Prediction and Classification of Web Application Attacks using Vulnerability Ontology

P. Salini
Pondicherry Engineering College
Puducherry, India

J. Shenbagam
Pondicherry Engineering College
Puducherry, India

ABSTRACT
Web application security is the major security concern for e-business and information sharing communities. Research showed that more than 75% attacks are being deployed at application layer and almost 90% applications are vulnerable to the attacks. This is due to the avoidance of security requirements during implementation by the developer because they are not trained on solving security issues and often need to depend on security experts.

In this paper, an approach for effective defenses against the application level attacks is proposed. The proposed system is an ontology based system that can predict and classify web application attacks. The system effectively stores threat, vulnerability and attack information. The attacks can be predicted by analyzing vulnerability and threats. The attacks are classified based on severity level of the attacks on security goals. Moreover, the system also provides suggestion for prevention and countermeasure to the predicted attacks, thereby assisting the developers in developing secure web applications. The results were promising when compared to the conventional method of knowledge base.

General Terms
Information Security, Web Application Security, Ontology, Information Retrieval

Keywords
Ontology, Attacks, Vulnerabilities, Threats, Security Measures, Security Goals, Web application

1. INTRODUCTION
Nowadays information sharing is rapidly increased in all around the world through the use of web application and web services, thereby efficiency of e-business is enlarged. Not only increase the throughput of the e-business and also cyber-threats are growing due to the usage of the web, in turn affecting the security goals such as Confidentiality, Integrity, and Availability [1, 2].

Various efforts have been taken to control attacks with different security mechanisms such as IDS, scanners and web application firewalls, but code level security has more importance to develop vulnerable free application. In the early stage of development and testing, developer must know the information about threats, vulnerabilities and attacks profile, so that they can easily identify the attack and mitigate them. Thereby developing secure and well protected web applications. Moreover, the dependency on security experts to check the web application security can also be minimized.

Security ontology [12] which exists provides taxonomy for threats, vulnerabilities and attacks but lacks to infer the knowledge to predict the attacks and does not classify attacks.

In order to solve these issues, the proposed approach analyses the web application threats and vulnerabilities that may be exploited by the attacks. The proposed approach uses SWRL rules and inference process to predict the attacks with respect to the vulnerabilities. The predicted attacks are classified based on their severity level and the system also suggests prevention and mitigation methods.

The paper makes the following contributions:

- A method to predict and classify the web application attacks. Our solution is an ontology based approach that specifies the web application attacks using the context of consequences, threats and vulnerability.
- The proposed system is capable of predicting sophisticated attacks effectively and efficiently by analyzing the vulnerability that may be exploited.
- The predicted the web application attacks are classified based on their severity level.
- The proposed system also gives suggestion for preventing and mitigating the attacks.

The rest of the paper is organized as follows. Section 2 briefs on the related work. Section 3 discusses the architecture of the proposed work and Section 4 presents about the experiments carried and the system evaluation. The results have been discussed followed by a conclusion and future work.

2. RELATED WORK
Undercoffer et al. [3] proposed a system of target-centric ontology; the system models the huge number of classes of computer intrusion and their corresponding attack strategies. This system protects from network level attacks. L.Daniel Costa [4] suggested the insider threat indicator ontology approach. The system detects, prevent, and mitigate from insider threat based on behavioural and technical observables of insider activity.

Ontology for information security proposed by Herzog[5] models the assets, threats, vulnerabilities, countermeasures and their relationships. The system is capable of generate new knowledge through inference and using OWL Reasoner. McHugh [6] proposed the system that focused on the classification of attacks based on protocol layer. The system carries only protocol layer attacks and ignores web application attacks. Carlos Blanco et al. [7] present a systematic review of existing security ontology proposals. The survey shows that various security ontologies are developed for each phase of software life cycle and for reusability.

Fenz et al. [8] focuses the security ontology for certification of ISO/IEC 27001 and security guidelines or policies. Parkin et al. [9] developed ontology for human-oriented security issues.

