A threat risk modeling framework for Geospatial Weather Information System (GWIS): a DREAD based study

K. Ram Mohan Rao[#], Durgesh Pant[*]

[#]Geoinformatics Division, Indian Institute of Remote Sensing
Kalidas Road, Dehradun, India
rammohan@iirs.gov.in

[*]Department of Computer Science, Kumaun University
Nainital, India
durgesh.pant@gmail.com

Abstract **— Over the years, the focus has been on protecting network, host, database and standard applications from internal and external threats. The Rapid Application Development (RAD) process makes the web application extremely short and makes it difficult to eliminate the vulnerabilities. Here we study web application risk assessment technique called threat risk modeling to improve the security of the application. We implement our proposed mechanism the application risk assessment using Microsoft's threat risk DREAD model to evaluate the application security risk against vulnerability parameters. The study led to quantifying different levels of risk for Geospatial Weather Information System (GWIS) using DREAD model.**

Keywords **— Rapid Application Development, Risk rating, Security assessment.**

I. Introduction

There has been tremendous success of World Wide Web (WWW). Today most of the applications are developed using web technologies in different areas *viz.*, banking, ecommerce, education, government, entertainment, webmail and training. Many companies are depending on their web sites for the publicity and business and some of the companies came into business like online shopping through the possibilities of WWW only. Many of customers also find convenient to get benefit from these services of web application rather than conventional or manual methods. The technology of web also enormously developed with modern technologies to build more reliable and cost effective web applications. The technology is now in a position to cope up with various issues like interoperability, multiple platforms and to connect with different database technologies.

Despite the importance of web applications with improved technologies, hacking techniques also gained momentum in cashing the vulnerabilities of the applications. Web Application Security Consortium gave report on web hacking statistics [1]. These statistics clearly states that the number is gradually increasing from year to year, even with the added security feature technology in web application development tools.

II. Security Challenges

Web applications are increasingly becoming high value target for attackers. 71% of the reported application vulnerabilities have affected the web technologies such as web servers, application servers and web browsers [2]. In 2007, a survey was conducted by the Cenzic and Executive alliance on the state of web application security level [3]. Some of the interesting key findings are, there is lack of confidence in the current state of web application security. Around 50% of the people are not confident about their application security, although most of them are happy about their application technology. 83% of the CEOs are aware of the web security, but most of them and other senior management are not sure about the financial implications of the unsecured web applications.

The above findings evidently show that, organizations are still not matured enough to take care of the application security issues against the ever growing threats. Therefore, it becomes imperative than ever to assess the web application security concerns. In the past, organization relied more on gateway defenses, Secure Socket Layer (SSL), network and host security to keep the data secured. Unfortunately, majority of the web attacks are application attacks and the mentioned technologies are generally unable to cope up with the security needs against the application attacks [4]. The gateway firewall and antivirus programs though offer protection at network and host level, but not at the application level [5]. Firewall may not detect malicious input sent to a web application. Indeed, firewalls are great at blocking ports, but not complete solution. Some firewall applications examine communications and can provide very advanced indication still. Typical firewall helps to restrict traffic to HTTP, but the HTTP traffic can contain commands that exploit application vulnerabilities. Firewalls are only an integral part of security, but they are not a complete solution [6]. The same holds true for Secure Socket Layer (SSL), which is good at encrypting traffic over the network. However, it does not validate the application's input or protect from a poorly defined port policy.

The Software Unlimited Organization [7] listed the top 10 firewall limitations. Web servers are becoming popular attack targets. Between 1998 and 2000, around 50 new attacks exploit the Microsoft's widely utilized web server Internet Information Server (IIS) and published these reports in the public domain [8]. Of these attacks 55% allowed an intruder to read sensitive information such as ASP source files, configuration files and finally the data records as well. These growing numbers of attacks target the databases which reside behind the web server. By exploiting the vulnerabilities in the web server it is possible to run SQL commands for gaining the access of database server. Hence protecting the web server is becoming huge concern in the web application security domain.

A. *Web application concerns*

Today's client/server technology has progressed beyond the traditional two tiered concept to three-tier architectures. Application architectures have three logical tiers called presentation services, process services, and data services. As with all these technologies, three tier gives the opportunity to reap these benefits, but a number of challenges to implementing three tier architecture exist. This is because of the number of services that need to be managed, and because the tools are still skeletons for the applications. Furthermore, three tier systems are inherently more complicated because of the multiple technologies involved in the design and development of the application. From pure security point of view, lack of security in any one of the technology will result the total system vulnerable.

Web application must be secured in depth, because they are dependent on hardware, the operating system, web server, database, scripting language and application code. So web applications have numerous entry points that can put database at risk. Hackers generally look into the different fundamental areas of application to break the security. The general types of attacks are IP access, port access, and application access. Hackers get the IP address of the server and do the telnet to exploit the server. There are so many tools for extracting the passwords of the logins. Applications are normally configured to listen on a predefined port for incoming requests. These vulnerable ports are also major sources for the attacks on the application. Web applications include the series of web servers, file servers and database servers etc. Each of these servers attracts potential point of entry to break the application security. But there are so many other areas where the application is vulnerable to the attacks. The major challenges associated with the web application are their most critical vulnerabilities that are often the results of insecure information flow, failure of encryption, database vulnerabilities etc [9]. They are inherent in web application codes, and independent of the technologies in which they are deployed [10]. Attacker may exploit these vulnerabilities at anytime. Almost every week, the media reports on new computer crimes, latest attack techniques, application vulnerabilities, system break-ins, malicious code attacks, and ever growing cyber crime threat. Web Application Security Consortium (WASC) has listed the top 10 web application vulnerabilities for the year 2007 out of reported 24 classes of attacks. Application vulnerabilities, network vulnerabilities, viruses, trojans etc. are some of the external threats. But there are many other internal threats other than external threats posed by rogue administrators, bad employees, some casual employees and social engineering. The solution to the web application security is more than technology. It is all about practices, precautions and countermeasures. That is why security is not a path, its destination. Security is about risk management and effective countermeasures [11].

