days}$.

If a particular system is affected by 10% of vulnerabilities reported, and it takes around half an hour to apply each patch, this results in workload for the administrator of approximately $1,000 \text{ vulnerabilities} * 30 \text{ minutes} = 50 \text{ days}$.

Total workload for an administrator in keeping system up-to-date is $167 + 50$ or 217 days on the average. This enormous workload is only about patching system holes, not including security configuration and monitoring. Thus, prioritization is needed.

According to the risk management principle, $risk = impact \times likelihood$ (Jaquith, 2007). A vulnerability is considered relevant if it brings about a significant impact with a high likelihood of attack. Many researchers attempted to develop a measurement of both impact and likelihood in order to quantify risk. Risk quantification schemes have been studied and defined as metrics for risk, or relevancy, of individual vulnerability. Many quantitative models have been developed that rely on vulnerability characteristics and the effect of losses. The major problem is the rapid growth of the number of vulnerabilities while the information in the model is manually and statically captured. Manual vulnerability analysis hinders risk management and can incorrectly rank some types of vulnerability. Risk quantification scheme based on attacker behavior (Dantu et al., 2004; Jha and Wing, 2001), severity of damage (Wita and Teng-Amnuay, 2005), probability of being exploited (Jumratjaroenvanit and Teng-amnuay, 2008), common vulnerability scoring system (NIST, 2007) and other schemes. Microsoft corp. (Microsoft, 2002) have been studied and defined as metrics used to evaluate risk, or relevancy, of individual vulnerability. These are static scoring schemes without the use of the age of vulnerability. Even CVSS (NIST, 2007) which publishes temporal score based on exploitability and remediation, obtaining updated information for those metrics over time needs much concentration from system administrators who are usually overworked.

One possibility in identifying relevancy of vulnerability over time is observing its life cycle. From vulnerability life cycle analysis (Arbaugh et al., 2000; Browne et al., 2001; Frei et al., 2006), events involving vulnerability and exploitation cycle have been identified. Browne et al. (Browne et al., 2001) identified that a vulnerability will die when there are no more instances of the flaw that can be exploited. They also defined that a vulnerability death will occur when either all instances of the vulnerable code have been patched or when they have been retired or replaced by a version of software that does not contain the flaw in question. Empirical result from (Arora et al., 2006) also illustrated life cycle of vulnerability. Moreover, from their finding, numbers of attack incidents tend to gradually increase right after vulnerability fix is released before decreasing. Their another study in (Arora et al., 2004) also emphasized the fact that information on patches benefits attacker as well. Result in (Qualys, 2006) depicted relationship between major

vulnerability incidents and life cycle of vulnerability. In these previous works the temporal of vulnerability was neither employed or nor consistent on relevancy, thus a possible approach in evaluating relevancy is to observe and to analyze from public information related to a particular vulnerability. Observing behavior through public information has increased significantly since web content become widespread (Cooley et al., 1997; Liu, 2007). Public information analysis, or, web data mining, is used in evaluating web usage patterns, page ranking (Adafre et al., 2006) and sentiment opinion analysis in discussion community (Mishne, 2006; Jindal and Liu, 2008). In this research, relevancy attributes and context sensitive profile were proposed as relevancy metric by using an ontology-based data mining on public information analysis.

1.1 Problem Statement

Due to exponentially increase of vulnerabilities, system administrator has difficulties in applying remediation. To maximize work performance and security level of the system with limited administrative resource, vulnerability needs to be prioritized. This research aims to define a quantitative measurement in evaluating relevancy of vulnerability based on the analysis of public information available globally.

1.2 Objectives of Study

The objectives of study are as follows:

- Study the relationship between public information on vulnerability available globally and its relevancy in terms of security management,
- Define relevancy attributes for vulnerability based on public information obtained from web data mining, and
- Define quantitative measurement, or scoring, for prioritizing vulnerability based on relevancy

1.3 Scopes of Study

The scopes of this study are as follows:

- Create concept ontology for describing vulnerability lifecycle based on character-

istic of vulnerability listed in CVE database,

- Limits webpages used in this research will be limited to search result only from Google search service, and
- Limit Initial information of vulnerability used in this research to a selection of CVE entries from the updated version in May 2008 with 32464 CVE entries maintained by Mitre Corporation.

1.4 Expected Contribution

This work will make the following contributions:

- Vulnerability prioritizing methodology based on public awareness and attention.
- Tools allowing semi-automated evaluation of vulnerability relevancy for administrators to prioritize their remediation.
- Concept ontology for vulnerability lifecycle.
- Understanding of the effect of public information on relevancy and risk analysis.

1.5 Research Methodology

This research employs the following methodology:

- Define the relationship between vulnerability relevancy and lifecycle states.
- Construct Vulnerability Lifecycle Ontology (VLO) from security knowledge base and CVE description.
- Refine content and information source classification by using suitable clustering technique.
- Refine the metrics using human expertise experience.
- Develop an automate data capturing module using API.
- Experiment with larger amount of dataset and refine the scoring mechanism.

- Evaluate the result from the relevancy metrics and scoring scheme.
- Conclude the result and prepare dissertation

1.6 Publications

Parts of this dissertation have been published in academic conferences and journal as follows:

- “Ontology for Vulnerability Lifecycle” by Ratsameetip Wita, Nattanatch Jiampanon, and Yunyong Teng-amnuay in the IEEE International Symposium on Intelligent Information Technology and Security Informatics 2010 (IITSI 2010), Jingtangshan, China, April 2010.
- “Ontology-Based Document Profile for Vulnerability Relevancy Analysis” by Ratsameetip Wita and Yunyong Teng-amnuay in the proceeding of 10th WSEAS International Conference on Applied Computer Science (ACS’10), Iwate, Japan, October 2010.
- “Context Sensitive Profile for Quantification of Vulnerability Relevancy” by Ratsameetip Wita, and Yunyong Teng-amnuay in IEICE Transaction of Information and System , 2011 (Under Review).

1.7 Organization

The remainder of this dissertation is structured as follows:

General background in vulnerability information and quantification, security related ontology construction and usage are described in Chapter 2. **Vulnerability Relevancy Ranking Framework** is defined in Chapter 3. In Chapter 4, **Vulnerability Lifecycle Ontology** construction and evaluation are presented to be used as vulnerability knowledge base for determining vulnerability content in webpages. Chapter 5 describes the process for creating **Context Sensitive Profile** from public information. Chapter 6 introduced the Vulnerability **Relevancy Quantification Model** including context sensitive document profile, an ontology web data mining, and its evaluation. Chapter 7 presents the analysis of relevancy attributes and the research results. Chapter 8 concludes this dissertation and describes future extension possible this work.

CHAPTER II

RELATED WORKS

Related literatures and research works are listed in this Chapter. The related works includes vulnerability information and classification, other vulnerability quantification methodology, the using of ontology as a knowledge representation and how the ontology has been used in security area, and web data mining.

2.1 Vulnerability Related Information

2.1.1 Information Sources and Standards

Many of system and software flaws are discovered and reported everyday from communities such as: system administrator, software vendor, security advisory or even from hacker. Different names have been used to identify the same flaw or vulnerability. In order to globally identify the flaw or vulnerability, a standard name and description of vulnerability itself and the related information are listed as follow:

Common Vulnerability and Exposures (CVE) (Mitre, 1999) is a standard naming system for identifying vulnerabilities and other exposures, as agreed upon by various security organizations. CVE identifiers (also called “CVE names,” “CVE numbers,” “CVE-IDs,” and “CVEs”) are unique and as used as common identifiers for publicly known information security vulnerabilities. CVE identifiers have “entry” or “candidate” status. Entry status indicates that the CVE Identifier has been accepted as a vulnerability to the CVE List while candidate status indicates that the identifier is under review for inclusion in the list. The process of review cve entry is manually done by cve committee. Table 2.1 shows an example of cve entry. In Table 2.1 CVE-1999-0002 is defined to a vulnerability with the given description and reference sited as listed. Description is a brief explanation about vulnerability and reference site list related security advisory.

Common Weakness Enumeration (CWE) (Mitre, 2007a) is a unified, measurable set of software weaknesses description for better understanding and management related to architecture and design of software. They create mappings between CWEs and CVE names so that each CWE group or element has a list of the specific CVE names that

Table 2.1: An Example of the Content of Each CVE Item Provided by MITRE

Field name	Content
CVE standard name	CVE-1999-0002
Description	Buffer overflow in NFS mountd gives root access to remote attackers, mostly in Linux systems.
References	SGI:19981006-01-I CERT:CA-98.12.mountd CIAC:J-006 BID:121 XF:linux-mountd-bo

belong to that particular CWE category of software security weaknesses.

CWE goals are to build multiple different views within CWE, for supporting multiple audiences, to improve the existing views so that their organization is more consistent, and to change the names and descriptions for more precise information of each CWE entry

The structure of CWE is built on well known taxonomies such as Seven Pernicious Kingdoms (7PK), the categories of errors in (CLASP), the Genesis and Location classifications used by Landwehr, and the Preliminary List of Vulnerability Examples for Researchers (PLOVER). As a result, the Development view can be readily understood by users who are already familiar with these other taxonomies. Two main organizational views of CWE are:

- Development Concepts (CWE-699) is geared towards developers and people who are familiar with other vulnerability-related taxonomies.
- Research Concepts (CWE-1000) is oriented towards academic research, creating a new framework for classifying weaknesses.

National Vulnerability Database (NVD) (NIST, 1999) is maintained by National Institute of Standards and Technology, is the U.S. government. NVD is a standards repository of vulnerability management data. NVD includes databases of security checklists, security related software flaws, misconfigurations, product names, and impact metrics.

2.1.2 Vulnerability Classification

Several flaw and intrusion classification schemes have been proposed. Landwehr, et. al. attempted to organize information on security flaws for software development (Landwehr, 1981). When new flaws are added, readers will gain a fuller understanding of which parts of the system and which parts of the system's life cycle are generating more security flaws than others. In Landwehr's classification scheme, they categorized flaws according to 3 criteria: genesis, time of introduction, and location.

Jiwnani and Zelkowitz proposed software testing strategy based on a classification of vulnerabilities to develop secure and stable systems (Jiwnani and Zelkowitz, 2002). They have defined the taxonomy scheme based on Landwehr's and evaluated it using a database of 1360 operating system vulnerabilities from Harris Corporation and Red Hat Linux Errata.

Hogan categorized security flaws in UNIX stand-alone and distributed system (Hogan, 1988) following by Saltzer and Schroeder's principles for protection (Saltzer and Schroeder, 1975). This classification is chiefly concerned with why the flaws are present in the system.

The classification stated above mainly focus on the result of the exploitation. The other approach in classification is considered the technique used to exploit. Neumann and Parker categorized computer misuse techniques into nine classes (Neumann and Parker, 1989) by collecting data from 3000 computer abuse cases.

Ranum groups attacks into eight intuitive categories based on techniques used by attacker (Ranum, 1996): social engineering, impersonation, exploits transitive trust, data driven, infrastructure, denial of service, and magic (unseen attack technique).

Wita and Teng-amnuay presented the profiling scheme of vulnerability severity based on CVE information for system administrative purpose (Wita and Teng-Amnuay, 2005). The severities of exploitation are classified into 4 types: confidentiality violation, integrity violation, availability violation, and system compromised.

Dantu et al. proposed a classification of attributes in risk management based on hypothesize that sequence of network actions by attackers depends on their social and

attack profile (Dantu et al., 2004). They surveyed individual attackers for their ability and attack intent to model attack behavior. They did their experiment by conducting a survey of 32 questions. The answers of those questions are used to infer the behavior of the survey participant. Scores are assigned for the questions' options. Sum of selected option is used to classify participant into one of three profiles: hacker-behavior, opportunist-behavior, and explorer-behavior based on skill, time and attitude.

Lai and Hsia proposed network security improvement method which composed of network management, vulnerability scanning, risk assessment, and access control (Lai and Hsia, 2007). In their work, vulnerability information is used to evaluate risk level of networked systems. By ranking the most threaten service ports, ACL can be created to set access restriction of those threaten ports, so that the system can be more secured.

2.1.3 Quantification Metrics

Tuper and Zincir-Heywood proposed VER-bility Security Metric to measure desirability of different network configuration (Tupper and Zincir-Heywood, 2008). VER-bility score is a number value returned from a function of three dimensions: vulnerability, exploitability, and attackability. VER-bility metric used data from three sources: network topology, attack graph, and scores assigned from CVSS. They did show the experimental result that different network connectivity restriction resulted in different secure level represented by VER-bility metric.

Common Vulnerability Scoring System (CVSS) (NIST, 2007) aims to define and communicate the fundamental characteristics of vulnerability and also to provide contextual information that more accurately reflects the risk to one's own unique environment. This allows system administrators to make more informed decisions when trying to mitigate risks posed by the vulnerabilities. CVSS is composed of three metric groups: Base, Temporal, and Environmental. Temporal and environmental scores are marked as optional.

Wang et.al. proposed temporal metrics for software vulnerabilities based on the Common Vulnerability Scoring System 2.0. (Wang et al., 2008) A mathematical model was provided to calculate the severity and risk of a vulnerability, which is time dependent including exploitability, remediation level, and report confidence attributes of an information asset in a computing environment.

HyunChul et. al proposed a framework for software risk evaluation with respect to the vulnerability lifecycle. Vulnerability lifecycle as a stochastic process (Hyunchul and K.M., 2010). CVSS metrics were used to evaluate the impact of the breach. The model used Frei's model (Frei et al., 2006) to identify transition rates with the related distributions and can lead to simplified as well as detailed modeling methods.

Microsoft Corp. proposed the process of risk management, DREAD, for identifying and rating threats based on a architecture and implementation application in the system (Meier et al., 2003). Architectural based threat modeling activity steps are defined. Threat rating are defined by considering 6 attributes as: Damage potential: How great is the damage if the vulnerability is exploited? Reproducibility: How easy is it to reproduce the attack? Exploitability: How easy is it to launch an attack? Affected users: As a rough percentage, how many users are affected? Discoverability: How easy is it to find the vulnerability?

2.2 Knowledge Representation using Ontology

In this research, we aim to create a base of concept of vulnerability lifecycle and define significant relationship between lifecycle state and vulnerability information published in each states. Ontology matches our needs in identifying lifecycle states concepts and their relationships.

An ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to define the domain. Ontologies are used in artificial intelligence, the Semantic Web, software engineering, biomedical informatics, library science, and information architecture as a form of knowledge representation about the world or some part of it. Common components of ontologies include: Individuals, Classes, Attributes, and Relationships. Ontology can be divided into two different types.

Upper Ontology represents semantic relationship between very general concepts across allknowledge domains. Upper Ontology support semantic interoperabilitybetween languages or domain. The example of well-known upper ontology are WordNet, BabelNet, Babilon WordNet.

Domain Ontology represents the pragmatic or the specific meaning/ relationship

between concepts of domain-specific concepts and relationships such as Public Health, Security, Industrial, etc.

Figure 2.1 shows the dimension of upper ontology and domain ontology. Upper Ontology express the content and its semantic relationship while Domain Ontology represent context sensitive meaning of the concepts in specific domains.

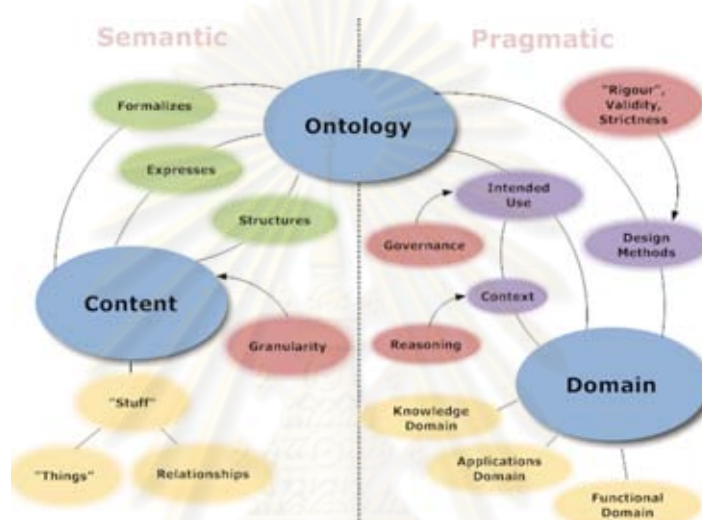


Figure 2.1: Representation Dimension of Ontology

In this research, lifecycle and its relationship to online document is organized for our automated inference classification. Web Ontology Language (OWL) is used to represent those concepts and relationships while Simple Knowledge Organization System (SKOS) is used to represent thesaurus of concepts.

2.2.1 Ontology Markup Languages

Web Ontology Language (OWL) is a recommendation from W3C for publishing and sharing ontology (Bechhofer and et al., 2004). OWL-DL (Description Logic) is one of OWL sub-language capable of ontology automated reasoning. This will facilitate our automated inference classification.

Simple Knowledge Organization System (SKOS) defines specification and standard to support the use of knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web (Miles, 2009).

SKOS is a data model which represents the logical characteristics of ontology concepts and relationships. SKOS data are expressed as RDF triples, and can be encoded in any concrete RDF syntax. SKOS itself is not a formal knowledge representation language. It is used as annotation vocabulary for OWL ontology.

We used "Formal/Semi-Formal Hybrids" pattern from (W3C, 2004) to construct the hierarchy of vulnerability related concepts and relationships by OWL structure and model the vocabulary of vulnerability concepts such as *preflabel* and *altlabel* to represent semantic-like vocabulary in SKOS data model.

2.2.2 Ontology Evaluation

Ontologies have been used to improve document classification and information extraction. Hotho et al., for example, used ontology in text preprocessing for K-Mean clustering (Hotho et al., 2001). The selection and aggregation of concepts improve the clustering results compared to the traditional strategy.

Deng and Peng presented the Concept Vector Model for document categorization (Deng and Peng, 2006). Terms in documents were extracted by a concept matching process in order to create the concept feature of the document.

D'Amato et al. proposed the extension of the k-nearest neighbor for OWL ontology (dAmato et al., 2008). Behavior similarity and dissimilarity measurement between concepts and keywords were used in the classifier.

Alani and Brewster presented AKTiveRank, an ontology ranking prototype based on structure analysis (Brewster et al., 2004). They introduced the Class Match Measure (CMM) to measure the coverage of ontology for the search term and the Betweenness Measure (BEM) to identify the central of ontology. Their work facilitated in the ranking and choosing of an appropriate ontology for a specific domain.

These works used ontology to classify totally different domains with different sets of concepts, our work, however, targets the same domain with fine-grained subcontexts.

2.2.3 Ontology in Information Security

Raskin, et. al. (Raskin et al., 2001) proposed a new, content-oriented, knowledge- and meaning based approach to form the basis of the NIP component of the information security research paradigm. The cumulative knowledge of the information security community about the classification of threats, their prevention and about defense against computer attacks should be formalized, and this knowledge are brought to bear in developing an industry-wide, constantly upgradeable manual for computer security personnel that may involve a number of delivery vehicles, including an online question-answer environment and a knowledge-based decision support system with dynamic replanning capabilities for use by computer security personnel.

Kim, et. al. developed the NRL Security Ontology (Kim et al., 2005) to provide the ability to annotate security related information in various levels of detail for commercial and military uses. They created the ontology to facilitate mapping of higher-level (mission-level) security requirements to lower-level (resource-level) capabilities. Sevens ontologies are combined to describe relationship in security as follow: Security Main Ontology, Credentials Ontology, Security Algorithms Ontology, Security Assurance Ontology, Service Security Ontology, Agent Security Ontology, and Information Object Ontology.

He, et. al. (He et al., 2004) proposed a cooperating detection framework among multi-sensor IDS based on ontology. They designed an ontology after analyzing some IDSs rules and the security vulnerabilities published by Common Vulnerabilities and Exposures (CVE). The complete ontology includes two kinds of nodes: value nodes and attribute nodes. Attribute nodes describe all the features that can be observed by multisensory and value nodes are the children of some attribute nodes which represent. By assigning the weight to the edge between values nodes and their parent attributed node, they provided a more flexible matchmaking method for intrusion detection.

Pinkston, et al. (Pinkston et al., 2003) proposed their model as a target-centric ontology that is to be refined and expanded over time by arguing that any taxonomic characteristics used to define a computer attack are limited in scope to those features that are observable and measurable at the target of the attack. They have produced an ontology specifying a model of computer attacks based upon an analysis of over 4,000 classes of

computer intrusions and their corresponding attack strategies and is categorized according to system component targeted, means of attack, consequence of attack, and location of attacker. They used DAML+OIL and have prototyped it using DAMLJessKB.