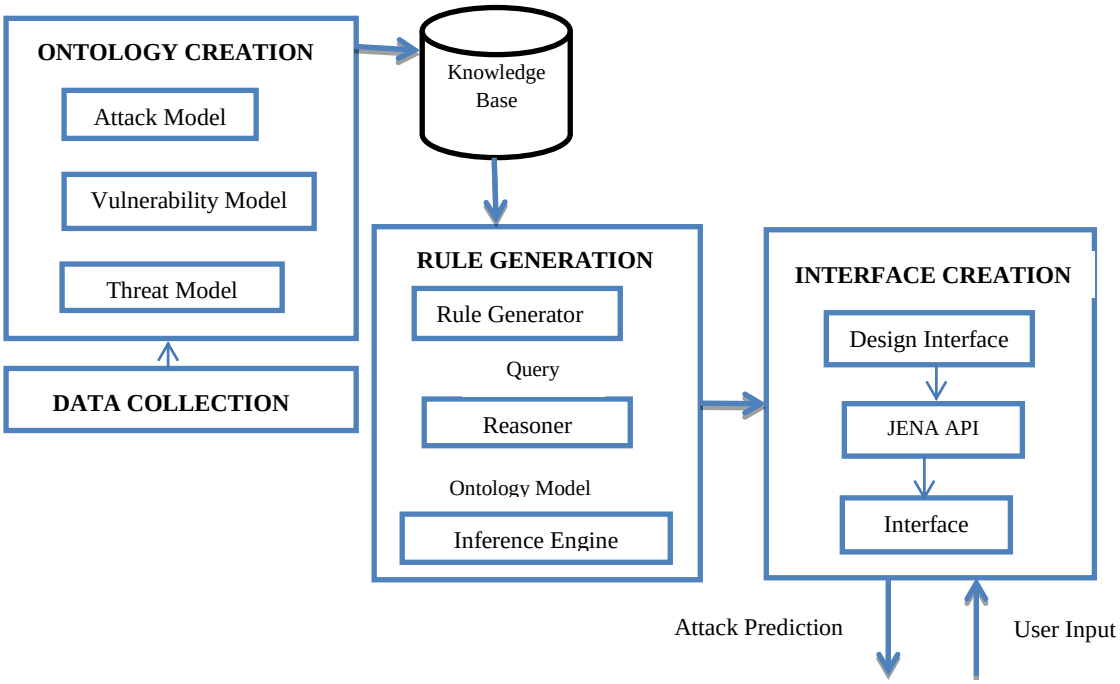

Figure 1: Architecture Diagram for Proposed System

Therefore ontology based system is can be used for as a knowledge source and reusability which avoids ambiguity and inconsistency.

Nadya ElBachir El Moussaid et al. [10] in a survey paper on web application attacks and vulnerabilities. They also use Intrusion Detection System (IDS) and scanners to improve web application security. F. Abdoli [11] developed Distributed IDS for detect attacks and intrusion using special attack ontology. Undercoffer [3, 6] provides better usage of ontology system to gather domain knowledge, protect network level attacks and ignore web application attacks. Carlos Blanco [7] provides comparison between various ontological approaches that can be reused or available for reuse. From study it is noted that only few ontology are created for vulnerability and attack which are used in testing phase and that too are not available for reuse. The existing ontologies ignore web application vulnerabilities and attacks.

As a solution, ontology based system is developed which can be used in testing phase and available for reusability. The proposed approach concentrates only web application threats, vulnerability and attacks.

3. PROPOSED METHOD

In this section, detailed descriptions of the components in the proposed system are discussed.

The proposed system architecture view is shown in Figure.1; it depicts the various components, such as data collection, ontology creation, rules generation and interface creation. Ontology generation and rule generation is core component which is used to generate attack prediction rules. The knowledge base is used to store concepts such as web application attack, threats, vulnerability in the form of ontology model. The main usage of Vulnerability ontology model is reusability and expandability based on the security requirements of the web application. Each ontology model is further sub divided into sub modules.

The proposed system can be used to gather knowledge about the threat which exploits the vulnerability to breach security and cause possible harm to the security goals. The query based on vulnerability can be used to identify the security flaws in web application that allows the attacker to attack can also be predicted. The assets are affected by this attack, impact for the attack, mitigation, prevention methods from that predicted attacks can also be retrieved.

3.1 Data Collection

First component of the proposed system is data collection, the function of this component is to gather the required information such as Threats, vulnerability, attack, countermeasures and prevention from various sources such as National Vulnerability Database (NVD) which is a repository for standard vulnerabilities, Open Web Application Security Project(OWASP) to gather attack information, Common Weakness Enumeration (CWE) gives the software weakness, Common Vulnerability Scoring System (CVSS), is a scoring system to compute the score of the vulnerabilities.

3.2 Ontology Creation

In this component there are three ontology models namely threat model, vulnerability and attack model. There is dependency relationship between them which is further used to predict attacks.

3.2.1 Threat Model

The threat ontology model provides the base for developing vulnerability ontology model and is most important to detect the vulnerability scenarios in turn to predict the attack scenarios. In this threat ontology model, web application related threats are modeled.

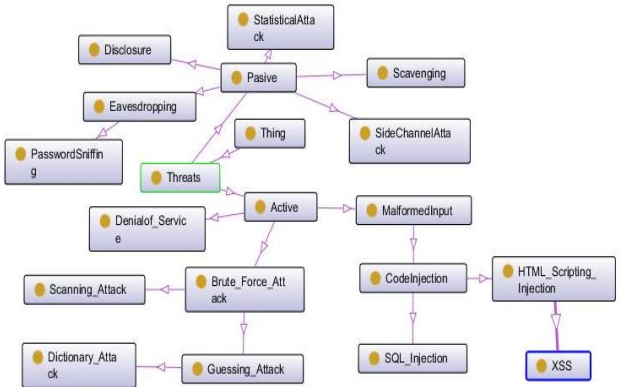

Figure 2: Threats and their concepts

The class threats are a superclass of all the specific web application threats such as active and passive. The model includes the additional concepts for Information Disclosure, Eavesdropping, Side Channel Attack, Denial Of Service, Malformed Input, Code Injection. Figure.2 depicts the interconnection of various threats and concepts.

3.2.2 Vulnerability Model

Vulnerability model gives the foundation for creating attack ontology model and is vital in predicting the attack scenarios. In this vulnerability ontology model, web application related vulnerabilities are modeled. The class vulnerabilities is parent of all the specific web application vulnerabilities such as Cross Site Scripting (XSS), SQL Injection, Cross Site Request Forgery (CSRF), Denial Of Services, Content Spoofing, Information Leakage, Insufficient Authentication, Insufficient Authorization and Brute Force etc.

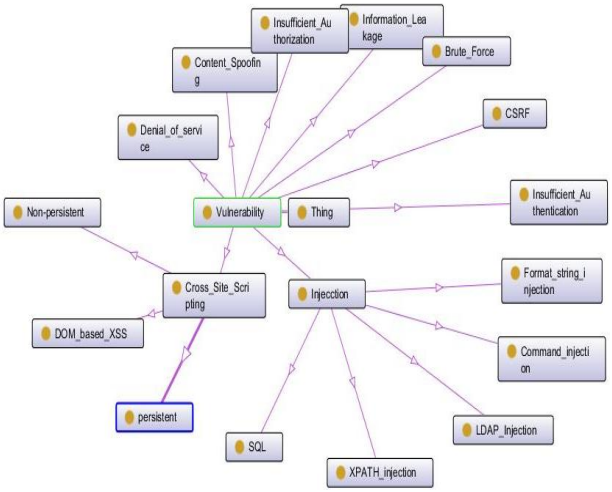

Figure 3: Vulnerabilities and their concepts

The Figure 3 shows the interconnection of vulnerabilities and concepts. The vulnerability model contains the further concepts such as DOM based XSS, persistent and non persistent XSS.

3.2.3 Attack Model

The attack model defines several important security terminologies. It contains various concepts including: web application threats and vulnerabilities, which are used to predict the web application attacks; the weakness of web application is taken as loop hole to perform attacks; assets that are affected by these attacks; the impact of the each web application attacks; countermeasures to mitigate such attacks; prevention methods applied to avoid the attacks. Web application attacks are continually growing, the model can be able to reused and simply extended over time. The main classes/concepts of the ontology model such as: the Attack Class which has object properties, hasThreats, hasExploitedby, affectedTo, resultingIn, migitigatedBy and avoidedBy, which are defined by the classes/concepts Threats, Vulnerability, Assets, Impact, Countermeasures and Prevention Methods is shown in Figure 4.