Moreira and his colleagues developed the security related ontology, ONTOVUL and ONTOSEC, to describe the relationship between vulnerability and security incident for security management in the organization (Moreira et al., 2008).

2.3 Managing Global Public Information

In analyzing public information related to vulnerability, many techniques will be used in gathering web information and analyzing difference or similarity of different structure documents. Web data mining (Liu, 2007), citeShim99 is a methodology in information analysis for the Internet. Web data mining discovers useful information or knowledge from the Web hyperlink structure, page content, and usage data. Although web mining uses many data mining techniques, as mentioned above it is not purely an application of traditional data mining due to the heterogeneity and semi-structured or unstructured nature of the web data. Many new mining tasks and algorithms were invented in the past decade. Web mining tasks can be categorized into three types: web structure mining, web content mining and web usage mining.

2.3.1 Information Retrieval

To evaluate vulnerability relevancy based on public information, related information need to be retrieved from the Internet with specific keywords. In this research, information retrieval technique will be studied and applied in the phase of public information gathering. Information retrieval (IR) (Grossman and Frieder, 2004), (Liu, 2007) is the study of finding information that matches needs. Technically, IR studies the acquisition, organization, storage, retrieval, and distribution of information. Historically, IR is about document retrieval, emphasizing document as the basic unit.

An IR model (Cooley et al., 1997) governs how a document and a query are represented and how the relevance of a document to a user query is defined. There are four main IR models: Boolean model, vector space model, language model and probabilistic model. Electronic document must be assigned, or classified, to one or more categories based on its contents. Document classification can be divided into three groups (Liu, 2007): super-

vised document classification where some external mechanism (such as human feedback) provides information on the correct classification for documents, unsupervised document classification where the classification must be done entirely without reference to external information, and semi-supervised document classification where parts of the documents are labeled by the external mechanism.

2.3.2 Web Mining

Web mining is the usage of data mining techniques to discover patterns from the Web. Web mining can be divided into three different classes: Web usage mining, Web content mining and Web structure mining.

Web usage mining is the process of extracting useful information from server logs, such as access statistic. Web usage mining is used to find out what users preference in using services which may use for select suitable information. For example, people who usually watch score report of soccer game might interested in buying soccer team souvenirs.

Web content mining, so called web text mining, is the process of mining content in webpages. The technologies that are normally used in web content mining are NLP (Natural language processing) and IR (Information retrieval). Although data mining is a relatively new term, the technology is not. But the challenge in web content mining is how to capture only content from a fancy webpages.

Web structure mining is the process of using graph theory to analyze the node and connection structure of a web site. According to the type of web structural data, web structure mining can be divided into two types: 1. Extracting patterns from hyperlinks in the web: a hyperlink is a structural component that connects the web page to a different location. 2. Mining the document structure: analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

2.3.3 Expectation-Maximization Algorithm

The algorithm which is used in practice to find the mixture of Gaussians that can model the data set is called Expectation-Maximization(EM) (Dempster et al., 1977).

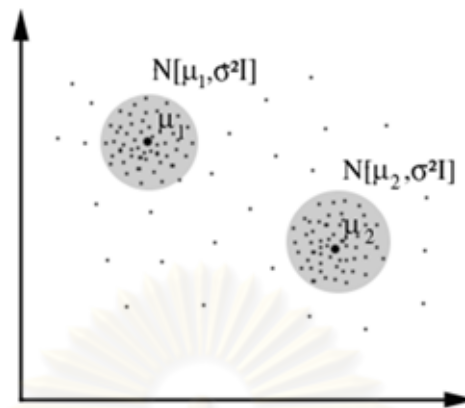


Figure 2.2: Example of Clustering of mixture of Gaussians

Mixture of Gaussians The most widely used clustering method of this kind is the one based on learning a mixture of Gaussians: we can actually consider clusters as Gaussian distributions centered on their centroid. From Figure 2.2, the grey circle represents the first variance of the distribution.

It chooses the Gaussians at the random probability $P(\omega_i)$, a sample point $N(\mu_i, \sigma^2 I)$

Suppose we have $P(\omega_1), \dots, P(\omega_k), \sigma$ as mixture of Gaussians and x_1, x_2, \dots, x_N as sample points.

We can obtain the likelihood of the sample as $P(x|\omega_i, \mu_1, \mu_2, \dots, \mu_k)$ (probability of a datum given the centers of the Gaussians).

The likelihood function will be

$$P(\text{data}|\mu_i) = \prod_{i=1}^N \sum_i P(\omega_i) P(x|\omega_i, \mu_1, \mu_2, \dots, \mu_k)$$

2.4 Summary

In this Chapter, we summarize general literature in vulnerability information, including standard naming, classification and quantification and vulnerability repository. Vulnerability lifecycle concept and the analysis of vulnerability economic is introducing. The usage of Ontology as a knowledge representation in information security and the

possibility of using ontology in data mining are also listed. In the next Chapter, we will introduce the vulnerability relevancy ranking framework based on global public information.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER III

VULNERABILITY RELEVANCY RANKING FRAMEWORK

3.1 Definition of Vulnerability Relevancy

This research defines quantitative measurement for prioritizing vulnerability based on vulnerability relevancy. Many researchers have been developing quantification measurement in prioritizing vulnerability. Dantu, et. al.(Dantu et al., 2004) proposed their measurement based on survey of intention and skill of attackers. Vulnerability relevancy depends on which group of attackers is likely to attack the system. Lai and Hsia (Lai and Hsia, 2007) defined vulnerability relevancy based on configuration of the system, for example: Is an important service port vulnerable?, and CVSS base score (NIST, 2007). Tuper and Zincir-Heywood (Tupper and Zincir-Heywood, 2008) proposed VER-bility to evaluate security level of different configuration based on the number of vulnerabilities and how hard they can be reached from network. Wita and Teng-amnuay (Wita and Teng-Amnuay, 2005) proposed a metric based on severity of loss after the vulnerability is exploited and Jumratjaroenvanit and Teng-amnuay (Jumratjaroenvanit and Teng-amnuay, 2008) defined different types of vulnerability maturity model based on analysis of life cycle.

These related works tried to prioritize vulnerability based on various aspects and approaches. Most of them employed static, readily available information in defining relevancy. Only CVSS (NIST, 2007) and POA (Jumratjaroenvanit and Teng-amnuay, 2008) which define an attribute based on phases in vulnerability life cycle using information gleaned from the public domain. However, none of these metrics provided relevance information about vulnerability obsolescence. In this research, we define vulnerability relevancy based on level of public awareness and the maturity of lifecycle of particular vulnerability. Public awareness can come from different types of information and sources such as security advisory, incident report from user, advertisement, news, etc.

3.1.1 Lifecycle Semantic

Our work defines the relationship between states of vulnerability lifecycle and related information gleaned from webpages. In this section, vulnerability lifecycle and the definition of its states are introduced.

Lifecycle Definition Vulnerability lifecycle has been defined differently in various researches. In this work, states in the lifecycle are based on (Frei et al., 2006) and are described as follows.

- **Discovery.** Vulnerability is discovered by vendor, security agent, or even hacker. This state can be before or after the software is released. The vulnerability is not yet widely known to the public.
- **Disclosure.** After a vulnerability is discovered, related information is available only among security teams or certain parties. Basic information released in this state contains a description about symptom and its cause. The vulnerability is discussed on mailing lists, security websites and underwent analysis by trusted channel. Any vulnerability reported to CVE website is also considered as being in this state.
- **Exploit.** A vulnerability in this state is described by the availability on the Internet of a sequence of commands or codes intended for exploitation. Availability of an automated exploit tools such as worm and virus or reports or news about the availability reflects exploit state of a particular vulnerability.
- **Publicity.** Vulnerability is in the publicity state when it is widely known. Full technical information, consequences and incident findings are available at large on the Internet. Vulnerability which develops in to this state widely impacts the world. Warning or alert are officially announces by vendors, governments, and news agencies.
- **Remediation.** This state of the lifecycle is slightly different from what is defined in (Frei et al., 2006). This is defined as any possible solution available from vendor and security agency in order to disable the exploit of the vulnerability. Remediation considered in this state includes software patch released from certified vendor, instruction, certain configuration change, security fix, or other security software effort in detecting and preventing exploitation such as IDS and anti-virus signature.

3.2 Vulnerability Relevancy in Online Document

3.2.1 Context Types

Information from various sources signifies different aspects of vulnerability. Different context types are represented various states of lifecycle. They are classified as follows.

- **Basic information:** This is defined as simple or easy-to-obtained information of a particular vulnerability. Basic information describes problem of a specific software or platform and provides information about consequence and severity of exploitation. It can be found in discovery state and may not be publicly available. It is further revealed in the early phase of disclosure state.
- **Technical detail:** This is information on precondition and postcondition in exploiting a particular vulnerability. Specific port number, vulnerable code section, and attack technique are discussed. It usually contains basic information plus more specific information in exploiting vulnerability. It can appear in mailing lists, security webboards, advisory pages, etc. Technical detail is revealed in disclosure state.
- **Exploit detail:** This is the availability of command sequence or source code that can facilitate the exploit of vulnerability and is available to the public. The availability of this corresponds to exploit state. News on availability also signifies that the lifecycle has entered this state.
- **Incident alert:** This is the report on widespread problems based on a particular vulnerability. Incident alert contains information about real exploitation incident which include damages, impact to the public, and also statistical report of affected systems. In this work, we consider incident alert from reliable information sources, such as government agency websites, news agencies, or system vendors. The availability of incident alert refers to the publicity state in the lifecycle.
- **Remediation detail:** This refers to any solution or workaround published and usually can be directly retrieved from system vendor. Remediation can also be available through security software such as intrusion detection rules or anti-virus signature updates. It refers to remediation state in the lifecycle.

3.2.2 Information Source

Vulnerability information comes from many different sources including announcements by software vendors and government agencies, news websites, technical discussion boards, and feedback forums hosted by software vendors. Different sources provide different types of information. For example, software vendors publish vulnerability information and remediation pertinent to their products while news websites are more concerned in outbreaks. Technical discussions board may contain in-depth information about exploitations, symptoms or workaround remediations. Kannan and Telang (Kannan and Telang, 2004) stated the different reliability level of information from different types of websites. These types of information source will be used to reflect different level of relevancy for a particular vulnerability. This research will briefly conduct a statistical results of gathered webpages to show the significant regularity publishers of vulnerability information.

3.2.3 Hits in Public Interest

Public interest is reflected by the amount of related information available on the Internet. This includes all contexts stated in 3.2.1. From the common knowledge, the higher hits from search engines reflect more public interest about the topic. People will discuss or post much information about their interested topics over time. Qamra et al. presented the relationship between community interest and time through blogger's behavior (Qamra et al., 2006). The results shows the relationship between content-community during each time period. The amount of results from different search services or combination of search services can be different depends on matching and ranking algorithms used in services. As the statistical report of Top 20 Sites and Engines of Hitwise (Hitwise, 2009), Google gain 65% of search engine market share and hold the first ranked in years, this can imply the correctness and reliable of Google search service. In this research, data analysis based on data distribution will be used to evaluate popularity of a particular vulnerability by limited the search result from Google search service through API.

3.2.4 Information Aging

On the assumption that public information on the Internet may span a long time interval, retrieved information from the web may have different validity in terms of age. From researches in vulnerability life cycle, an obsolescence of vulnerability can be re-

ferred from lowering of public awareness or attention on a particular vulnerability. In our research, we consider information age by considering the context type over time from the inception of CVE to the current date.

3.3 The Framework

In our definition of relevancy, information relating to the lifecycle principally used to identify the level of relevancy of a particular vulnerability. In this work, we devise a relationship between public information of a vulnerability to its lifecycle states. Figure 3.1 shows how public information is related to vulnerability lifecycle. Lifecycle information of vulnerability *A* is extracted from its related document on the Internet using the domain specific ontology (Vulnerability Lifecycle Ontology-VLO).

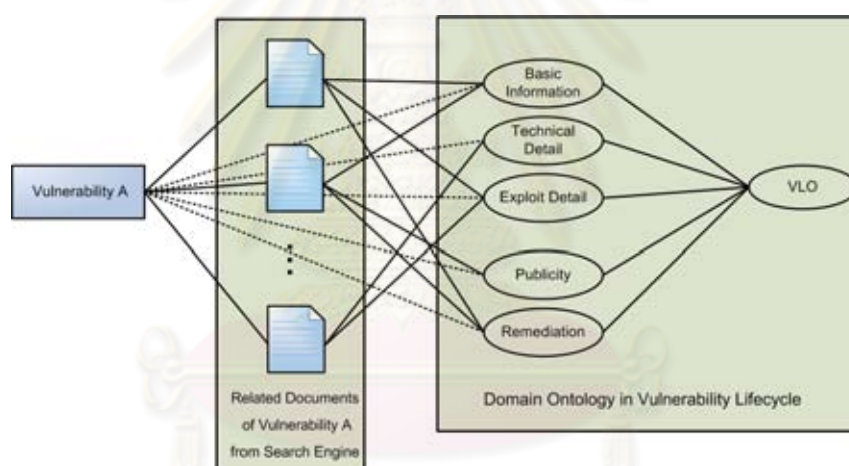


Figure 3.1: Public Information and Lifecycle State Relationship.

The Vulnerability Relevancy Ranking Framework is composed of three parts: knowledge management, lifecycle analysis and relevancy quantification, as shown in Figure 3.2. Each part is described as follows.

3.3.1 Knowledge Management

Firstly, vulnerability lifecycle knowledge is built. Vulnerability related information is extracted from security websites, software vendors, vulnerability standard naming systems, and well-defined taxonomy. Ontology (VLO) was devised to describe the relationship between vulnerability related information and their states in the lifecycle defined in 3.1. The detail design, usage of ontology, and the evaluation of knowledge base are discussed in Chapter 4.

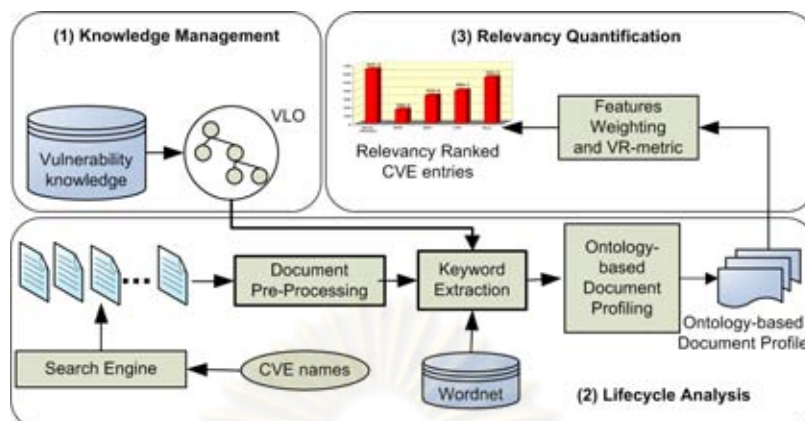


Figure 3.2: Vulnerability Relevancy Ranking Framework.

3.3.2 Lifecycle Analysis

To identify relevancy level of a vulnerability. The public information have to be processed. In Lifecycle Analysis, the selection and analysis of related information based on vulnerability lifecycle ontology will be described. We also introduce the concept of subcontext in ontology and the Context Sensitive Profile to represent a vulnerability in term of its lifecycle. Detail information is described in Chapter 5.

3.3.3 Relevancy Quantification

We defined four possible relevancy factors as context type, information age, data distribution ,and reliability of information source in Section 3.2. To evaluate relevancy level reflected from these factors, In Chapter 6, we present the analysis of these factors and how each factor effect relevancy quantification.

3.4 Summary

We introduce the vulnerability relevancy definition based on public interest. We also proposed the Vulnerability Relevancy Ranking Framework. The framework comprise of Knowledge Management for evaluating the public information of a vulnerability, Lifecycle Analysis for creating individual page profile, and Relevancy Quantification for evaluating vulnerability relevancy based on a collection of related information. The detail information of subsystems in this framework is discussed in Chapter 4, Chapter 5, and Chapter 6 respectively.

CHAPTER IV

VULNERABILITY LIFECYCLE ONTOLOGY

Our research roadmap is to define a framework for prioritizing vulnerabilities based on relevancy gleaned on online public information. In this chapter, we focus on the use of ontology as a knowledge base for describing the relationship between vulnerability-related information and their states in the lifecycle.

4.1 Information Source

To create the knowledge base of vulnerability lifecycle, information was gathered from various reliable sources as follows.

Vulnerability Taxonomy and Ontology: We study taxonomy of vulnerability and attack in order to gather related concepts as a baseline for our ontology. Landwehr, et al. attempted to organize information on security flaws for software development (Landwehr, 1981). Wita and Teng-amnuay presented the profiling scheme of severity based on CVE information for system administrative purpose (Wita and Teng-Amnuay, 2005). Moreira and his colleagues developed the security related ontology, ONTOVUL and ONTOSEC, to describe the relationship between vulnerability and security incident for security management in the organization (Moreira et al., 2008).

Vulnerability Standards and Databases: A major information source used to create VLO is that organized by Mitre: Making Security Measurable project (Mitre, 2007b) which includes CVE (Mitre, 1999) , CWE (Mitre, 2007a), CAPEC (Mitre, 2008), and CPE (Mitre, 2009). This provides standard knowledge representations, enumerations, exchange formats and languages, as well as sharing of standard approaches to key compliance and conformance mandates. Another information source used in this research is online vulnerability databases. NVD (NIST, 1999) and OSVDB (OSVDB, 2008) maintain vulnerability information for public use. OSVDB provides type of solution available for particular vulnerability while NVD provides information about exploitation requirements and consequences.

Online Information: Another information source is online documents. In this research, online information is defined as global web-based information available to the public. Online information is a form of long-term archive which can be used as knowledge base for any particular topic. We explored reliable security websites, governments, news agencies, and system vendors which continuously publish vulnerability information and discussion for the public because they reflect the concern of the public at large. For example, a lot of hits on the search of a particular vulnerability implies related incidents, such as disclosure of vulnerability information, available of exploit code or remediation process, and related news in critical impact of a specific vulnerability. Security websites such as CERT, VUPENSecurity, ISS X-Force, Secunia, and SecurityFocus are global security advisories that provide technical detail of vulnerability, while exploit information is available from Milw0rm, Packetstorm, and SecurityVulns. Vendor websites, such as Redhat, Mozilla, and Microsoft provide disclosure and remediation information of their own products.

4.2 Knowledge Representation

In this research, lifecycle states and their relations to online document are the basis for our vulnerability relevancy ranking framework. We define online document categories and types related to each lifecycle state. Web Ontology Language-Description Logic (OWL-DL) (Bechhofer and et al., 2004) and Simple Knowledge Organization System (SKOS) (Miles, 2009) are used to represent concepts, relationships and thesaurus of concepts. In this work, we follow “Formal/Semi-Formal Hybrids” pattern from World Wide Web Council (W3C, 2004) to construct the hierarchy of vulnerability related concepts and relationships by OWL structure. Furthermore, we use SKOS to model the vocabulary of vulnerability concepts such as prelabel and alltable in representing the semantic related vocabulary.

4.3 Creation of VLO

To identify the various states of the lifecycle of a particular vulnerability from online information, priori knowledge is needed. Ontology is selected to represent vulnerability lifecycle information due to its hierarchical structure. In this section, building framework for Vulnerability Lifecycle Ontology (VLO, pronounced vee-lo) is described. We gathered vulnerability related information and vulnerability lifecycle concepts to create VLO.

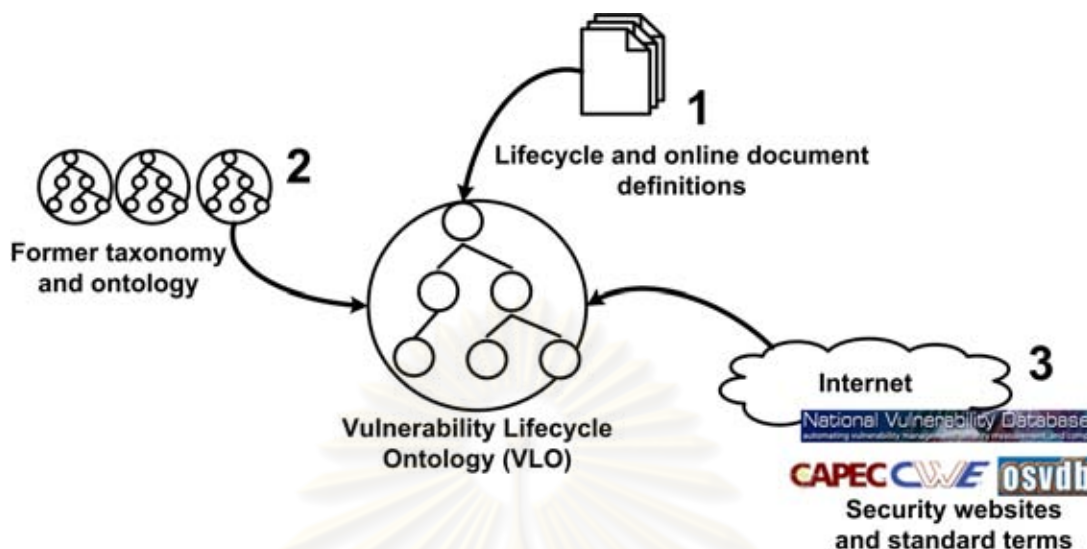


Figure 4.1: VLO Knowledge Building Process.

VLO knowledge building process is depicted in Figure 4.1.

Creation of VLO comprises three steps as follows.

Step 1: Identify main concepts from lifecycle and web document.

From the definition of vulnerability lifecycle and its information types, lifecycle states act as fundamental ontology concepts in VLO. Vulnerability has five states including discovery, disclosure, exploit, publicity, and remediation. The relationship “identified by” and “identify” relates the lifecycle states to online information and vice versa.

Step 2: Import existing concepts from ontologies.

To avoid reinventing the wheel, some of the concepts can be gleaned from existing ontologies. Moreira et al. have collected vulnerability and security related concepts to construct ONTOVUL and ONTOSEC (Moreira et al., 2008). In this work, we selected vulnerability related concepts from ONTOVUL and ONTOSEC to describe online information as identifier concepts. Figure 4.2 illustrates concepts and relationships in VLO that encompass imported concepts from ONTOVUL and ONTOSEC.

Step 3: Populate concepts with security keywords.

Keywords are manually retrieved from various vulnerability-related standards and

reliable security websites to populate the knowledge under main concepts of the VLO by domain experts. Keywords from CWE, and CPE are used in vulnerability types, asset names, and consequence and severity in basic information. NVD provide severity and consequences. CAPEC provided attack method to populate in exploit detail. Technical detail, Publicity and Remediation are mostly extracted from reliable security company, advisories, and system vendors websites.

Table 4.1 summarizes the lifecycle states and online document related to each state and also specify identifier concepts from extraction in step 3.

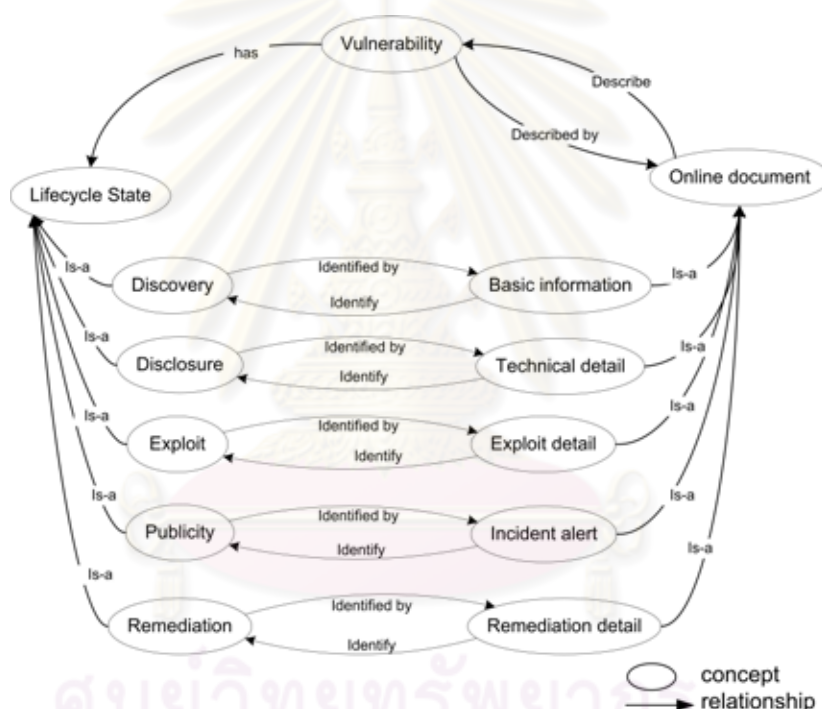


Figure 4.2: Base Concepts and Relationships of VLO.

```

\footnotesize
<owl:Class rdf:about="#Remediation">
  <<<<rdfs:subClassOf rdf:resource="#Web_document"/>
</owl:Class>

<owl:ObjectProperty rdf:about="#describeRemediation">
  <<<<rdfs:range rdf:resource="#&Ontovul;Correction"/>
  <rdfs:domain rdf:resource="#Remediation"/>
</owl:ObjectProperty>

```

Figure 4.3: OWL Definition of "Remediation" Concept in VLO

OWL Concepts Representation: Concepts and relationships in VLO are created

Table 4.1: Lifecycle States and Related Public Information

Lifecycle states	Indicated by	Identifier concepts
Discovery	Basic information	Type –CWE (328 concepts) Asset –CPE (17000 concepts) Consequence –NVD Severity
Disclosure	Technical detail	Precondition Attack –CAPEC (298 concepts)
Exploit	Exploit detail	Tools
Publicity	Incident alert	Security Incidents
Remediation	Remediation detail	Correction Patch Workaround Configuration changed Security updates

in OWL file using Protege-OWL software version 4.0 (Stanford, 2007) with SKOSEd extension (Simon, 2009). Defined concepts and relationships are represented using `< owl : class >` and `< owl : ObjectProperty >` respectively. In Figure 4.3, definition of remediation concepts and its relationships in OWL format is shown.

VLO Vocabulary Enrichment: After defining concepts and their relationships in OWL format, each concept is also organized into

```
< rdfs : subclassOf rdfs : resource = "&skos; Concept" / >
```

in order to create a vulnerability thesaurus. The vocabulary in SKOS extension is used to represent preferred label and alternative label of ontology concept using `< skos : prefLabel >` and `< skos : altLabel >` respectively. Labels of concepts defined in this work are retrieved from reliable security website defined in Section 4.1 with the help of domain expert. Figure 4.4 demonstrates a concept “Tools” and related vocabulary defined in VLO and in Figure 4.5, main structure of VLO are presented.

```

\footnotesize

<owl:Class rdf:about="&OntoSec;Tool">
  <rdfs:subClassOf rdf:resource="&skos;_Concept"/>
</owl:Class>

<OntoSec:Tool rdf:about="#Tools">
  <rdf:type rdf:resource="&owl;Thing"/>
  <skos:prefLabel>Tools</skos:prefLabel>
  <skos:altLabel>attack tools</skos:altLabel>
  <skos:altLabel>exploit tools</skos:altLabel>
  <skos:altLabel>script</skos:altLabel>
  <skos:altLabel>attack script</skos:altLabel>
  <skos:broader rdf:resource="&#Exploit_detail"/>
</OntoSec:Tool>

```

Figure 4.4: Thesaurus of “Tools” Concept in VLO.

4.4 Example Usage in Document Extraction

This section demonstrates how VLO is used to classify online document and to infer lifecycle states of the document. We employed one of SANS Top 20 Security Vulnerability in Web Browsers (SANS, 2007), CVE- 2007-0217. Information from the CVE website and from iDefense are selected as example in classification here.

Example 1: CVE information from <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0217> (published on January, 07, 2007). The description on CVE page is:

“The wininet.dll FTP client code in Microsoft Internet Explorer 5.01 and 6 might allow remote attackers to execute arbitrary code via an FTP server response of a specific length that causes a terminating null byte to be written outside of a buffer, which causes heap corruption.”

After pre-processing the description above, vulnerability related keywords are extracted. For example, Internet Explorer and heap corruption are considered as vulnerability related keywords. These keywords are mapped into VLO so that we can label them. In this case, Internet Explorer is mapped under Microsoft in User Application subclass, and heap corruption is known as Type, as shown in Figure 4.6. From Figure 4.6, content in CVE website contain basic information about vulnerability. These can be inferred that this web content is relevant to a particular CVE as basic information which indicates

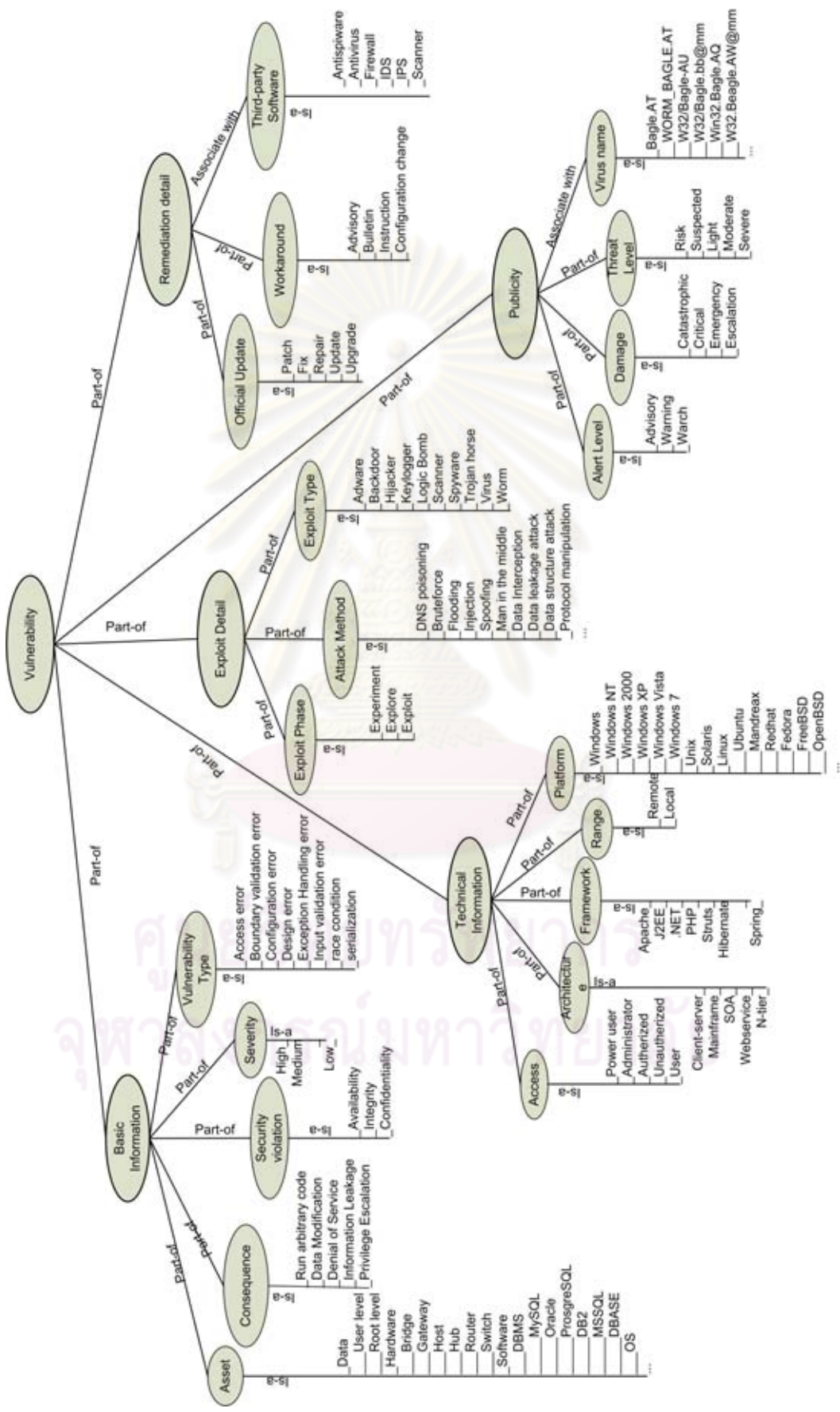


Figure 4.5: The Structure of VLO

discovery state in lifecycle.

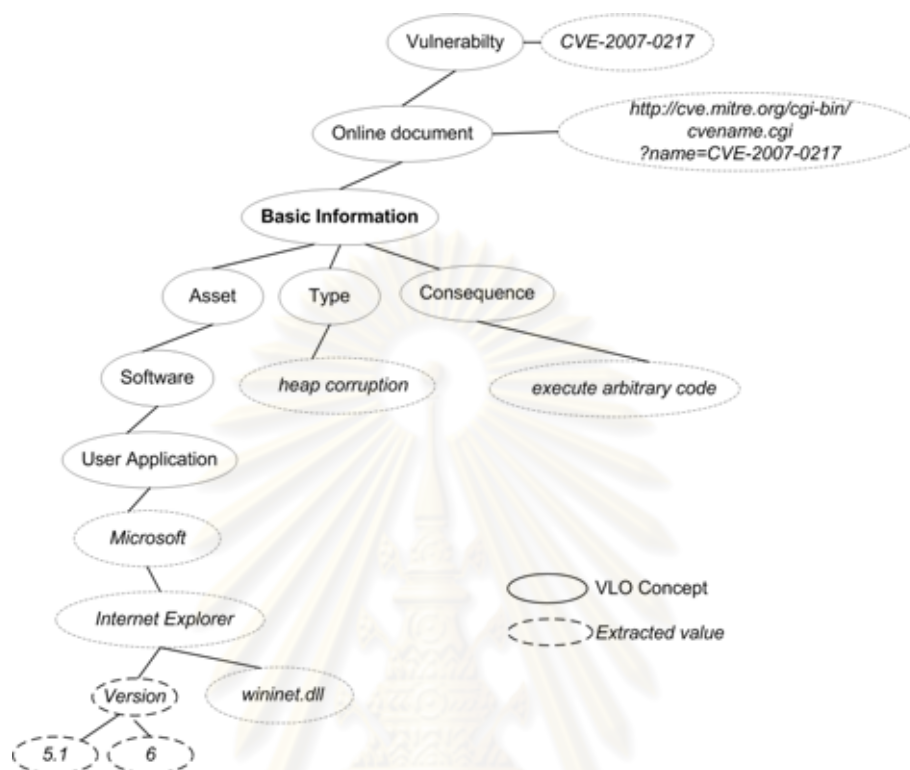


Figure 4.6: Extracted Keywords from CVE Website Describing CVE-2007-0217.

Example 2: Vulnerability information related to CVE-2007-0217 described in iDefense website (published on February, 13, 2007) available on : <http://labs.iddefense.com/intelligence/vulnerabilities/display.php?id=473> Information from iDefense contains full disclosure information and remediation of CVE-2007-0217. Some part of the information in the link above is described here.

“Successful remote exploitation of this vulnerability would allow a attacker to execute arbitrary commands in the context of the currently logged in user.

In order to exploit this vulnerability, the attacker must convince the target to follow a link in a program which uses the vulnerable functions, such as Internet Explorer, Word, or Outlook. For any of these applications it is sufficient to embed an image linked to a malicious ftp server; but for modern versions of Outlook, the image will not render unless the user allows it. iDefense is unaware of any effective workarounds for this vulnerability. Blocking outgoing port 21 (ftp) requests is not effective, as this it is possible to supply an ftp URL with an alternative port. It may be possible to limit exposure to this vulnerability by configuring systems to use a proxy server for all ftp requests and only allowing white-

listed sites.”

This information describes technical detail and remediation information. In Figure 4.7, we can depict the extracted keywords as basic information, technical detail, and remediation. It can be inferred that this document is the relevant to remediation state in lifecycle.

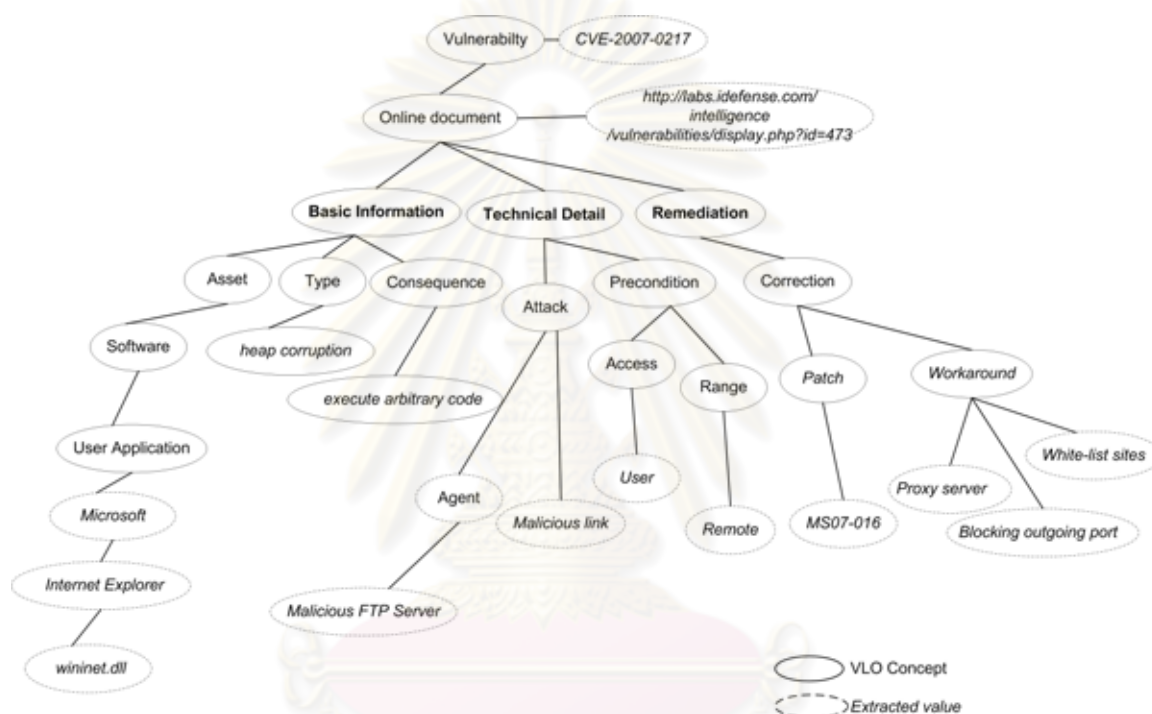


Figure 4.7: Extracted Keywords from iDefense Describing CVE-2007-0217.

Example 1 and 2 demonstrate the applicability of VLO to online information for vulnerability analysis. Our ongoing work uses VLO in collecting and classifying various online information. The collection will be further used in relevancy analysis of vulnerability.

4.5 Evaluation

We evaluate VLO on its fitness to the domain knowledge on vulnerability. Our evaluation procedure follows the Data Driven Ontology Evaluation technique (Brewster et al., 2004). A corpus of 363 CVE-related documents is collected via search engine using CVE names. Keywords are extracted from the corpus using KEA with 96 training data, and 297 testing data. We also expand the corpus with synonym from WORDNET (Princeton). Precision, Recall and F1 measure are used to measure semantic fit between

the ontology and the corpus. Precision denotes the ratio of ontology-corpus match and the ontology keywords while Recall indicates ratio of matched corpus keywords and total corpus keywords. A comparison of the three ontologies: ONTOSEC, ONTOVUL, and VLO is shown in Table 4.2.

VLO yield a more satisfying result in semantic fit to the CVE-related corpus, as shown in Table 4.2. But keep in mind that VLO is constructed from standard keywords in vulnerability-related community, while ONTOSEC and ONTOVUL are constructed from generic concepts in security and vulnerability domain.

Table 4.2: Comparison Results between Three Ontologies

Ontology	Precision	Recall	F1 Measure
ONTOVUL	0.0435	0.0034	0.0062
ONTOSEC	0.0441	0.0164	0.0240
VLO	0.0971	0.0279	0.0433

4.6 Summary

We described the creation of Vulnerability Lifecycle Ontology (VLO). The VLO is based on vulnerability concepts, taxonomy, and online information. The VLO is to be used in information retrieval to classify any vulnerability and estimate its relevancy.

Trial use of the VLO on online sources indicates the ability of using ontology in classifying vulnerability related online document. In our ongoing work, VLO will be used in conjunction with search strategy to retrieve and analyze web document related to a particular vulnerability in order to indicate relevancy of vulnerability.