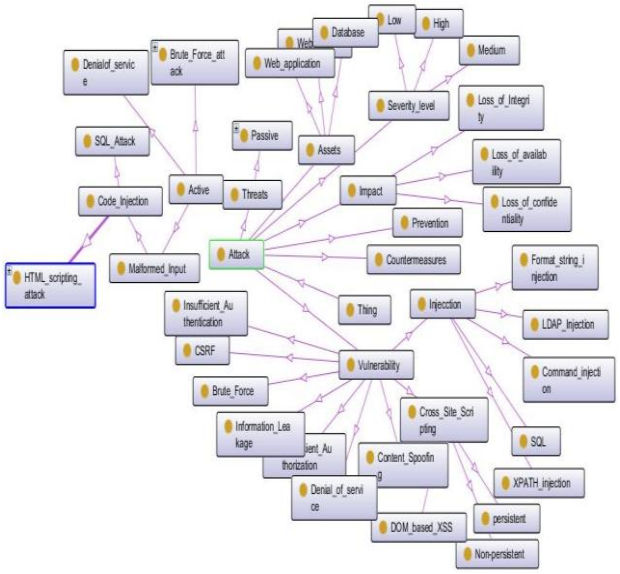

Figure 4: Attack model and their concepts

For instance, the XSS attack is a subclass Attack class that exploited by XSS vulnerability in the application. This attack uses active/passive threats, exploited by (XSS) vulnerability, affects the Assets (Client Browser), resulting in Impact (Page modification and redirect) mitigated by the countermeasures (Use HTTP only cookie flag), and avoided by the prevention methods (Restrict untrusted javascript).This logical construct can be expressed as given below:

XSSAttack⊑Attack⊓∃hasThreats. Threats (malicious code) ⊓∃exploitedBy. Vulnerability (XSS) ⊓∀effects. Assets (Client Browser) ⊓∃resultingIn. Impact (Page modification and redirect) ⊓∃mitigatedBy. Countermeausres (Use HTTP only cookie flag) ⊓∃avoidedBy. PreventionMethod (Restrict untrusted javascript)

3.3 Rule Generation

The next component is the generation of the prediction rule, which consists of following steps:

i. Vulnerability ontology is stored in RDF-tuples and the ontology model is gained by inference process.

ii. The obtained inferred knowledge is created by specific rule grammar and rule template.

iii. Generating the prediction rules through querying the inferred knowledge. The pattern of the query is 3-tuple (triple) and is used to retrieve results rules and ontology models. Inference process also used to generate new rules, such capability is to satisfy the user's requirements.

The rule template can be formulated as shown in Figure 5. It defines the basic information flow of a user input, and process the input (a vulnerability as query) to predict the attack. It shows the attack information which includes ''web application'', an ''effects'' variable that represents the assets that is affected by this attack. The ''impact'' variable

describes the consequence of the attack and a ‘‘condition’’ variable used to predict the attack based on their user input as the result of some ‘‘action’’ that is countermeasures for that attack and also prevention methods to control that attack.

The rule template also describes severity level of the web application attack. In order to predict a specific attack situation the template is important with the help of inferred knowledge such as attack and mitigation mechanisms. Every rule instance specifies the conditions under which vulnerability exploited attack is predicted and action to be taken as given in the Figure 6. Generally each rule contains a condition variable that has some value for predict a specific attack from the user input and the corresponding mitigation, control action.

Web application: “has some attack”
effects: “Assets”;
impact: “consequences of the attack”;
condition: “threats and vulnerability information”
action: “mitigated information prevention methods”

Figure 5: Rule Template

Web application: “any web application”
effects: Web server (client Browser);
impact: Admin privilege is hijacked| Page modification and redirect
condition:” passive threats| malicious code and XSS”
action: “Use libraries| Use HTTP only cookie flag and Restrict untrusted JavaScript”
severity level: High

Figure 6: Instances of rules for attack prediction

WebApplication(?WA) ∧ hasweakness(?WA, ?T) ∧ relatedTo(?W, ?V) → Vulnerability(?V) **Rule (1)**

Vulnerable(?N) ∧ hasVulnerability(?N, XSS) ∧ hasExploitedby(?N, Attack) ∧ hasMitigatedBy(?N, Update_filter_periodically) ∧ hasPreventionMethods(?N, Use_Libraries) ∧ hasSeverityLevel(?N, High) → XSSAttack **Rule (2)**