Evaluation of the VLO on CVE-related online sources indicates the ability of using VLO in classifying vulnerability related online document. In our ongoing work, VLO will be used in conjunction with search strategy to retrieve and analyze web document related to a particular vulnerability in order to indicate relevancy. Next Chapter, the usage of VLO in creating Context Sensitive Profile will be discussed.

CHAPTER V

ONTOLOGY BASED CONTEXT SENSITIVE PROFILE

5.1 Introduction

Web documents from search results are assumed to be more or less relevant to the query, but may appear in different contexts. Some documents may contain only one context which can be classified by traditional text classification, but some documents may contain more than one context. In this Chapter, we introduce the subcontext in ontology and the Ontology based Context Sensitive Profile in order to express different context information related to vulnerability. We consider two layers in the profile, single document and document collection layer. The richness or completeness of information in each context is considered in single document layer. This means the readable document should have informative detail in some specific contexts. While frequency of context in document collection refers to availability of the context. Document collection means a certain amount of documents collected from search result and related to a particular vulnerability. Figure 5.1 shows steps of process and intermediate result in estimating context sensitive profile of a particular vulnerability. In this Chapter, the process in round rectangular will be described.

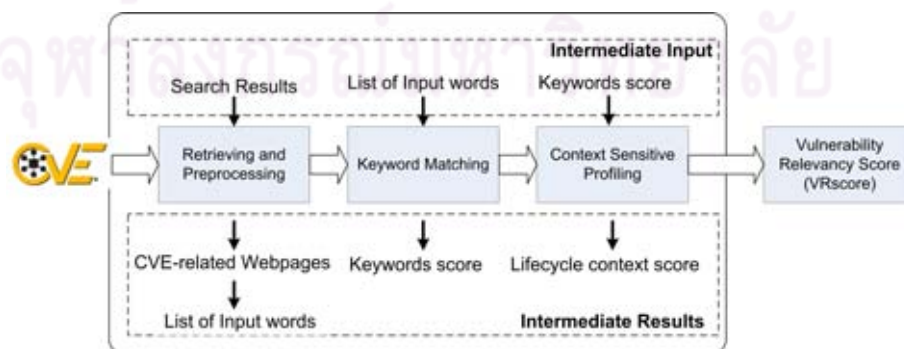


Figure 5.1: Processes and Intermediate Results

5.2 Subcontext in Ontology

In this research, domain ontology is used as a knowledge base. We introduce an *ontology subcontext* in order to indicate the relevancy of a document to the different aspects of a particular concept. Figure 5.2 shows the idea of concepts and subcontexts. We define a subcontext as the subtree rooted at a related concept of the target concept. For example, in Figure 5.2, concept *A* is a target concept. *B*, *C* and *D*, which are first level children of *A*, are the related concepts of *A*. Subcontext of *A* consists of 3 ontologies rooted at *B*, *C* and *D*.

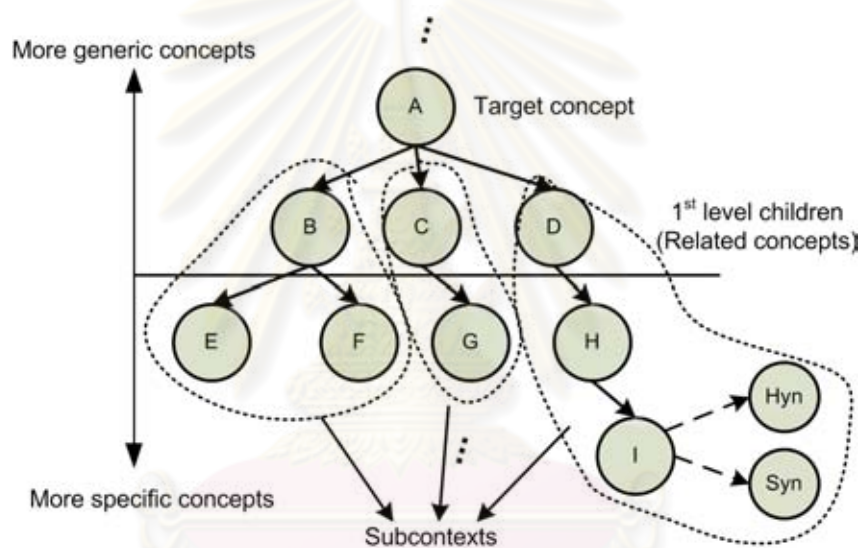


Figure 5.2: Ontology with Subcontext Structure

In Later section of this Chapter, we define matching function used in Context Sensitive Profile between input words from webpages and the ontology. We consider 2 layer of matching, keyword matching and context matching. Keyword matching identifies weight of keywords in specific domain based on vocabulary matching. Context matching considers the relationship type between first level child and root of subcontext (between *E*, *F* to *B*, *G* to *C*, and *H* to *D* in Figure 5.2).

The Idea of subcontext are used in VLO. Figure 5.3 presents the example of subcontext in exploit detail and remediation detail. *Exploit phase*, *Attack method*, and *Exploit type* are related to *Exploit detail* using “*part-of*” relationship. This means that to explain the existence of *Exploit detail*, we have to subsume from the existence of three related concepts. *Official update* are related to *Patch*, *Fix*, *Repair Update* and *Upgrade* by “*Is-a*” relationship. The existence of *Official update* can be one or more concepts from *Patch*,

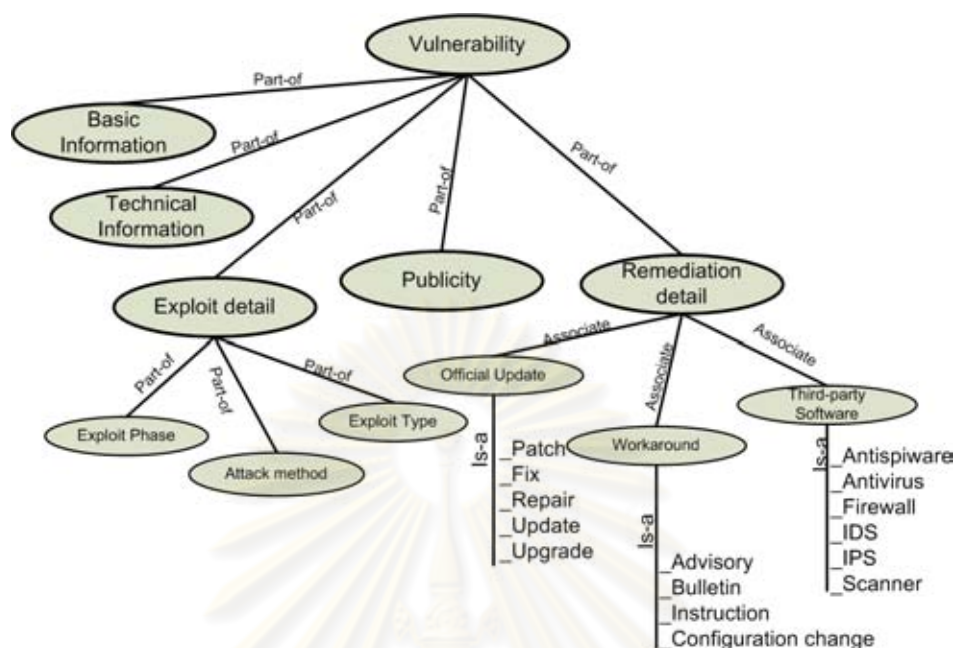


Figure 5.3: VLO and Subcontexts of Lifecycle States

Fix, *Repair Update* and *Upgrade* available. While the relation between Remediation detail and Official update can not be explained as taxonomic relationship, like “is-a” or “part-of”, we used “associate with” to represent relationship between them. This means that Remediation detail can be inferred by availability of Official Update, Workaround, or Third-party software to a certain degree, defined by domain expert. The detail information of using context in VLO is described in section 5.4.2.

5.3 Retrieving and Preprocessing of Public Information

The information used in this work is based on search result from Google search API. We use vulnerability standard name from CVE (Mitre, 1999) as a keyword searching for related public information. The search result for a CVE is collected and the top 30 ranked pages are captured using webcrawler. We previously capture 30 pages as the number 30 is the least statistical significant input. We conducted the experiment in 5.5.1 to select suitable number of webpages used in the framework.

After webpages were gathered, they were pre-processed. Normally, webpages are documents based on HTML structure. Aside from information content of the page, webpages also consist of decoration such as advertising banners, menus, links, etc. These decoration is considered as noisy data. Webpage preprocessing is the process that extract

the content of the webpage from decorations. We identify the noisy data structure based on the study of html tags and script. The special html tags and scripts for pictures, menus, and advertisements with external links are filtered out. Filtered webpages were tokenized and process using Porter stemmer with stopword removal (Rijsbergen et al., 1980). Each webpage is thus reduced to a list of input words.

5.4 Context Sensitive Profile

Context Sensitive Profiling is devised to signify specific context on the information of a webpage. In our research, Context Sensitive Profile is used to identify Lifecycle state information presented in a group of webpages which is a representative of a particular vulnerability.

Figure 5.4 shows the process of creating a context sensitive profile and a vulnerability relevancy score for vulnerability *A*. Vulnerability *A* has several webpages containing information in different context. Top ranked result from searching of vulnerability *A* are collected and lifecycle related information is extracted from webpages to create document profile and the collection of document profiles will be used to create context sensitive profiling and evaluate the relevancy of vulnerability *A*.

Later in this section, the process of creating context sensitive profile is presented.

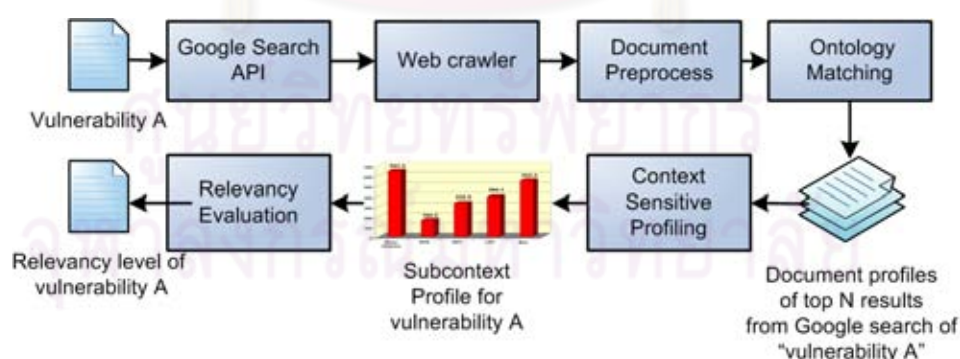


Figure 5.4: Example of Relevancy Evaluation for Vulnerability *A*

5.4.1 Keyword Matching

Keyword Matching based on ontology is used to identify domain fitness of information from webpages to a specific domain. Extracted keyword from webpages described in section 5.3 are matched with vulnerability knowledge base in VLO described in Chap-

ter 4. We proposed ontology matching algorithm considering thesaurus matching and concept matching between keywords and concepts in ontology.

5.4.1.1 Thesaurus Matching

Ontology is composed of upper ontology and domain ontology. Upper ontology provides core glossary across domains while domain ontology considers the domain-specific meaning and relationships of the concepts.

In this work, we used the Vulnerability Lifecycle Ontology (VLO) as domain ontology in order to identify the lifecycle-related information in the webpages. And we employ WordNet (Princeton) as an upper ontology in order to expand the vocabulary as a thesaurus of concepts in VLO. Synonyms and hypernyms of ontology concepts were used in order to provide alternatives to a particular concept. We developed the idea of thesaurus matching from (Varelas et al., 2005). Let l be any single word from a concept's label, s be its synonym and h its hypernyms, respectively. The input word w from the document was processed using thesaurus matching into its weights $Tres(w)$ as in Equation 5.1.

$$Tres(w) = \begin{cases} 1, & \text{if } w = l \\ 0.7, & \text{if } w = s \\ 0.5, & \text{if } w = h \end{cases} \quad (5.1)$$

5.4.1.2 Concept Matching

Concept matching uses the longest possible match between a list of input words from document and the concept label from the VLO in a particular subcontext. The concept label may be multi-word. The weighting in this work considered the maximum number of words matched in a particular concept label instead of a static weight for exact match and partial match used in (Brewster et al., 2004) because our empirical experiment indicated the dominance of partial weighting of compound words in a concept label from the sample documents. Let T be a target concept which has a direct relationship with its $R_{(i)}$ related concepts where $1 \leq i \leq p$. $X_{(1)}, X_{(2)}, X_{(p)}$ is a set of subcontexts rooted at T 's related concepts. For any $M_W \in X_i$, where M_W is the compound word in the concept label that matches a list of input words W , where $W = w_1, w_2, \dots, w_n; n \geq 1$, and

the concept matching score is expressed as in Equation 5.2:

$$match(w_i, M_W) = \frac{1}{length(M_W)Tres(w_i)}$$

$$Match(W, M_W) = \sum_{i=1}^n match(w_i, M_W), \quad (5.2)$$

where $length(M_W)$ is the number of words in the label of concept M_W . From Equation 5.2, for any single word label M_W where $n = 1$, $Match(W, M_W) = Tres(w_i)$.

5.4.2 Context Matching

We consider two level in the profile, single webpage level and collection of webpages level. The richness or completeness of information in each context is considered in single document layer. This means the readable document should have informative detail in specific context. While frequency of context in document collection refers to availability of the context. Document collection means a certain amount of documents collected from search result and related to a particular vulnerability.

5.4.2.1 Context Richness

The relationship in domain ontology is basically defined as is-a (subclass-superclass), part-whole (composite/aggregate), and association (Gulla and Brasethvik, 2008).

Definition 1: Is-a relationship. *Is-a* relationship describes taxonomic relationship between concepts or between instance and concepts. This represents subclass-superclass notion in class diagram. From Figure 5.3, patch, fix, repair, update, and upgrade are all subclass of *Official Update* concept. We can subsume Official Update when one of the subclass concept exists in the document.

Given $\alpha(a) = \{0, 1\}$ represent the existence of concept a in a document. For a is-a A ,

$$\text{If } \alpha(a) \text{ then } \alpha(A), \quad (5.3)$$

where $\alpha(A) = \{0, 1\}$ infers to the existence of concept A .

Definition 2: Part-whole relationship. Part-whole relationship describes combination of concepts in respect to another concept. Part-whole relationship is bi-directional represented by *part-of*, and *has-part* as its reverse relationship. Gulla and Brasethvik also stated in their work that part-whole relationship includes both the notion of aggregation and composition from UML (Gulla and Brasethvik, 2008). From Figure 5.3, *Exploit Detail* concept is described by a combination of exploit phase, attack method, and exploit type. To subsume Exploit Detail, document should have all or almost of its part concepts.

Given $\alpha(a_i) = \{0, 1\}$ as the existence of concept a_i in document. For any a_i *part-of* A ,

$$\text{If } \sum_i^N (\alpha(a_i)) > \mu_p N \text{ then } \alpha(A), \quad (5.4)$$

where $\alpha(A) = \{0, 1\}$ infers the existence of concept A , N is the number of *part-of* relationship from A , and μ_p is a certain threshold for *part-of* relationship.

Definition 3: Associate with Relationship. *Associate with* represents non-taxonomic logical relationship between concepts. In Figure 5.3, Official Update, Workaround, and Third-Party Software, while having no logical relation among each other, are related to *Remediation Detail* concept. Having one of the concept can infer remediation detail.

Given $\alpha(B) = \{0, 1\}$ represent the existence of concept B in document. For A *associate with* B ,

$$\text{If } \sum_i^M (\alpha(\omega_{AB_i})) > \mu_s M \text{ then } \alpha(A), \quad (5.5)$$

where $\alpha(A) = \{0, 1\}$ infers the existence of concept A , ω_{AB} is the weight of semantic similarity between A and B , and M is the number of *associate with* relationship from A . μ_s is a certain threshold for *associate with* relationship.

5.4.2.2 Context Availability

The Context Sensitive Profiling identifies subcontext relevancy of crawled web-pages. Matched concepts in a crawled page are used to create document profile of each

page, and frequency of context in document collection represents the context sensitive profile of a vulnerability.

For subcontext A in document collection V .

$$\theta(A_V) = \frac{\sum_i^{|V|} (\alpha(A_i))}{|V|}, \quad (5.6)$$

where $\theta(A_V)$ represents availability of A in the document collection V . $\alpha(A_i) = \{0, 1\}$ is the existence of concept A in document i .

5.5 Experiments

5.5.1 Experiment 1 - Context Richness Evaluation

We conducted the experiment based on our context richness for a collection of CVE-related information from the search engine result. Thirty top ranked results were gathered from searching 12 CVE names. A total of 299 labeled documents were used. Each document was labeled by the expert as 0, 1 for non-relevant, and relevant for each subcontext described in Table 5.1. Note that a document may address more than one state of the lifecycle. The evaluation was made by comparing the classification results between using the Context Sensitive Profile, the traditional term-frequency document vector and the Class Match Measure Model (CMM) (Alani and Brewster, 2006) as described in Table 5.2, which evaluated the coverage of the ontology over words, using a 10-fold cross-validation SVM classification in Weka software.

Table 5.1: Training Dataset for Context Richness Evaluation

	Relevant	Non-Relevant
Basic Information	271	28
Technical Detail	262	37
Exploit Detail	115	184
Publicity	266	33
Remediation	194	105

5.5.2 Experiment 2 - Context Availability Evaluation

We did the experiment on real data from the CVE website (Mitre, 1999). 3000 CVE are randomly selected. 90000 public information pages are collected to create context sensitive profile for selected CVE with differing conditions. Table 5.3 shows the scope of

Table 5.2: Extraction Method of DV, CMM and CSP

	Document Vector (DV)	Class Match Measure (CMM)	Context Sensitive Profile(CSP)
Word Frequency	✓	✓	✓
Semantic Expansion	X	✓	✓
Ontology based Proximity	X	✓	✓
Concept Label Matching	X	✓	✓
Subcontext Consideration	X	X	✓

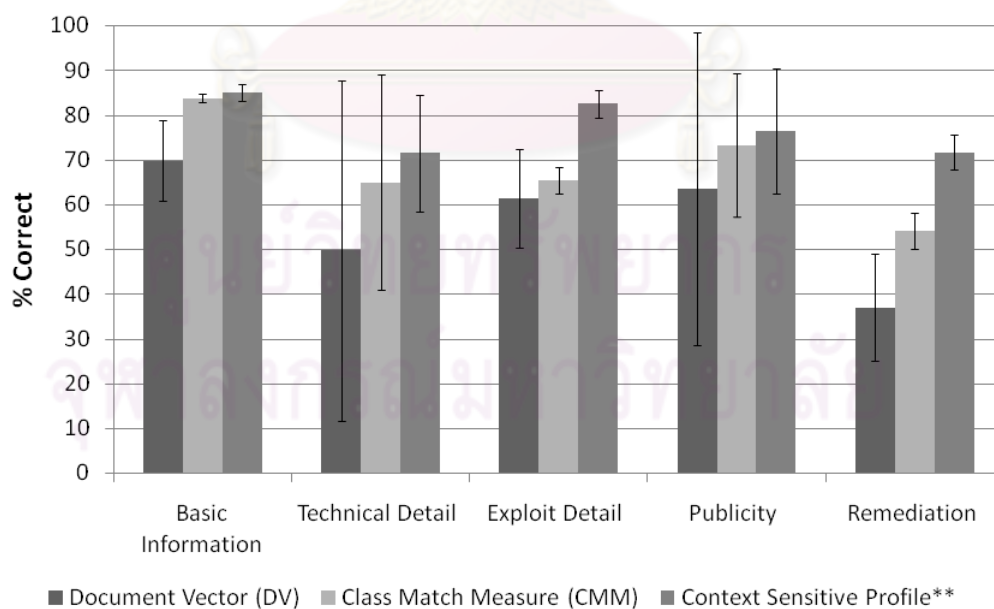


Figure 5.5: Evaluation Result of Context Richness

the training dataset used in this research.

Table 5.3: Training Dataset for Context Availability Evaluation

Year	no. of CVEs	crawled webpages
2005	500	15000
2006	500	15000
2007	500	15000
2008	500	15000
2009	500	15000
2010	500	15000
Total	3000	90000

We create the experiment to choose suitable parameters used for creating context sensitive profile. EM clustering algorithm is used for observing the characteristic of data distribution among lifecycle contexts. The experiment is done on 3000 CVE profiles. Five context sensitive profiles are created based on different parameters in Table 5.4. Term weight-Inverse document weight (TFIDF) is a weight often used in information retrieval and text mining (Liu, 2007). This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Context threshold is used to determine context richness from section 5.4.2. Number of pages is the number of webpages from search result which is collected to create vulnerability's context sensitive profile.

Table 5.4: Clustering Mode in the Experiment

Mode	Ontology Match Condition		No. of Pages
	w/ TFIDF	Context Threshold	
10_0.5	X	0.5	10
20_0.5	X	0.5	20
30_0.5	X	0.5	30
10_0.6	X	0.6	10
20_0.6	X	0.6	20
30_0.6	X	0.6	30
10_0.5_tf	✓	0.5	10
20_0.5_tf	✓	0.5	20
30_0.5_tf	✓	0.5	30
10_0.6_tf	✓	0.6	10
20_0.6_tf	✓	0.6	20
30_0.6_tf	✓	0.6	30

Figure 5.6 depicts likelihood ratio for each subcontext in different mode from tabel 5.4. From the clustering result, mode “10_0.6_tf” or ontology match with TFIDF, using 0.6 as threshold and selecting 10 pages from search result to represent a CVE. TFIDF helps to capture keywords which may not be frequently stated but relevant, such as “solution available” or virus names. Threshold is reflected from context richness in Section 5.4.2.1. If the threshold is less than relationship weight in context richness, the meaning of relationship will be lost. Selecting 10 pages reduces duplication of information in pages. When the vulnerability is popular, the online document will be repost again and again. Multiple page duplication will distort the value in context profile.

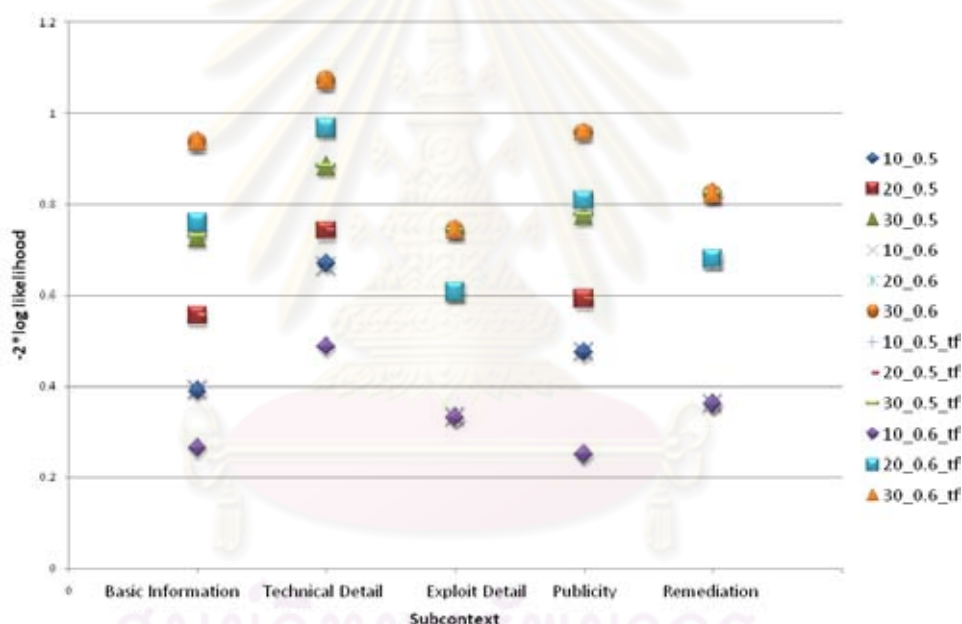


Figure 5.6: Evaluation Result of Context Availability

5.6 Summary

In this Chapter, the concept of ontology subcontext and Context Sensitive Profile are introduced in order to represent CVE-related document in terms of lifecycle contexts. The Context Sensitive Profile is conducted based on the analysis of the fitting of webpages content to specific context in ontology. The structure of Context Sensitive Profile can be depicted in Figure 5.7. Information level in consideration is divided into a webpage level and a collection of webpages level which collected from search result of the interested topic. Ontology matching level is divided into *context richness* which consider vocabulary

and different type of relationship matching and *context availability* consider information availability of a CVE on the Internet. Next Chapter, Context Sensitive Profile of a CVE will be used to evaluate relevancy of a vulnerability represented by a particular CVE.

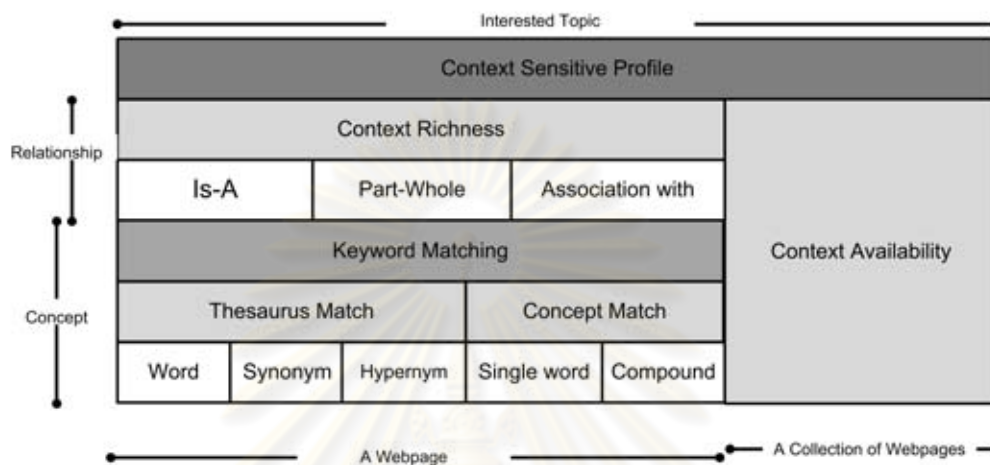


Figure 5.7: Context Sensitive Profile.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER VI

VULNERABILITY RELEVANCY QUANTIFICATION

In Chapter 3, we defined vulnerability relevancy as a level of public awareness or attention to a particular vulnerability. We defined four possible relevancy factors as hits in public interest, reliability of information source, information age, and context type. In this chapter, we present the analysis of these factors and how each factor effect relevancy quantification.

6.1 Hits in Public Interest

Hits in public interest of vulnerability information means the number of webpages that relate to a particular vulnerability. Since we cannot practically retrieve all related information, our model relies on the result from search service available. As stated in section 3.2.3 we select Google as our search service instead of a combination of various search services because of the market share (Hitwise, 2009). Moreover, combination of search result from many search services will result in redundancy. To avoid redundancy of information, we limit the result from only Google search service in this research.

From Google search result on exact match using “ ” of a particular vulnerability, we use the number of search result and top 30 ranked search results for this research. Figure 6.1 depict the example of data collected from search result on Google website.

Google searchAPI provided maximum results at 1000 pages for a query (Google, 2007). We valued the weight of search result ω_{hits} as log of the number of result as shown in Equation 6.1. High ω_{hits} reflects higher level in public interest and relates to higher relevancy. For those reached the maximum Google results is consider as maximum ω_{hits} .

$$\omega_{hits} = \begin{cases} \frac{\log_{10}(\text{searchresult})}{\log_{10}(1000)}, & \text{if } \text{searchresult} \leq 1000 \\ 1, & \text{if } \text{searchresult} > 1000 \end{cases} \quad (6.1)$$

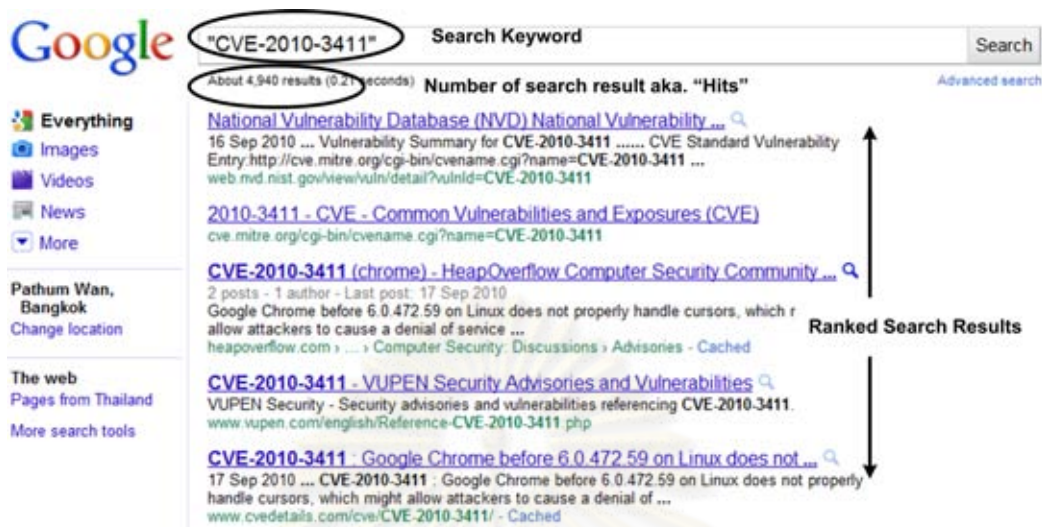


Figure 6.1: Search Result from Google Search Service

6.2 Information Source

As this research is focused on vulnerability context type based on lifecycle. Vulnerability information used in this research comes from different sources including announcements by software vendors and government agencies, news websites, technical discussion boards, and feedback forums hosted by software vendors. Different information sources usually provide different types of information, but may not always be the case. For example, software vendors publish vulnerability information and remediation pertinent to their products while news websites are more concern in outbreaks. Technical discussion webboards may contain in-dept information about exploitations, symptoms, or workarounds, and remediation.

Kannan and Telang (Kannan and Telang, 2004) proposed the different reliability level of information from different types of websites. Formal website which constantly publish vulnerability information reflect more reliable information about remediation than blogs, or personal website. Meanwhile, technical discussion websites and personal blogs contain more technical detail. The reliability of information source needs to be analyzed based on statistical analysis, which is beyond the scope of this work.

This research will briefly conduct a statistical result of gathered webpages to show the significant regular publishers of vulnerability information.

6.3 Information Aging

On the assumption that public information on the Internet may span a long time interval, retrieved information from the web may have different validity in terms of age.

An obsolescence of vulnerability can be referred from lowering of public awareness or attention on a particular vulnerability. From the empirical study, monitoring the publish date of the webpages is not feasible. The reason is that webpages contain both static and dynamic parts. Some static webpages have specific publish date or last update date that refers published date or modification date. But in the dynamic page, such as php page, “last update date” in the page is refer to php code modification. The age of vulnerability is considered as a weight for context sensitive profile as information age estimation. In our research, we consider information age by considering the context type over time from the inception of CVE to the current date. The inception of CVE is stated in CVE name of a vulnerability, such as “CVE-2010-0031” is a vulnerability number 31 discovered in year 2010. Equation 6.2 shows the weight of information age (ω_{age}) used in this research. We considered relevancy weight for vulnerability incepted only 10 years from recent year according to approximate software lifespan (Baxter and Pidgeon, 1997), (MacKay, 2006).

$$\Delta(year) = currentyear - inceptionofCVE$$

$$\omega_{age} = \begin{cases} \frac{10-\Delta(year)}{10}, & \text{if } \Delta(year) \leq 10 \\ 0, & \text{if } \Delta(year) > 10 \end{cases} \quad (6.2)$$

6.4 Subcontext Availability

From relevancy attributes, we create lifecycle based relevancy metric in order to evaluate the level of information in the states of the lifecycle. From (Jumratjaroenvanit and Teng-amnuay, 2008) and (Frei et al., 2009), different ordering of information on lifecycle states reflects different development of a vulnerability. In this work, we focus on availability and completeness of information based on Context Sensitive Profile stated in Chapter 5. From preliminary study and prior works in lifecycle analysis, we found that vulnerability have different distributions in different contexts.

The availability of information can be used to identify characteristic of vulnerabil-

ity, such as a vulnerability with higher availability in exploit detail reflects higher attacker intension (Dantu et al., 2004). Normal vulnerability usually has only *Basic Information* and some part of Technical Detail available on the Internet. The information will be duplicated to other security or news websites for more interesting vulnerability. From our definition, *Exploit Detail* reflects availability of global report on incident and possible attack scripts, *Publicity* reflects availability of automate attack tools, such as worm, and virus. The higher availability of information on *Publicity* infers higher impact on the public. *Remediation Detail* provides solution in dealing with vulnerability. It is usually available from vendor's website or security product's website. Availability of Remediation Detail reflects the effort of fixing that particular vulnerability.

From the subcontext availability, we define a metric based on the heuristic analysis in order to quantify the relevancy as shown in Table 6.1. Availability of information in each context is divided into 3 level: low, medium, and high. In Table 6.1, we define

Table 6.1: Context-Based Relevancy Metric.

Lifecycle Context	Information Availability		
	High	Medium	Low
Basic Information (λ_B)	3	2	1
Technical Detail (λ_T)	3	2	1
Exploit Detail (λ_E)	6	4	1
Publicity (λ_P)	6	4	1
Remediation Detail (λ_R)	1	4	6

the relationship between vulnerability relevancy score and the level of information availability in each context. From the definition of *Basic Information* and *Technical Detail*, information is usually describes based on characteristic of vulnerability. The score from availability level of these two context is assigned as 1,2, and 3 for low, medium, and high availability, respectively.

For *Exploit Detail* and *Publicity*, the availability of these two contexts reports possible impact and damage caused by vulnerability, we assign a higher relevancy score for high and medium availability of these two contexts.

Remediation reflects the availability of protection from vulnerability. If the remediation level is high, it means the vulnerability have ample information in protection or recovery. This results in lover relevancy of the CVE. We assign 6, 4, and 1 for Remediation

detail in low, medium, and high availability, respectively.

Figure 6.2 depicts the process of creating the Vulnerability Relevancy Quantification Model. We use sample data of 3000 CVEs to find the normal distribution of Context Sensitive Profile for each context described in Section 6.4.1.

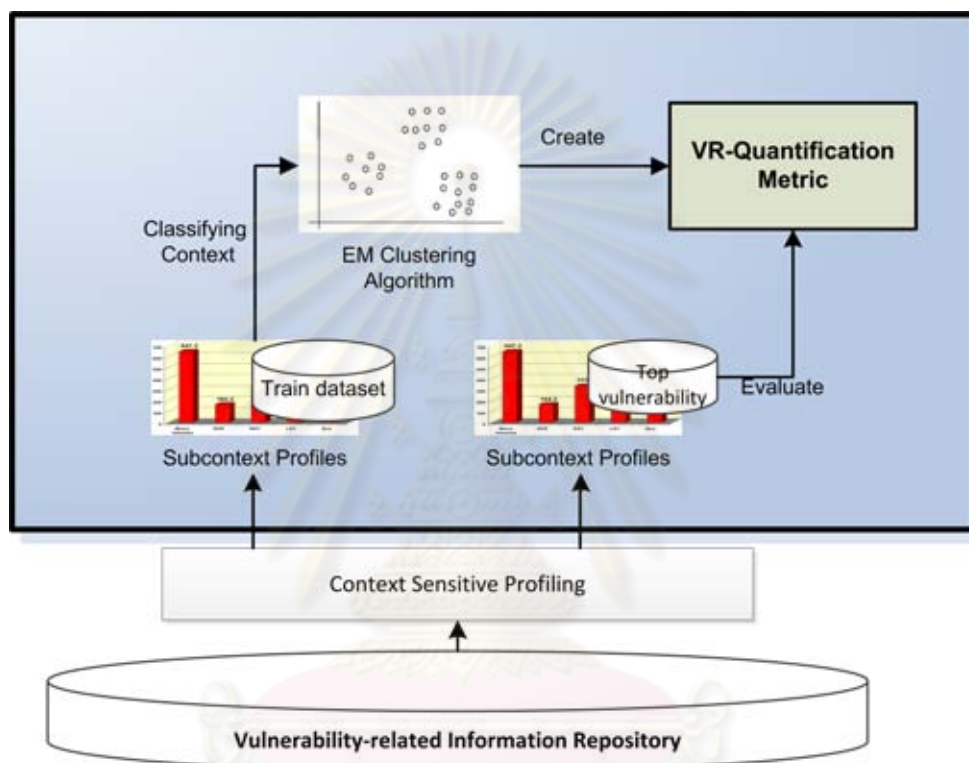


Figure 6.2: Vulnerability Relevancy Quantification Model.

6.4.1 Context Relevancy

We conduct another experiment using training dataset described in ?? to find the normal distribution of Context Sensitive Profile for each context, using EM clustering algorithm (Dempster et al., 1977) to separate data into 3 clusters representing high, medium, and low relevancy. EM clustering algorithm runs with 100 iterations.

Table 6.2 and Figure 6.3 summarize the result of clustering each context in Context Sensitive Profile. Figure 6.4 depicts the clustering distribution in training dataset for each context in Context Sensitive Profile.

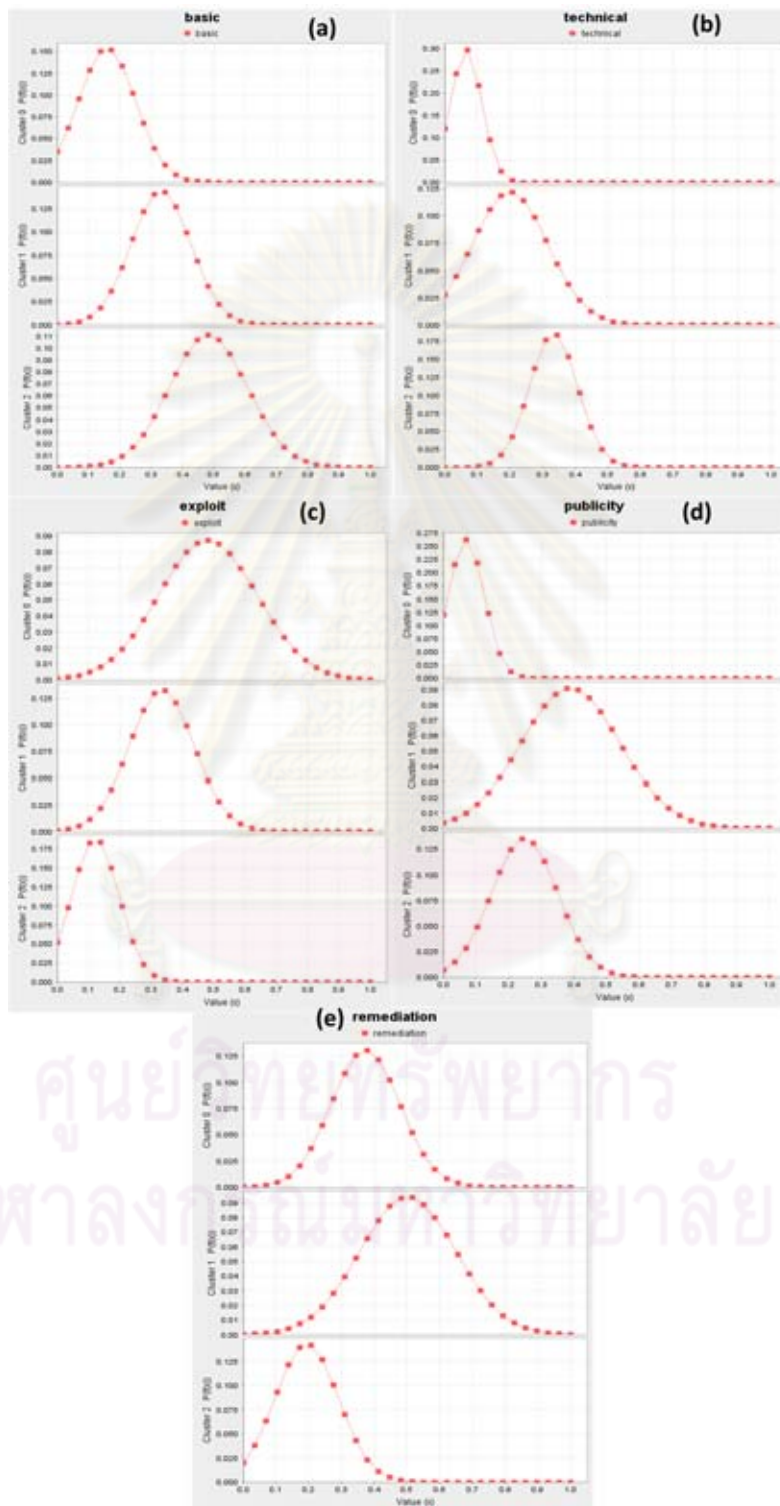


Figure 6.3: Clustering Result in (a) Basic Information, (b) Technical Detail, (c) Exploit Detail, (d) Publicity, and (e) Remediation.