Vulnerability(?V) ∧ hasSeverityLevel(?V, ?H) → High(?H) **Rule (3)**

Figure 7: Prediction Rules Based on Vulnerabilities

3.3.1 Inference Process

With the knowledge base and the existing relationship between the concepts help to infer new solution. This inference process is used to create new assertions. The predictive inference rules are stated in Figure 7. Rule one states that how web application is vulnerable and rule 2 represents overall pattern of attack prediction, mitigation and prevention. Rule 3 represents the classification of attack based on their severity level. Semantic Rule Language (SWRL) is used to represent the rules for the weakness of the web application and vulnerability that may cause attack. Through inference process a new knowledge is attained, thus attack is prediction is performed effectively.

3.4 Interface Creation

The last component in the proposed system is the interface creation.

3.4.1 Design Interface

Using the net beans environment and Java, the interface is designed for interaction. Jena API is used to integrate the ontology model, rules and application interface.

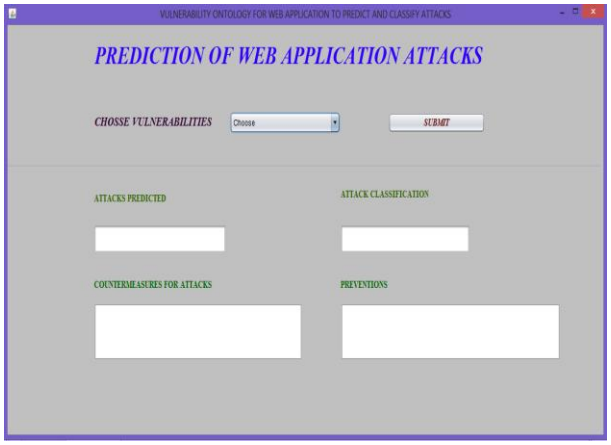

Figure 8: Application Interface

3.4.2 Functionality

Interface is designed to support user friendly as depicted in the Figure 8. It consists of two main parts: first is to get the input from the user and the second part is to display information. The detailed functionality of application interface is discussed below.

The first part is used to get the user input; the vulnerability is chosen by the user from the list box and query is submitted, the ontology model, the rules and SPARQL query are loaded to predict the attacks. The alert message is displayed as shown in Figure 9, to handle any error.

Figure 9: Application Interface with alert

The Second part is to predict and classify attack based on user input as shown in the Figure 10. The countermeasures that needed to control the attack and preventive measures to protect the web application attack are also displayed.