Table 6.2: Clustering Result for Vulnerability Relevancy Quantification Model.

	Cluster	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation Detail
	0					
mean (μ_0)		0.1582	0.0651	0.4804	0.0696	0.3724
std. dev.		0.0921	0.0481	0.1575	0.0554	0.1051
	1					
mean (μ_1)		0.3316	0.2	0.3342	0.3785	0.503
std. dev.		0.0956	0.1163	0.1036	0.1509	0.1463
	2					
mean (μ_2)		0.4828	0.3338	0.1214	0.2491	0.1936
std. dev.		0.1241	0.0741	0.0755	0.1019	0.0975

6.5 Vulnerability Relevancy Quantification Model

From the clustering result in section 6.4.1, we have the distribution of different level of context availability in Context Sensitive Profile. From the normal distribution of each context from clustering result in Figure 6.3, mean and standard deviation of each cluster in each subcontext are listed in Table 6.2. From context characteristic, we can define the level of each subcontext that corresponds to subcontext availability analysis in Table 6.1. Table 6.3 shows the relevancy level mapping from normal distribution of each context in clustering result.

Table 6.3: Vulnerability Relevancy Level from Clustering Result.

Relevancy Level/Cluster no.	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation
High	2	2	0	1	1
Medium	1	1	1	2	0
Low	0	0	2	0	2

Relevancy level and relevancy scores are calculated based on the metric in Table 6.1. From the EM likelihood in equation 6.3, the value in each context in Context Sensitive Profile is classified into clusters linked to relevancy level listed in Table 6.2 with μ_0, μ_1, μ_2 as the mean of the cluster 0, 1, and 2 respectively in each context.

$$P(\text{data}|\mu_i) = \prod_{i=1}^N \sum_i P(\omega_i) P(x|\omega_i, \mu_1, \mu_2, \dots, \mu_k) \quad (6.3)$$

We calculate Context based relevancy score (R_{context}) from multiplication of con-

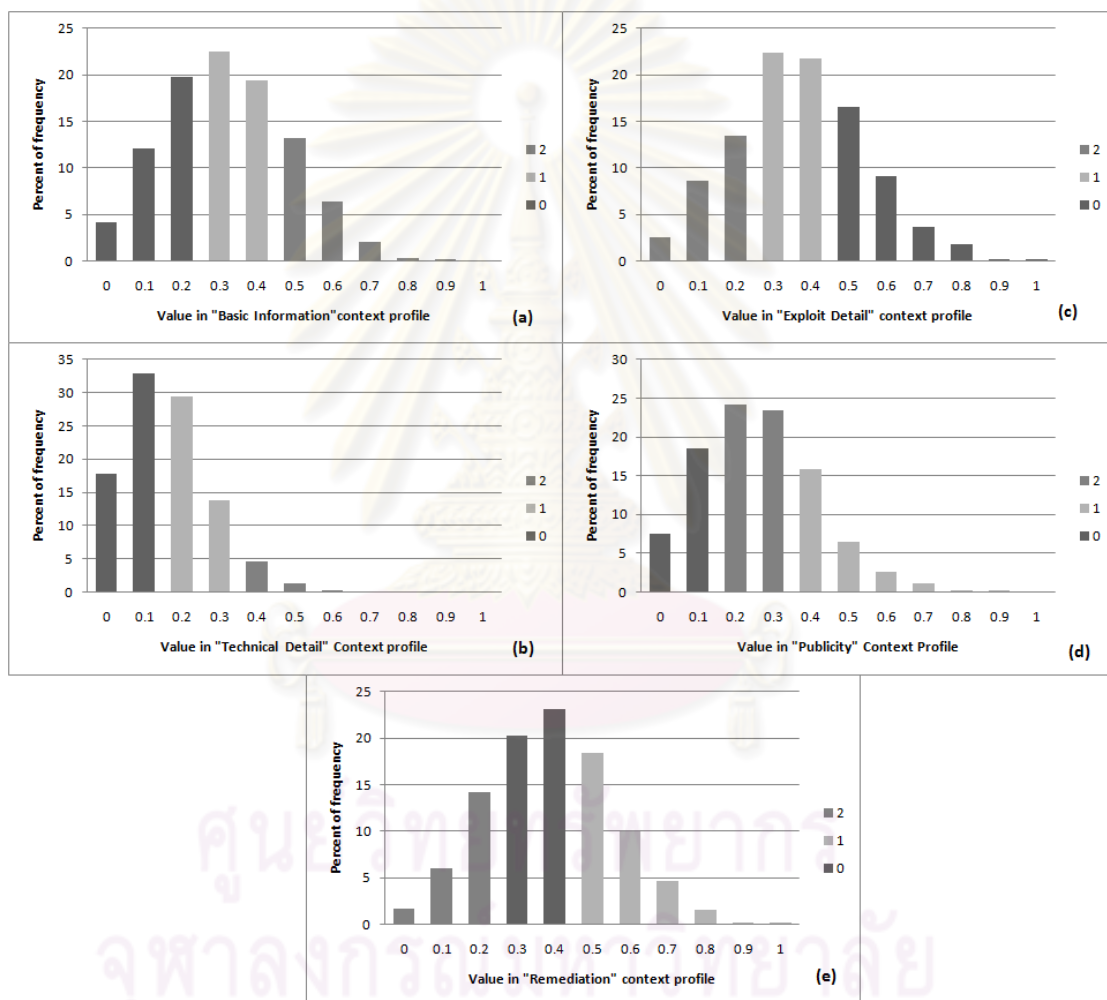


Figure 6.4: Lifecycle Context Distribution in (a) Basic Information, (b) Technical Detail, (c) Exploit Detail, (d) Publicity, and (e) Remediation

text scores. Equation 6.4 show how $R_{context}$ is calculated.

$$R_{context} = \lambda_B \times \lambda_T \times \lambda_E \times \lambda_P \times \lambda_R \quad (6.4)$$

The maximum value of $R_{context}$ come from vulnerability with fully publish information about Basic Information, Technical Detail, Exploit Detail, and Publicity with low availability of Remediation will get the highest score. This is because the vulnerability will have full information beneficial to exploit or attack, but no available in protection information. The Minimum value of $R_{context}$ come from vulnerability with fully available Remediation information with no or less exploit information available.

Table 6.4: $R_{context}$ Value Range

Context	max	min
Basic Information	High =3	Low = 1
Technical Detail	High =3	Low = 1
Exploit Detail	High =6	Low = 1
Publicity	High =6	Low = 1
Remediation	Low =6	High = 1
$R_{context}$	1944	1

We normalized $R_{context}$ to be in range 0-1, as in Equation 6.5

$$R_{norm} = \frac{R_{context} - 1}{1944 - 1} \quad (6.5)$$

The possible relevancy levels and vulnerability scores are tabulated in Appendix A

Vulnerability relevancy in our definition is represented by hits in public information, vulnerability age, and context sensitive information. We calculate vulnerability score (VRscore) for each vulnerability based on these three attributes as Equation 6.6.

$$VRscore = \omega_{age} \times \omega_{hits} \times R_{norm} \quad (6.6)$$

Figure 6.5 demonstrates the distribution of vulnerability relevancy score using different attributes on CVE inception in test dataset. In figure 6.5 ,(a) presents only context sensitive profile, the higher VRscore infer to more relevant vulnerability. We can spot the

highest relevancy score from a CVE incepted from year 2003 which is CVE-2003-0907, a moderate severity vulnerability in Microsoft XP possible for DOS attack. (b) consider context sensitive profile and vulnerability age. With this calculation the newer vulnerability will be raised as more relevant, but still maintain the relevant context sensitive profile vulnerability. (c) consider context sensitive profile and public interest hits. With this calculation the vulnerability with higher search results will be raised as more relevant, but still maintain the relevant context sensitive profile vulnerability. (d) consider all three attributes which represent the completeness of information, the availability of information and age of information.

6.5.1 Example Calculation of VRscore

For clarity, we present an example in constructing context sensitive profile and how it relates to vulnerability relevancy score. CVE-2005-0344 and CVE-2007-0038 are selected for this example.

The description of CVE-2005-0344 is “*Directory traversal vulnerability in 602LAN SUITE 2004.0.04.1221 allows remote authenticated users to upload and execute arbitrary files via a .. (dot dot) in the filename parameter.*”

CVE-2007-0038 is a “*Stack-based buffer overflow in the animated cursor code in Microsoft Windows 2000 SP4 through Vista allows remote attackers to execute arbitrary code or cause a denial of service (persistent reboot) via a large length value in the second (or later) anih block of a RIFF .ANI, cur, or .ico file, which results in memory corruption when processing cursors, animated cursors, and icons, a variant of CVE-2005-0416, as originally demonstrated using Internet Explorer 6 and 7.*”

To create the context sensitive profile, CVE-ID is used to search for related document from the Internet using Google Search API. CVE-2005-0344 brings about 340 search results while CVE-2007-0038 has 18,900 pages. In Table 6.6, we demonstrate the first rank in each search result on how the information is captured and used to create the Context Sensitive Profile. The rules in Section 5.4 is used to create Context Sensitive Profile with μ_p and μ_s both equal to 0.6 . Context Sensitive Profile based on 10 webpages from search results of CVE-2005-0344 and CVE-2007-0038 are as follows

Context Sensitive Profile

$$[\theta(B), \theta(T), \theta(E), \theta(P), \theta(R)]_{CVE-2005-0344} = [0.2, 0.1, 0.3, 0.1, 0.1], \text{ and}$$

$$[\theta(B), \theta(T), \theta(E), \theta(P), \theta(R)]_{CVE-2007-0038} = [0.3, 0.1, 0.8, 0.3, 0.4]$$

To calculate relevancy score for a vulnerability, Context Sensitive Profile is used to estimate Context based Relevancy (R_{norm})-based on clustering model in section 6.4.1, Hits (ω_{hits}), and Age (ω_{age}) weight.

Hits and Age factors

ω_{hits}

$$\omega_{hits}^{CVE-2005-0344} = \frac{\log_{10}(340)}{\log_{10}(1000)} = 0.8443$$

$$\omega_{hits}^{CVE-2007-0038} = 1.0000$$

ω_{age}

$$\omega_{age}^{CVE-2005-0344} = \frac{10 - (2011 - 2005)}{10} = 0.4$$

$$\omega_{age}^{CVE-2007-0038} = \frac{10 - (2011 - 2007)}{10} = 0.6$$

Vulnerability Relevancy Score

$$[\lambda_B, \lambda_T, \lambda_E, \lambda_P, \lambda_R]_{CVE-2005-0344} = [1, 1, 4, 1, 6]$$

$$R_{norm}^{CVE-2005-0344} = \frac{(1 \times 1 \times 4 \times 1 \times 6) - 1}{1944 - 1} = 0.0118$$

$$[\lambda_B, \lambda_T, \lambda_E, \lambda_P, \lambda_R]_{CVE-2007-0038} = [2, 1, 6, 6, 4]$$

$$R_{norm}^{CVE-2007-0038} = \frac{(2 \times 1 \times 6 \times 6 \times 4) - 1}{1944 - 1} = 0.1477$$

From the example of VRscore calculation results and the relevancy rank in Table A.1, CVE-2007-0038 has R_{norm} in rank 9 while CVE-2005-0344 has got R_{norm} in rank 22. This means that from the availability of lifecycle information, CVE-2007-0038 is more relevant than CVE-2005-0344. The R_{norm} can also be computed in conjunction

with age and hits of the vulnerability. In this case the VRscore will be 0.0039 for CVE-2005-0344 and 0.0886 for CVE-2007-0038. The exploitation of CVE-2007-0038 was found in “animated cursor” library that is used in multiple Microsoft products. The exploit code impacts widely. Meanwhile, CVE-2005-0344 affected 602LAN SUITE, which is more likely specific software. The exploitation of CVE-2005-0344 is also limited to specific system, and thus is considered low relevancy vulnerability. When the system contains both vulnerability, CVE-2007-0038 is recommend to be managed before CVE-2005-0344.

Table 6.5: Comparison of Relevancy Attributes from Example.

CVE	VR Rank	VR_{norm}	ω_{hits}	ω_{age}	VRscore
CVE-2005-0344	22	0.0118	0.8443	0.4	0.0039
CVE-2007-0038	9	0.1477	1.0000	0.6	0.0886

Table 6.6: Comparison of CVE-2005-0344 and CVE-2007-0038.

	CVE-2005-0344	CVE-2007-0038
Page	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-0344	http://www.microsoft.com/technet/security/bulletin/ms07-017.aspx
Basic information	1	1
Asset	- Webmail	- Operating System
Consequence	- Execute arbitrary files	- Denial of service
Security Violation	- Confidentiality	- Availability
Severity	- Low	- High
Vulnerability Type	- Directory traversal	- Stack-based buffer overflow
Technical Detail	1	1
Access	- Authenticated users	- Unauthorized
Architecture	-	-
Framework	-	-
Range	- Remote	- Remote
Platform	- 602LAN SUITE	- Microsoft Windows 2000 SP4
Exploit Detail	0	1
Exploit Phase	-	- Exploit
Attack Method	-	-
Exploit Type	-	- zero-day
Publicity	0	1
Alert Level	-	- Advisory
Damage	-	- Critical
Threat Level	-	- Severe
Virus Name	-	-
Remediation Detail	0	1
Official Update	-	-
Workaround	-	- MS07-017
Third-Party Software	-	-

6.5.2 Relevancy Quantification Service

We have developed the Vulnerability Relevancy Quantification Service (Wita et al.). The system is composed of Vulnerability Relevancy System and VR-Ranking webservice. Vulnerability Relevancy System periodically calculates VRscore for each CVE and the VRRanking webservice provides ranking for user-specific vulnerability by product, by system, and by operating system in order to help system administrator or software user to determine the risk level and prioritizing their work on securing the system.

6.6 VRscore Evaluation

To evaluate the Vulnerability Relevancy Quantification Framework, the top ranked vulnerabilities from established sources: (SANS, 2007), (Qualys, 2011), and (Jumrat-jaroenvanit and Teng-amnuay, 2008) are used as input to Context Sensitive Profile and the VRscore in our framework. Table 6.7 describes test dataset used in this research. Test dataset is composed of 3 different sources with 7 different types. We select SANS Top 20 list of 2007 because it is well-known in ranking vulnerability from impact. The report in 2007 is the last publicly available without subscription payment. The lists from SANS are composed of cross platform and windows related vulnerability (SANS, 2007). Another source is from Qualys top 10 vulnerability report in January 2011.

The lists from Qualys are windows related vulnerabilities, The *Top 10 External Vulnerabilities* are the most prevalent and critical vulnerabilities which have been identified on Internet facing systems. The *Top 10 Internal Vulnerabilities* show this information for systems and networks inside organization's firewalls (Qualys, 2011).

POA list divided vulnerability based on the development of lifecycle states. *Exploit* is vulnerability with available exploit code or script. *No exploit* is vulnerability which have no or less exploit code available. *Pseudo-zero-day* results from administrators not applying a particular patch even though the patch was released by vendor some time ago, and later become a highly publicized news. *Zero-day* indicates highly publicized news on attack before remediation is available.

From Figure 6.7, each test group has different relevancy trend. We would like to explain the result of those groups based on characteristic of dataset. *SANS wintop20* and *SANS crossplatform* contain vulnerabilities discovered between 2000 to 2007. VRscores

Table 6.7: Test Dataset

Information Source	Amount					Total
	2000-2005	2006	2007	2008	2009	
SANS Top 20 (SANS, 2007)						
Cross Platform	17	9	0	0	0	26
Windows	7	16	0	0	0	23
Qualys Top 10 (Qualys, 2011)						
External	16	0	0	5	2	23
Internal	0	0	3	27	29	59
Probability of Attack -POA (Jumratjaroenvanit and Teng-amnuay, 2008)						
Exploit	65	33	6	0	0	104
No exploit	67	19	5	0	0	91
Pseudo-zero-day	14	4	0	0	0	18
Zero-day	1	16	4	0	0	21

of CVE in *SANS wintop20* are mostly below 100, because the CVEs are listed as high impact in Windows system in the past. The official patches from vendor are fully available, reflected as high in remediation detail. The relevancy of these CVEs are degraded by age and the availability of remediation detail. *SANS crossplatform* contains vulnerabilities that affect multiple platforms. Although they are old vulnerabilities, some of them were mentioned in multiple software websites. Once the information was published in one software, it was also re-posted in other effected platform. For example, a vulnerability affecting in Linux kernel was used in multiple Linux distributions. Once exploit code or incident are available, it would widely effected among those system which use the same version of Linux kernel.

CVEs in *Qualys External* and *Internal* list are recently discovered between 2006 to 2009. The relevancy scores are distributed over a wide range. *External* list contains CVEs related to network or Internet connection, while *Internal* list needs firewall traversal. Mostly the *Internal* list contains CVEs from office software, e.g. Adobe acrobat, Flash player, and Internet explorer. Trend of relevancy scores in *Qualys Internal* list is surprisingly high. From SANS Top Cyber Risk Report, they pointed out that the zero-day exploitation target would be “File Format Vulnerability” which usually found in 3rd-party add-ons to popular and widely spread software suits like Microsoft Office Suite, Flash player, and Adobe reader (SANS, 2009). This is fully supported by our results.

CVE from POA list are Windows based vulnerability discovered between 2000 to 2007. *POA no exploit* list contains vulnerabilities which no exploit information avail-

able, *exploit* are those with exploit information available. *POA zeroday* contains CVE which have wide-spread of exploitation available before basic and remediation information available, while *POA pseudo zeroday* are CVEs that have wide-spread exploitation after remediation are available. From the result in Figure 6.7, we can use the same reason about the age of vulnerability as SANS wintop20 to explain why the relevancy score of *POA zeroday* and *pseudo-zeroday* are not very high. Microsoft provided auto update option for Windows system, *POA pseudo-zeroday* affects only those systems that have not applied patches, while *zeroday* list happened to have more relevancy trend because it is more wide spread.

The *POA exploit list* and the training data results in a normal distribution in relevancy trend. We can assume this as a normal vulnerability behavior. The relevancy trend reflects public interest of a particular vulnerability. Comparing this to the trend in Qualys External and Internal lists, the recent top ranked vulnerabilities tend to have high relevancy score.

6.7 Summary

This Chapter presents the analysis of the acquisition methodology of vulnerability relevancy factors introduced prior in Chapter 3.

On the assumption that relevancy of a vulnerability are related to public interest, search result, information age, and subcontext are used to evaluate the vulnerability relevancy. Web data mining technique is used to create Context Sensitive Profile based on ontology. EM clustering algorithm is used for analyzing the level of information distribution in Context Sensitive Profile in order to determine relevancy score. Next Chapter, we will present the research result, and analysis of reflection and each factor on vulnerability relevancy.

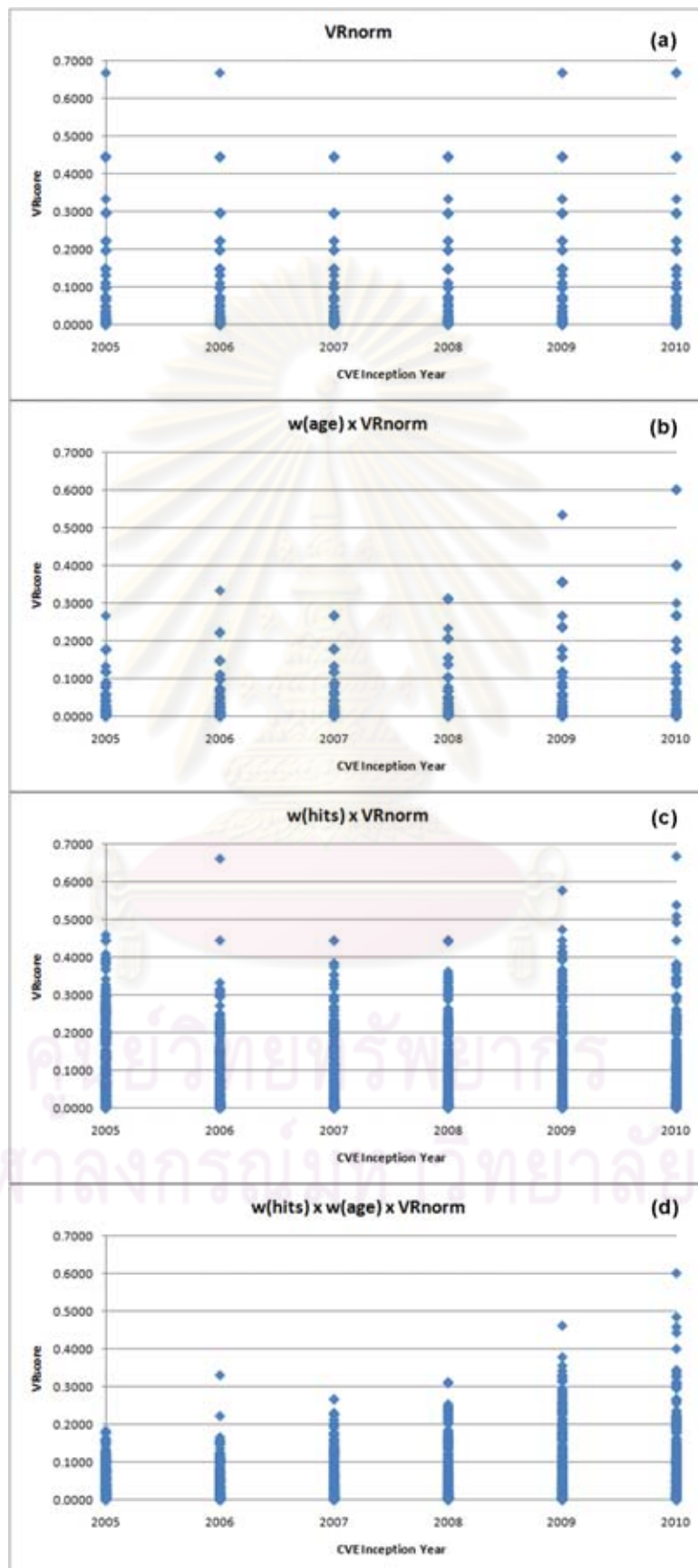


Figure 6.5: Variation of Vulnerability Relevancy Score from Different Attributes Used.

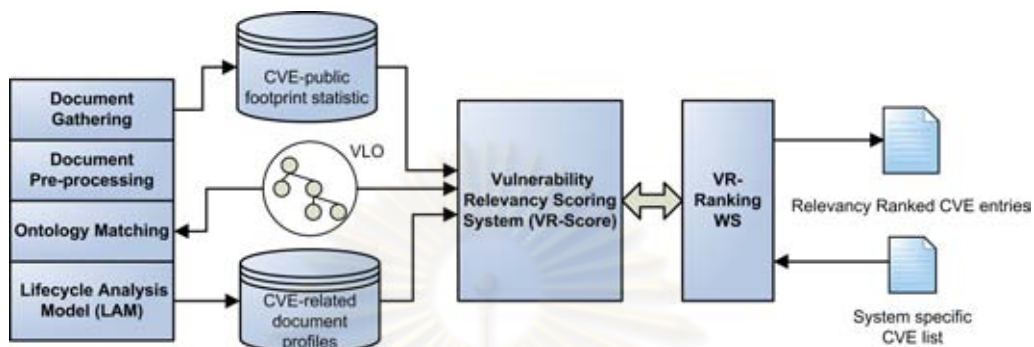


Figure 6.6: Relevancy Quantification Service.

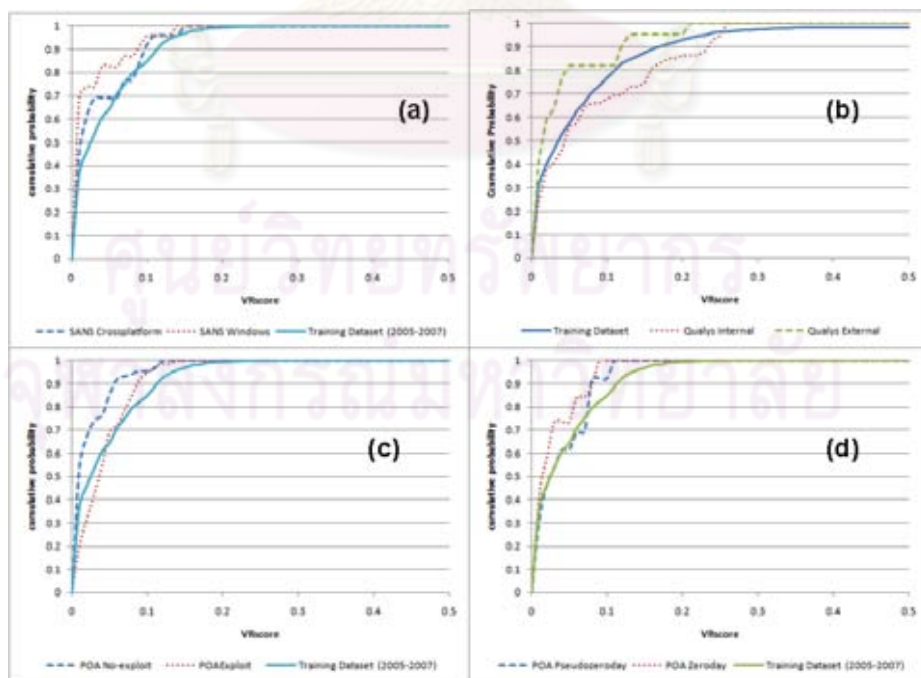


Figure 6.7: Comparison of Cumulative Probability of VRScore in Test Dataset

CHAPTER VII

RESEARCH RESULTS

This chapter describes the results of this research. Each of possible relevancy factors defined in Chapter 6 are discussed. This includes context type, information age, hits in public interest, and information source.

7.1 Information source

Figure 7.1 presents Top20 source of vulnerability information. The results in the graph are collected from top 10 search results of each CVE in training dataset. The most frequently listed source in search result is CVE website which constantly provide vulnerability standard name and description for most known vulnerability. The less frequent sources are composed of security websites, vulnerability databases, and vendor websites. This means the information used in this research usually gathered from official website of security related organizations. We can assume reliable information from these sources.

Nevertheless, some of individual websites, such as academic webpages or pages from blogspot are also captured due to specific vulnerability discussion and the limitation of page ranking in search service. In this research all sources are assigned with the same weight.



Figure 7.1: Distribution of Top 20 Information Source from Search Result.

7.2 Hits in Public Interest

Hits in Public Interest is considered as one of relevancy factors in our framework. On the assumption that the relevant vulnerability should have much public concern as reflected on the number of search result. Figure 7.2 depicts the accumulative probability distribution of hits in public interest of training dataset and test dataset that are considered high-impact vulnerabilities. From the test dataset characteristic in 6.7, vulnerability list from SANS and POA are usually incepted before 2005 to 2007 while vulnerabilities in Qualys lists are usually distributed between 2007 to 2009.

The results shows a high distribution of hits in Qualys external, POA exploit, POA pseudo-zero-day and POA no exploit, while information from SANS are more clustered and saturated around value 3. This means vulnerabilities from Qualys and POA vary in public interest, while vulnerabilities in SANS list have almost the same level in public interest. This is because the measurement of top vulnerabilities listed in SANS are based on impact and the vulnerability age is quite older than listed in Qualys. Older vulnerability age refers to older affected software and component which may already be obsolete and not interesting anymore.

POA-based classification depends on availability and order of exploit and remediation information that can change over time but does not consider the amount of information available. This results in vulnerabilities with same characteristic but different level of public interest being classified in the same group.

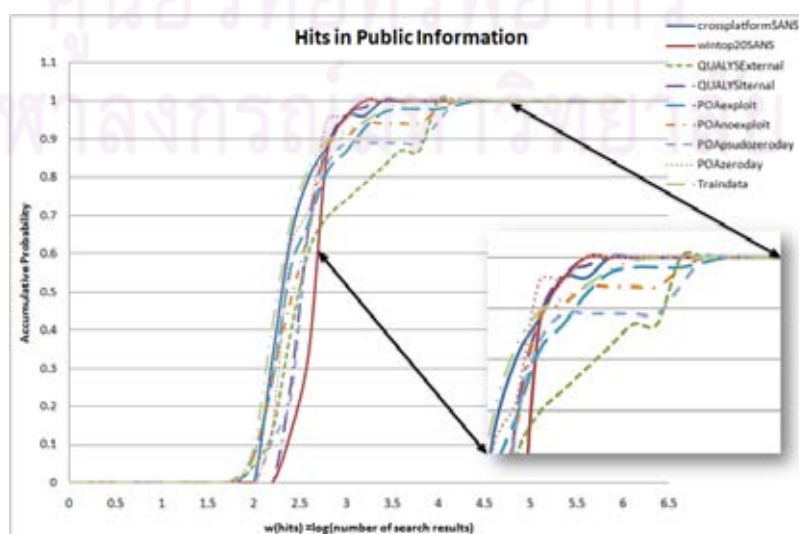


Figure 7.2: Cumulative Probability Distribution of Hits in Public Interest in Training Dataset and Test Dataset

7.3 VRscore and Risk Rank

Vulnerability relevancy in our definition is represented by public interest and data distribution in the context of vulnerability lifecycle, while the top ranked vulnerabilities listed by security advisory are focused on high severity or impact to the attacked system while our Vulnerability Relevancy Scoring System focused on public interest of a particular vulnerability.

We calculate VRscore on ten random Windows related vulnerabilities. Table 7.1 shows VRscore and attributes. Each vulnerability has different level of ω_{age} , ω_{hits} , and R_{norm} which resulted in different level of VRscore. In our definition VRscore defined for likelihood level of a vulnerability based on observing public interest. From Risk management principle, $Risk = Severity \times Likelihood$, VRscore can be used in conjunction with CVSS as likelihood and severity level of a vulnerability to identify risk level of a particular vulnerability. Table 7.2 shows ranked risk score based on severity and VRscore.

An analysis of the CVE in sample list indicates that most CVE listed as high risk were from desktop application softwares, e.g. Adobe acrobat, Flash player, and Internet explorer. These are reflected from high consideration in application software vulnerabilities from public interests. The supportive reasons are from SANS Top Cyber Risk Report in 2009 (SANS, 2009) and Security trends for 2010 (SANSInstitute, 2011). They pointed out that vulnerability problems are moving from operating system to common libraries and application softwares. The number of vulnerabilities in software are increasing while the availability of remediations are still slower than those in operating system.

Table 7.1: VRscores and Attributes of Sample CVE

Vulnerability	ω_{age}	ω_{hits}	R_{norm}	VRscore
CVE-2010-0555	0.9	0.8213	0.2959	0.2187
CVE-2009-0238	0.8	0.9121	0.2959	0.2159
CVE-2009-0119	0.8	0.7999	0.2218	0.1419
CVE-2009-0001	0.8	1.0000	0.1477	0.1182
CVE-2008-0407	0.7	1.0000	0.2959	0.2072
CVE-2009-0008	0.8	0.9968	0.2959	0.0587
CVE-2010-0718	0.9	0.7897	0.1477	0.1050
CVE-2010-0162	0.9	0.8421	0.0736	0.0558
CVE-2010-0654	0.9	0.7713	0.0736	0.0511
CVE-2010-0107	0.9	1.0000	0.0118	0.0107



Table 7.2: Sample of Risk Ranked Vulnerability based on Severity and Relevancy

Vulnerability	Affected software	Severity	VRscore	Risk	Rank
CVE-2010-0555	Internet Explorer	High	0.2187	0.2034	1
CVE-2009-0238	MS Excel	High	0.2159	0.2008	2
CVE-2009-0119	Windows XP	High	0.1419	0.1419	3
CVE-2009-0001	QuickTime	High	0.1182	0.1099	4
CVE-2008-0407	HFS	Medium	0.2072	0.1036	5
CVE-2009-0008	QuickTime	High	0.0587	0.0446	6
CVE-2010-0718	Windows Media Player	Medium	0.1050	0.0452	7
CVE-2010-0162	Firefox	Medium	0.0558	0.0240	8
CVE-2010-0654	Firefox	Medium	0.0511	0.0220	9
CVE-2010-0107	ActiveX	High	0.0107	0.0100	10

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER VIII

CONCLUSIONS

This dissertation proposed the framework for quantifying vulnerability relevancy in order to optimize security level of the system with limited administrative resource. Vulnerability relevancy is defined based on public interest and lifecycle states including number of information available, age of information, and lifecycle subcontext. Relevancy factors from public interest have been analyzed. Vulnerability Relevancy Ranking framework and its necessary components have been defined.

Vulnerability Lifecycle Ontology (VLO) is defined, developed and evaluated to describe the relationship between vulnerability lifecycle states and information context. We introduced the concept of subcontext in ontology and the procedure to create Context Sensitive Profile to represent topic in various context. Public interest of a vulnerability acquisition process are conducted and represent as Context Sensitive Profile of a particular vulnerability.

We performed the experiment on 3000 randomly chosen vulnerabilities discovered from 2006 to 2010 to analyze the behavior of public interest. Context-based Relevancy metric is defined based on the experiment. Vulnerability Relevancy Score is calculated based on predefined relevancy factors and compared by established top vulnerability lists including SANS Top 20 vulnerability (SANS, 2007), Qualys Top 10 vulnerability (Qualys, 2011), and Probability of Attack Profile (POA) (Jumratjaroenvanit and Teng-amnuay, 2008).

From the experiment on eight different lists, although the vulnerability from SANS Top20, Qualys Top10, POA pseudo-zero-day, and POA zero-day are considered as notable high vulnerability with widespread impact, the relevancy scores vary. This is mainly because of vulnerability aging, availability of official remediation, and the number of effected platform. Moreover, we also found that the top relevant vulnerability are application services and client-side applications such as Java Runtime Environment, Internet explorer, and Microsoft Office Suite in conforming SANS Top Cyber Security Risks Re-

port in 2009 (SANs, 2009).

From the analysis of vulnerability relevancy and the high impact vulnerability from established lists, we can conclude the type of risk based on severity and vulnerability relevancy as shown in Table 8.1. Vulnerability Relevancy metric can be used in conjunction with vulnerability severity evaluation in order to define risk level from vulnerability.

High severity vulnerability with high relevancy are consider as urgent risk and needs to be monitored and fixed as soon as possible, while high severity vulnerability that has low relevancy level can wait . Meanwhile, vulnerability with high relevancy but low in severity may need to be watched, since low severity may not cause much trouble to the system but is annoying so it appear on public interest.

Table 8.1: Risk Level of Vulnerability

Relevancy	Severity		
	High	Medium	Low
High	Urgent Risk	Moderate Risk	Unnecessary Risk
Medium	Moderate Risk	Risk	Low Risk
Low	Unnecessary cost for admin	Low Risk	Negligible Risk

8.1 Discussion and Suggestion

The experiments were conducted based on public information from Google search service. Since information available changes everyday and the page ranking in search service is relies on multiple factors such as the number of fan-in and fan-out or the number of query (Google, 2011). It is possible to use this to reflect the relevancy of a vulnerability.

From the empirical study, formal websites that constantly publish vulnerability information reflect more reliable information about remediation than blogs, or personal websites. Meanwhile, technical discussion websites and personal blogs contain more technical detail. Behavior of information source should further be studied and analyzed in order to provide reliability evaluation of information providers.

The age of information used in this research is derived from the age of vulnerability. This factor can be improved by using different search result within the varies with time. These methodology also have to consider duplicated pages.

The vulnerability lifecycle ontology contains vulnerability concepts and relation-

ships derived from vulnerability standards, vulnerability taxonomy, and security websites. Some structural, standard concepts are mostly static information while new virus names and official patch names appear over time. One possible suggestion for further research is an automate ontology enhancement and pruning. This concerns confliction on concepts and relationships in ontology, cyclic reference, and duplicated information.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

References

- Adafre, S.F., Jijkoun, V., and Rijke, M. Link-based vs. content-based retrieval for question answering using wikipedia. In Web Information Quality Assessment Framework- Cross Language Evaluation Forum, : 537–540,. Springer-Verlag Berlin Heidelberg. 2006.
- Alani, H., and Brewster, C. Metrics for ranking ontologies. In Proceedings of 4th Int. EON Workshop, 15th Int. World Wide Web Conf., Edinburgh. 2006.
- Arbaugh, W.A. A patch in nine saves time? Computer 37:82–83. 2004.
- Arbaugh, W.A., Fithen, W.L., and McHugh, J. Windows of vulnerability: a case study analysis. Computer 33,(12):52–59. 2000.
- Arora, A., Krishnan, R., Nandkumar, A., Telang, R., and Yang, Y. Impact of vulnerability disclosure and patch availability-an empirical analysis. In Third Workshop on the Economics of Information Security. 2004.
- Arora, A., Nandkumar, A., and Telang, R. Does information security attack frequency increase with vulnerability disclosure? an empirical analysis. Information Systems Frontiers 8,(5):350–362. 2006.
- Baxter, I.D. and Pidgeon, C.W. Software change through design maintenance. In Proceedings of International Conference on Software Maintenance, : 250 –259. 1997.
- Bechhofer, S. and al., et . 2004. Owl web ontology language reference. Technical report, W3C. Available from: <http://www.w3.org/TR/owl-ref/>. [2011, January, 30].
- Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. Data driven ontology evaluation. In Proceedings of International Conference on Language Resources and Evaluation, Lisbon, Portugal. 2004.
- Browne, H.K. , Arbaugh, W.A. , McHugh, J., and Fithen, W.L. A trend analysis of exploitations. In Proceedings of 2001 IEEE Symposium on Security and Privacy, : 214 –229, California, USA. 2001.

- Cooley, R., Mobasher, B., and Srivastava, J. Web mining: Information and pattern discovery on the world wide web. In Proceedings of the 9th International Conference on Tools with Artificial Intelligence. IEEE Computer Society. 1997.
- dAmato, C., Fanizzi, N., and Esposito, F. Distance-based classification in owl ontologies. In Knowledge-Based Intelligent Information and Engineering Systems, : 656–661. 2008.
- Dantu, R., Loper, K., and Kolan, P. Risk management using behavior based attack graphs. In Proceedings of International Conference on Information Technology: Coding and Computing, : 445 – 449, Nevada, USA. 2004.
- Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via them algorithm. Journal of the Royal Statistical Society 39,(1):1–38. 1977.
- Deng, S., and Peng, H. Document classification based on support vector machine using a concept vector model. In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society. 2006.
- Frei, S., May, M., Fiedler, U., and Plattner, B. Large-scale vulnerability analysis. In Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, : 131–138. 2006.
- Frei, S., Schatzmann, D., Plattner, B., and Trammel, B. Modelling the security ecosystem - the dynamics of (in)security. In Workshop on the Economics of Information Security (WEIS), Cambridge, UK. 2009.
- Google Inc. Google trends. [Online], 2011. Available from: <http://www.google.com/trends>. [2011, January, 30].
- Google Inc. Google soap search api. [Online], 2007. Available from: <http://code.google.com/apis/soapsearch/reference.html>. [2011, January, 30].
- Grossman, D.A, and Frieder, O. Information Retrieval: Algorithms and Heuristics, volume 15 of The Information Retrieval Series. Springer Berlin Heidelberg. 2004.
- Gulla, J.A., and Brasethvik, T. A hybrid approach to ontology relationship learning. In Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, NLDB '08, : 79–90, Berlin, Heidelberg. 2008.

- He, Y., Chen, W., Yang, M., and Peng, W. Ontology based cooperative intrusion detection system. In Network and Parallel Computing, : 419–426. 2004.
- Hitwise. Top 20 sites and engines. [Online], 2009. Available from: [www.hitwise.com/us/datacenter /main/dashboard-10133.html](http://www.hitwise.com/us/datacenter/main/dashboard-10133.html). [2011, January, 30].
- Hogan, C.B. Protection imperfect: The security of some computing environments. Operating System Review 22,(3):7–27. 1988.
- Hotho, A., Staab, S., and Maedche, A. Ontology-based text clustering. In Proceedings of the IJCAI-2001 Workshop Text Learning: Beyond Supervision. 2001.
- Hyunchul, J., and K.M., Yashwant. A framework for software security risk evaluation using the vulnerability lifecycle and cvss metrics. In Proceedings of International Workshop on Risk and Trust in Extended Enterprises, RTEE2010, : 430–434. 2010.
- Jaquith, A. Security Metrics: Replacing Fear, Uncertainty, and Doubt. Addison-Wesley Professional. 2007.
- Jha, S. and Wing, J. Survivability analysis of networked systems. In Proceedings of the 23rd International Conference on Software Engineering. 2001.
- Jindal, N. and Liu, B. Opinion spam and analysis. In Proceedings of the international conference on Web search and web data mining. ACM. 2008.
- Jiwnani, K. and Zelkowitz, M. Maintaining software with a security perspective. In Proceedings of 18th IEEE International Conference on Software Maintenance. 2002.
- Jumratjaroenvanit, A., and Teng-amnuay, Y. Probability of attack based on system vulnerability life cycle. In Proceedings of the 2008 International Symposium on Electronic Commerce and Security, ISECS '08, : 531–535. 2008.
- Kannan, K. and Telang, R. An economic analysis of market for software vulnerabilities. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences, HICSS '04. 2004.