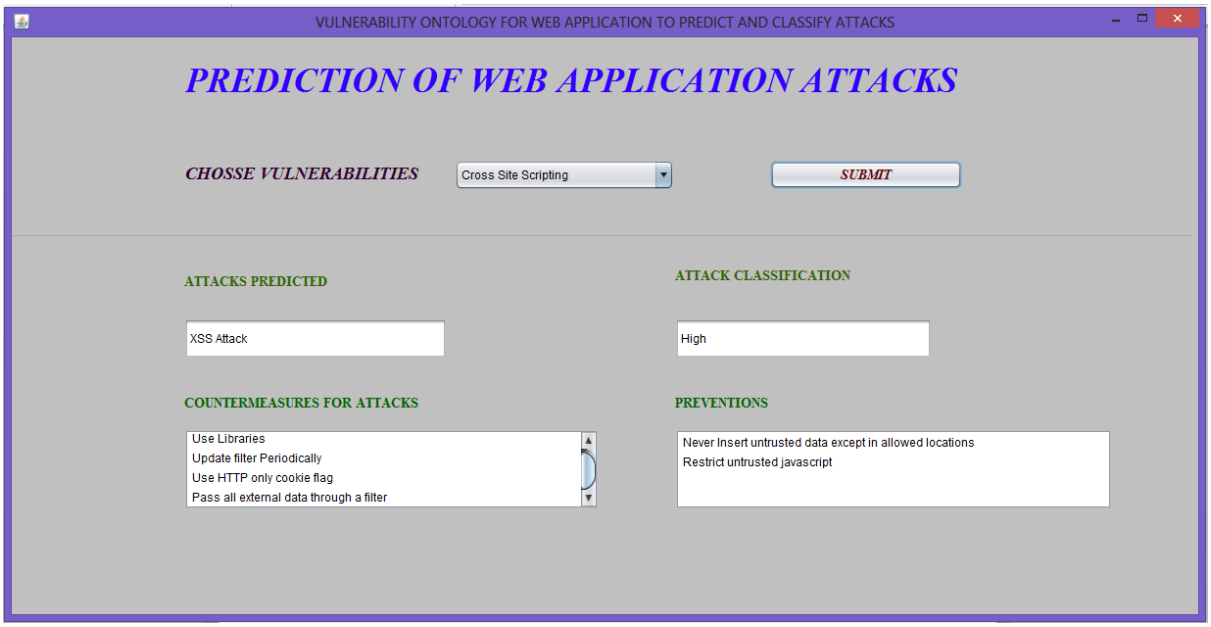

Figure 10: Predict and classify attacks

4. EVALUATION

In order to evaluate the proposed method, a web application was developed with resolving the vulnerabilities and implementing countermeasures against the predicted attacks. The web application was scanned and found less vulnerabilities. The retrieved information were measured using the parameter such as prediction rate, accuracy of the retrieved information are precision, recall and F-measure.

They are computed by using the following:

Precision = Correct / (Correct + Wrong)

Recall = Correct/ (Correct + Missed)

Where,

- Correct-The number of expected record is retrieve from total number of irrelevant records (information) by the system as well as human
- Wrong- The number of expected record is retrieve from total number of irrelevant records (information) by the system but not by the human.
- Missed- The number of expected record is retrieve from total number of irrelevant records (information) by the human but not by the system.

F-Measure = (2*(Precision*Recall))/ Precision + Recall

The prediction rate of the proposed system plotted for each attacks as shown in Figure. 11. From the chart, one can observe that in each attack the prediction rate is high, which help the attackers to understand the consequence of the vulnerabilities in the web application being developed.

The prediction rate of proposed system results are compared against security ontology [12] and tabulated in the Table 1. From the Table 1 it is clear that the proposed system predict more attacks with the help of the inference process than the existing system.

The attack classification is also compared with the existing system. The Figure 12 shows the chart comparing the percentage of attack classification. Capability of attack classification of existing system is low.

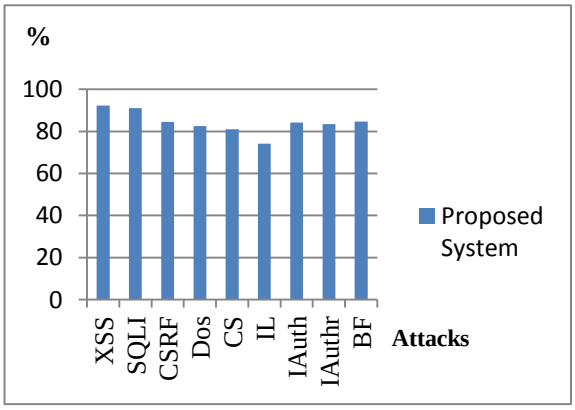

Figure 11: Prediction rate of proposed system

Table: 1. Comparison of Prediction rate

Web Application attacks	Proposed System	Existing System
Cross Site Scripting (XSS)	92.3	86.9
SQL Injection (SQLI)	91.05	87.09
Denial of services (Dos)	84.56	63.04
Cross Site Request Forgery (CSRF)	84.56	63.04
Content Spoofing (CS)	82.57	72.43
Information Leakage (IL)	74.09	70.08
Insufficient Authentication (IAuth)	84.23	66.89
Insufficient Authorization (IAuthr)	83.45	64.23
Brute Force (BF)	84.67	60.56

The experimental results shows that the prediction capability and attack classification rate of our proposed system are significantly better than the existing system. The system successfully predict web application attacks, the average prediction rate is high compared to existing system.

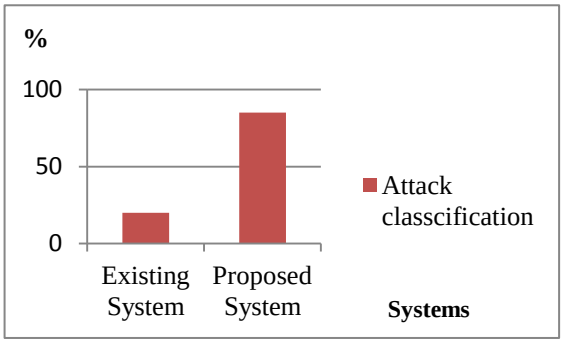

Figure 12: Comparison of Attack classifications

Additionally, our proposed system attack classification rate is also high compared to existing system. This is because our system is capable of predict sophisticated attacks effectively by analyzing the vulnerability that may exploit the attacks and the threats that was used. An inference process is to acquire new knowledge and rules, used to predict the maximum attacks.

5. CONCLUSION AND FUTURE WORK

The survival shows information sharing and e-business is increased rapidly on the other hand security of the individual information are important. Through web application hacker and crackers getting secure information of the users and exploit vulnerabilities. Web application developers and testers do not have enough knowledge about threats, vulnerability, attacks so that they end up in developing insecure application.

Various technologies are available to handle this problem but they are all ineffective to manage and not provide full security solution for developing secure web application. The ontology based system can predict and classify web application attacks.

The proposed system effectively analyzing threats and vulnerability that may exploits web application attacks. Ontology model for threats, vulnerability, attack and rules are used to predict sophisticated attacks effectively and efficiently. The process of inference engine based on reasoning the web application vulnerabilities information generates the list of attacks. The attacks are classified based on the severity level of the attacks on security goals (Confidentiality, Integrity and availability).

The proposed system also gives suggestion to mitigate and prevention for predicted attacks. This information retrieved is very useful for developer and testers to manage the attacks so that secure application is designed. As the future work the proposed ontology model can be reused to detect the web application attacks in the testing phase.

6. ACKNOWLEDGMENTS

Our sincere thanks to the experts and reviewers for their valuable comments and suggestions.

7. REFERENCES

[1] M. Vrancianu, L.A. Popa, Considerations regarding the security and protection of e-banking services consumers' interests, The Amfiteatru Economic Journal 12 (28) (2010) 388–403.

[2] J. Kannan, P. Maniatis, B.G. Chun, Secure data preservers for web services, in: Proceedings of the 2nd USENIX Conference on Web Application Development, USENIX Association, 2011, pp. 3–3.

[3] J. Undercoffer, J. Pinkston, A. Joshi and T. Finin, "A target-centric ontology for intrusion detection", In 18th International Joint Conference on Artificial Intelligence, pp. 9-15, March 2004.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[4] L.Daniel Costa, L. Matthew Collins, J.Samuel Perl, J.Michael Albrethsen, J.George Silowash, L. Derrick Spooner, An Ontology for Insider Threat Indicators Development and Applications, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA .

[5] A. Herzog, N. Shahmehri, C. Duma, An ontology of information security, Techniques and Applications for Advanced Information Privacy and Security: Emerging Organizational, Ethical, and Human Issues (2009) 278–301.

[6] J. McHugh, Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by Lincoln laboratory, ACM Transactions on Information and System Security 3 (4) (2000) 262–294.

[7] Carlos Blanco, Joaquín Lasheras, Eduardo Fernández-Medina, Rafael Valencia-García and Ambrosio Toval, "Basis for an integrated security ontology according to a systematic review of existing proposals", Computer standards and Interfaces, Vol. 33, No. 67, pp. 372-388, June 2011.

[8] S. Fenz, G. Goluch, A. Ekelhart, B. Riedl, E. Weippl, Information security fortification by ontological mapping of the iso/iec 27001 standard, in: 13th Pacific Rim International Symposium on Dependable Computing, 2007, PRDC 2007, IEEE, 2007, pp. 381–388.

[9] S.Parkin, A. Moorsel, and R. Coles, (2009). An information security ontology incorporating human-behavioural implications. In Proceedings of 2nd International Conference on Security of Information and Networks, pp. 46–55.

[10] Nadya ElBachir El Moussaid, Ahmed Toumanari, "Web Application Attacks Detection: A Survey and Classification", International Journal of Computer Applications, 2014,Vol 103, No.12.

[11] F. Abdoli and M. Kahani, "Ontology-based Distributed Intrusion Detection System", In Proceedings of the 14th International CSI Computer Conference.

[12] Golnaz Elahi, Eric Yu, and Nicola Zannone, "A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations", Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 99-114, 2009.