- Kim, A., Luo, J., and Kang, M. Security ontology for annotating resources. In On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE, : 1483–1499. 2005.
- KoHyung, J., E., Lee, J. and Lee, J.W. Ontology-based context modelling and reasoning for u-healthcare. IEICE TRANSACTIONS on Information and Systems E90-D ,(8):1262–1270. 2007.
- Lai, Y.P., and Hsia, P.L. Using the vulnerability information of computer systems to improve the network security. Computer Communications 30,(9):2032–2047. 2007.
- Landwehr, C.E. Formal models for computer security. ACM Computing Survey 13,(3). 1981.
- Lee, S.C., and Davis, L.B. Learning from experience: operating system vulnerability trends. IT Professional 5:17–24. 2003.
- Liu, B. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Data-Centric Systems and Applications. Springer Berlin Heidelberg. 2007.
- Longstaff, T. Cert experince with security problems in software. In Software Security - How Should We Make Software Secure? 2003.
- MacKay, B. Lifespan of software. [Online], 2006. Available from: <http://stackoverflow.com/questions/360297/lifespan-of-software-how-often-do-you-expect-to-do-start-from-scratch>. [2011, January, 30].
- Meier, J.D., Mackman, A., Dunner, M., Vasireddy, A., and Escamilla, A., R andMurukan. Improving web application security: Threats and countermeasures. [Online], 2003. Available from: <http://msdn.microsoft.com/en-us/library/ff648644.aspx>. [2011, January, 30].
- Microsoft. Microsoft security response center security bulletin severity rating system. [Online], 2002. Available from: <http://www.microsoft.com/technet/security/bulletin/rating.msp>. 2011, January, 30.

- Miles, S., and Bechhofer, A. Skos simple knowledge organization system reference. [Online], 2009. Available from: <http://www.w3.org/TR/skos-reference/>. [2011, January, 30].
- Mishne, G. Multiple ranking strategies for opinion retrieval in blogs. In Proceeding of Text REtrieval Conference, TREC'06. NIST. 2006.
- Mitre. Common attack pattern enumeration and classification. [Online], 2008. Available from: capec.mitre.org. [2010, December, 30].
- Mitre. Common platform enumeration. [Online], 2009. Available from: cpe.mitre.org. [2010, December, 30].
- Mitre. Common vulnerability and exposure. [Online], 1999. Available from: cve.mitre.org. [2010, December, 30].
- Mitre. Common weakness enumeration. [Online], 2007a. Available from: cwe.mitre.org. [2010, December, 30].
- Mitre. Making security measurable. [Online], 2007b. Available from: msm.mitre.org. [2010, December, 30].
- Moreira, E.S., Martimiano, L.A.F., and Brandao, M.C., A.J.S.and Bernardes. Ontologies for information security management and governance. Information Management & Computer Security 16,(2):150–165. 2008.
- Neumann, P.G., and Parker, D.B. A summary of computer misuse techniques. In Proceedings of 12th National Computer Security Conference, Baltimore, MD, : 396–406. 1989.
- NIST. Common vulnerability scoring system. [Online], 2007. Available from: <http://nvd.nist.gov/cvss.cfm?version=2>. [2010, December, 30].
- NIST. National vulnerability database. [Online], 1999. Available from: nvd.nist.org. [2010, December, 30].
- OSVDB. The open source vulnerability database. [Online], 2008. Available from: www.osvdb.org. [2011, January, 30].

- Pinkston, J., Undercoffer, A., J. and Joshi, and Finin, T. A target-centric ontology for intrusion detection. In Proceedings of The 18th International Joint Conference on Artificial Intelligence. 2003.
- Princeton University. About wordnet. [Online]. Available from: <http://wordnet.princeton.edu>.
- Qamra, A., Tseng, B., and Chang, E.Y. Mining blog stories using community-based and temporal clustering. In Proceedings of the 15th ACM international conference on Information and knowledge management, CIKM '06, : 58–67, New York, NY, USA. ACM. 2006.
- Qualys. The laws of vulnerabilities: Six axioms for understanding risk (whitepaper). [Online], 2006. Available from: <http://www.qualys.com/docs/Laws-Report.pdf>. [2011, January, 30].
- Qualys. Top 10 vulnerabilities. [Online], 2011. Available from: <http://www.qualys.com/research/top10/>. [2011, January, 30].
- Ranum, M. A taxonomy of internet attacks. Tutorial Notes 1996,(Apr 20). 1996.
- Raskin, V., Hempelmann, C.F., Triezenberg, K.E., and Nirenburg, S. Ontology in information security: a useful theoretical foundation and methodological tool. In Proceedings of the 2001 workshop on New security paradigms, NSPW '01. ACM. 2001.
- Rijsbergen, C.J., Robertson, S.E., and Porter, M.F. New models in probabilistic information retrieval. In British Library Research and Development Report, volume 5587. 1980.
- Saltzer, J.H., and Schroeder, M.D. The protection of information in computer systems. Proceedings of the IEEE 63,(9):1278–1308. 1975.
- SANs. Top cyber security risks - zero-day vulnerability trends. [Online], 2009. Available from: <http://www.sans.org/top-cyber-security-risks/zero-day.php>. [2011, January, 30].
- SANs. 2006 sans top 20 spring update technical details. [Online], 2007. Available from: http://www.sans.org/top20/2005/spring_2006_detail.php. [2011, January, 30].

- SANS. Sans top security risks. [Online], 2007. Available from: <http://www.sans.org/top-cyber-security-risks/>. [2011, January, 30].
- SANS Institute. Security predictions for 2010. [Online], 2011. Available from: <http://www.sans.edu/research/security-laboratory/article/2010-predictions>. [2011, January, 30].
- Simon, J. Skosed - thesaurus editor for the semantic web. [Online], 2009. Available from: <http://code.google.com/p/skoseditor/>. [2010, December, 30].
- Stanford University. Protege-ontology editor software. [Online], 2007. Available from: <http://protege.stanford.edu/>. [2011, January, 30].
- Tupper, M., and Zincir-Heywood, A.N. Ver-bility security metric: A network security analysis tool. In Proceedings of the Third International Conference on Availability, Reliability and Security, : 950–957, Washington, DC, USA. 2008.
- Varelas, G., Voutsakis, E., Raftopoulou, E.M., Petrakis, P. and Milios, E.E. Semantic similarity methods in wordnet and their application to information retrieval on the web. In Proceedings of the 7th annual ACM international workshop on Web information and data management, WIDM '05, : 10–16, New York, NY, USA. ACM. 2005.
- W3C. 2004. Skos concept semantics patterns for working with skos and owl. Technical report, W3C. Available from: <http://www.w3.org/TR/owl-ref/>. [2011, January, 30].
- Wang, J., Zhang, F., and Xia, M. Temporal metrics for software vulnerabilities. In Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead, CSIIRW '08, : 44:1–44:3. 2008.
- Wita, R., and Teng-Amnuay, Y. Vulnerability profile for linux. In Proceedings of the 19th International Conference on Advanced Information Networking and Applications - Volume 1, AINA '05, : 953–958, Washington, DC, USA. IEEE Computer Society. 2005.

- Wita, R., Kuapongthai, S., Techaveerapong, P., and Teng-Amnuay, Y. Vulnerability relevancy service. [Online]. Available from: <http://isel.cp.eng.chula.ac.th/asvrrs>. [2011, March, 30].
- Wu, W., Yip, E., Ray, P., and Yiu, F. Integrated vulnerability management system for enterprise networks. In Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service, EEE '05, : 698–703, Washington, DC, USA. IEEE Computer Society. 2005.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



APPENDIX

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

RANKED VULNERABILITY RELEVANCY SCORES

Table A.1: Ranked of Lifecycle attributes and Vulnerability Relevancy Scores

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
1	high	high	high	high	low	1
2	medium	high	high	high	low	0.6665
	high	medium	high	high	low	0.6665
	high	high	medium	high	low	0.6665
	high	high	high	medium	low	0.6665
	high	high	high	high	medium	0.6665
3	medium	medium	high	high	low	0.4442
	medium	high	medium	high	low	0.4442
	medium	high	high	medium	low	0.4442
	medium	high	high	high	medium	0.4442
	high	medium	medium	high	low	0.4442
	high	medium	high	medium	low	0.4442
	high	medium	high	high	medium	0.4442
	high	high	medium	medium	low	0.4442
	high	high	medium	high	medium	0.4442
	high	high	high	medium	medium	0.4442
4	low	high	high	high	low	0.3330
	high	low	high	high	low	0.3330
5	medium	medium	medium	high	low	0.2959
	medium	medium	high	medium	low	0.2959
	medium	medium	high	high	medium	0.2959
	medium	high	medium	medium	low	0.2959
	medium	high	medium	high	medium	0.2959
	medium	high	high	medium	medium	0.2959
	high	medium	medium	medium	low	0.2959
	high	medium	medium	high	medium	0.2959
	high	medium	high	medium	medium	0.2959

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	high	high	medium	medium	medium	0.2959
6	low	medium	high	high	low	0.2218
	low	high	medium	high	low	0.2218
	low	high	high	medium	low	0.2218
	low	high	high	high	medium	0.2218
	medium	low	high	high	low	0.2218
	high	low	medium	high	low	0.2218
	high	low	high	medium	low	0.2218
	high	low	high	high	medium	0.2218
7	medium	medium	medium	medium	low	0.1971
	medium	medium	medium	high	medium	0.1971
	medium	medium	high	medium	medium	0.1971
	medium	high	medium	medium	medium	0.1971
	high	medium	medium	medium	medium	0.1971
8	high	high	low	high	low	324
	high	high	high	low	low	324
	high	high	high	high	high	324
9	low	medium	medium	high	low	0.1477
	low	medium	high	medium	low	0.1477
	low	medium	high	high	medium	0.1477
	low	high	medium	medium	low	0.1477
	low	high	medium	high	medium	0.1477
	low	high	high	medium	medium	0.1477
	medium	low	medium	high	low	0.1477
	medium	low	high	medium	low	0.1477
	medium	low	high	high	medium	0.1477
	high	low	medium	medium	low	0.1477
	high	low	medium	high	medium	0.1477
	high	low	high	medium	medium	0.1477

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
10	medium	medium	medium	medium	medium	0.1312
11	low	low	high	high	low	0.1107
	medium	high	low	high	low	0.1107
	medium	high	high	low	low	0.1107
	medium	high	high	high	high	0.1107
	high	medium	low	high	low	0.1107
	high	medium	high	low	low	0.1107
	high	medium	high	high	high	0.1107
	high	high	low	medium	low	0.1107
	high	high	low	high	medium	0.1107
	high	high	medium	low	low	0.1107
	high	high	medium	high	high	0.1107
	high	high	high	low	medium	0.1107
	high	high	high	medium	high	0.1107
12	low	medium	medium	medium	low	0.0983
	low	medium	medium	high	medium	0.0983
	low	medium	high	medium	medium	0.0983
	low	high	medium	medium	medium	0.0983
	medium	low	medium	medium	low	0.0983
	medium	low	medium	high	medium	0.0983
	medium	low	high	medium	medium	0.0983
	high	low	medium	medium	medium	0.0983
13	low	low	medium	high	low	0.0736
	low	low	high	medium	low	0.0736
	low	low	high	high	medium	0.0736
	medium	medium	low	high	low	0.0736
	medium	medium	high	low	low	0.0736
	medium	medium	high	high	high	0.0736
	medium	high	low	medium	low	0.0736
	medium	high	low	high	medium	0.0736

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	medium	high	medium	low	low	0.0736
	medium	high	medium	high	high	0.0736
	medium	high	high	low	medium	0.0736
	medium	high	high	medium	high	0.0736
	high	medium	low	medium	low	0.0736
	high	medium	low	high	medium	0.0736
	high	medium	medium	low	low	0.0736
	high	medium	medium	high	high	0.0736
	high	medium	high	low	medium	0.0736
	high	medium	high	medium	high	0.0736
	high	high	low	medium	medium	0.0736
	high	high	medium	low	medium	0.0736
	high	high	medium	medium	high	0.0736
14	low	medium	medium	medium	medium	0.0654
	medium	low	medium	medium	medium	0.0654
15	low	high	low	high	low	0.0551
	low	high	high	low	low	0.0551
	low	high	high	high	high	0.0551
	high	low	low	high	low	0.0551
	high	low	high	low	low	0.0551
	high	low	high	high	high	0.0551
16	low	low	medium	medium	low	0.0489
	low	low	medium	high	medium	0.0489
	low	low	high	medium	medium	0.0489
	medium	medium	low	medium	low	0.0489
	medium	medium	low	high	medium	0.0489
	medium	medium	medium	low	low	0.0489
	medium	medium	medium	high	high	0.0489
	medium	medium	high	low	medium	0.0489
	medium	medium	high	medium	high	0.0489

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	medium	high	low	medium	medium	0.0489
	medium	high	medium	low	medium	0.0489
	medium	high	medium	medium	high	0.0489
	high	medium	low	medium	medium	0.0489
	high	medium	medium	low	medium	0.0489
	high	medium	medium	medium	high	0.0489
17	low	medium	low	high	low	0.0365
	low	medium	high	low	low	0.0365
	low	medium	high	high	high	0.0365
	low	high	low	medium	low	0.0365
	low	high	low	high	medium	0.0365
	low	high	medium	low	low	0.0365
	low	high	medium	high	high	0.0365
	low	high	high	low	medium	0.0365
	low	high	high	medium	high	0.0365
	medium	low	low	high	low	0.0365
	medium	low	high	low	low	0.0365
	medium	low	high	high	high	0.0365
	high	low	low	medium	low	0.0365
	high	low	low	high	medium	0.0365
	high	low	medium	low	low	0.0365
	high	low	medium	high	high	0.0365
	high	low	high	low	medium	0.0365
	high	low	high	medium	high	0.0365
18	low	low	medium	medium	medium	0.0324
	medium	medium	low	medium	medium	0.0324
	medium	medium	medium	low	medium	0.0324
	medium	medium	medium	medium	high	0.0324
18	high	high	low	low	low	0.0273
	high	high	low	high	high	0.0273

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	high	high	high	low	high	0.0273
20	low	medium	low	medium	low	0.0242
	low	medium	low	high	medium	0.0242
	low	medium	medium	low	low	0.0242
	low	medium	medium	high	high	0.0242
	low	medium	high	low	medium	0.0242
	low	medium	high	medium	high	0.0242
	low	high	low	medium	medium	0.0242
	low	high	medium	low	medium	0.0242
	low	high	medium	medium	high	0.0242
	medium	low	low	medium	low	0.0242
	medium	low	low	high	medium	0.0242
	medium	low	medium	low	low	0.0242
	medium	low	medium	high	high	0.0242
	medium	low	high	low	medium	0.0242
	medium	low	high	medium	high	0.0242
	high	low	low	medium	medium	0.0242
	high	low	medium	low	medium	0.0242
	high	low	medium	medium	high	0.0242
21	low	low	low	high	low	0.0180
	low	low	high	low	low	0.0180
	low	low	high	high	high	0.0180
	medium	high	low	low	low	0.0180
	medium	high	low	high	high	0.0180
	medium	high	high	low	high	0.0180
	high	medium	low	low	low	0.0180
	high	medium	low	high	high	0.0180
	high	medium	high	low	high	0.0180
	high	high	low	low	medium	0.0180
	high	high	low	medium	high	0.0180

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	high	high	medium	low	high	0.0180
22	low	medium	low	medium	medium	0.0160
	low	medium	medium	low	medium	0.0160
	low	medium	medium	medium	high	0.0160
	medium	low	low	medium	medium	0.0160
	medium	low	medium	low	medium	0.0160
	medium	low	medium	medium	high	0.0160
23	low	low	low	medium	low	0.0118
	low	low	low	high	medium	0.0118
	low	low	medium	low	low	0.0118
	low	low	medium	high	high	0.0118
	low	low	high	low	medium	0.0118
	low	low	high	medium	high	0.0118
	medium	medium	low	low	low	0.0118
	medium	medium	low	high	high	0.0118
	medium	medium	high	low	high	0.0118
	medium	high	low	low	medium	0.0118
	medium	high	low	medium	high	0.0118
	medium	high	medium	low	high	0.0118
	high	medium	low	low	medium	0.0118
	high	medium	low	medium	high	0.0118
	high	medium	medium	low	high	0.0118
24	low	high	low	low	low	0.0087
	low	high	low	high	high	0.0087
	low	high	high	low	high	0.0087
	high	low	low	low	low	0.0087
	high	low	low	high	high	0.0087
	high	low	high	low	high	0.0087
25	low	low	low	medium	medium	0.0077
	low	low	medium	low	medium	0.0077

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	low	low	medium	medium	high	0.0077
	medium	medium	low	low	medium	0.0077
	medium	medium	low	medium	high	0.0077
	medium	medium	medium	low	high	0.0077
26	low	medium	low	low	low	0.0057
	low	medium	low	high	high	0.0057
	low	medium	high	low	high	0.0057
	low	high	low	low	medium	0.0057
	low	high	low	medium	high	0.0057
	low	high	medium	low	high	0.0057
	medium	low	low	low	low	0.0057
	medium	low	low	high	high	0.0057
	medium	low	high	low	high	0.0057
	high	low	low	low	medium	0.0057
	high	low	low	medium	high	0.0057
	high	low	medium	low	high	0.0057
27	high	high	low	low	high	0.0041
28	low	medium	low	low	medium	0.0036
	low	medium	low	medium	high	0.0036
	low	medium	medium	low	high	0.0036
	medium	low	low	low	medium	0.0036
	medium	low	low	medium	high	0.0036
	medium	low	medium	low	high	0.0036
29	low	low	low	low	low	0.0026
	low	low	low	high	high	0.0026
	low	low	high	low	high	0.0026
	medium	high	low	low	high	0.0026
	high	medium	low	low	high	0.0026
30	low	low	low	low	medium	0.0015
	low	low	low	medium	high	0.0015

Continued on next page

Table A.1 – cont.

Rank	Basic Information	Technical Detail	Exploit Detail	Publicity	Remediation	VRscore
	low	low	medium	low	high	0.0015
	medium	medium	low	low	high	0.0015
31	low	high	low	low	high	0.0010
	high	low	low	low	high	0.0010
32	low	medium	low	low	high	0.0005
	medium	low	low	low	high	0.0005
33	low	low	low	low	high	0.0000

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Biography

Ratsameetip Wita was born in Lampang, Thailand, in September, 1980. She received B.Sc., in Applied Computer Science, from King Mongkut Institute of Technology North Bangkok, Thailand, in 2000. She received M.Sc., in Computer Science, from Chulalongkorn University, Thailand, in 2003. Her master degree has been supervised by Dr. Yunyong Teng-amnuay, as well as her Ph.D.

During her Ph.D study, she has got scholarships from Higher Education Commission of Thailand between 2007-2009 and 90th Year Chulalongkorn research scholarship. She has two cooperate researches with professor Saurabh Bagchi at School of Electrical and Computer Engineering, Purdue University, USA. (August 2008 - April 2009) and professor Nigel Collier at National Institute of Informatics (NII), Japan (August 2010-October 2010).

She also has two publications related to this dissertation in International conference during her study. The paper titled “*Ontology for Vulnerability Lifecycle*” by Ratsameetip Wita, Nattanatch Jiamnapanon, and Yunyong Teng-amnuay in the proceeding of IEEE International Symposium on Intelligent Information Technology and Security Informatics 2010 (IITSI 2010), Jingangshan, China, April 2010, and the paper titled “*Ontology-Based Document Profile for Vulnerability Relevancy Analysis*” by Ratsameetip Wita and Yunyong Teng-amnuay in the proceeding of 10th WSEAS International Conference on Applied Computer Science (ACS' 10), Iwate, Japan, October 2010.

Her field of interest includes various topics in Vulnerability quantification, Security management, Knowledge Engineering and Ontology