B. *Security assessment*

Traditionally, security assessment has been considered sa sub function of network management, and has been identified as one of the functional areas of the open system interconnection, management framework. As defined in the OSI management framework, security assessment is concerned not with the actual provision and use of encryption or authentication techniques themselves but rather with their management, including reports concerning attempts to breach system security. Two important aspects are identified (i) managing the security environment of a network including detection of security violations and maintaining security audits, and (ii) performing the network management task in a secure way [12]. Sloman et al, 1994 defines security assessment as the support for specification of authorization policy, translation of this policy into information which can be used by security mechanisms to control access, management of key distribution, monitoring and logging of security activities [13]. Meier et al, 2004 defines security assessment involves holistic approach, applying security at three layers: the network layer, host layer, and the application layer [14]. Additionally, applications must be designed and built using secure design and development guidelines following good security principles. Russ et. al., 2007 concludes security assessment is an organizational level process that focuses on the nontechnical security functions within an organization [15]. In the assessment, it examines the security policies, procedures, architectures, and organizational structure that are in place to support the organization. Although there is no hands on testing (such as scans) in an assessment, it is a very hands on process, with the customer working to gain an understanding of critical information, critical systems, and how the organiation wants to foucs the future of security.

Application security is the use of software, hardware and procedural methods to protect applications from external threats. Security measures built into application and sound application security procedures minimize the likelihood of the attack. Security is becoming an increasingly important concern during development as applications are more frequently accessible over networks. As a result, applications are becoming vulnerable to a wide variety of threats. Application security can be enhanced by rigorously by implementing a security framework known as threat modelling. It is the process of defining enterprise assets, identifying what each application does with respect to these assets, creating security profile for each application, identifying and prioritizing potential threats.

III. General Threat Modeling Principles

Threat is a specific scenario or a sequence of actions that exploits a set of vulnerabilities and may cause damage to one or more of the system's assets. Threat modeling is an iterative process that starts in the early phases of analysis, design, coding & testing and continues throughout the application development life cycle. It systematically identifies and rates the threats that are most likely to effect the web application. By identifying and rating the possible threats with detailed understanding of application architecture the appropriate countermeasures can be implemented against all possible threats in a logical order. Fig. 1 shows the threat modeling process, which is an iterative process

Threat modeling is an essential process for securing web application. It allows organizations to determine the correct controls and product effective countermeasures against all vulnerabilities in the application. Fig. 2 shows the interrelation between a threat and assets, vulnerabilities and countermeasure entities. The threat described in the figure may cause damages to any of application assets and even may exploit all possible vulnerabilities in the system. A successful attack exploits all vulnerabilities in the application and may take over the total control of application. It is probably because of weak design principles, weak coding practices, and configuration mistakes of the applications. Well defined countermeasures can be implemented to the application to mitigate attacks as shown in fig. 2.

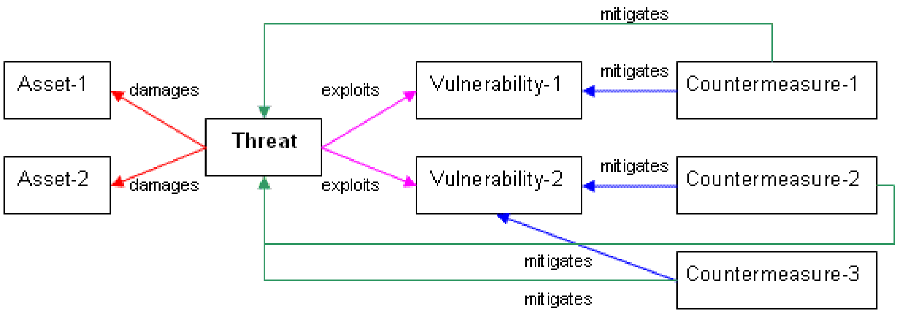

Fig. 2 Interrelation between threat, asset, vulnerability and countermeasure [17]

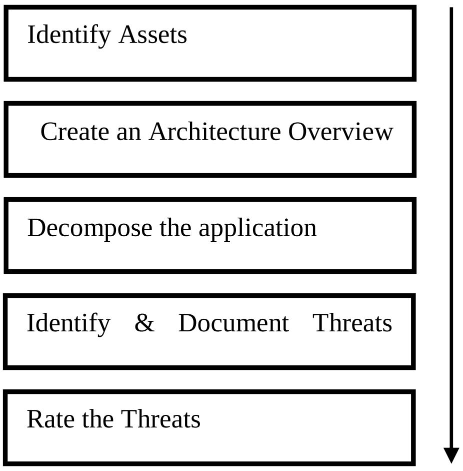
Identify Assets
Create an Architecture Overview
Decompose the application
Identify & Document Threats
Rate the Threats

Fig. 1 Threat modeling process [16]

A. *Performing threat risk modelling*

Application development team needs to understand the organization security policy and the overall objectives of the application. Asset is information, capability, an advantage, a feature, a financial or a technical resource that should be defined from any damage, loss or disruption. The damage to an asset may affect the normal functionality of the system as well as the individuals or organizations involved with in the systems. Normally, in the web application technology assets are database, application and web servers.

It is always a difficult task to build a secure web application without knowledge of possible threats. The purpose of the threat modeling is to analyze the application design with solid understanding of application architecture.

The next step is documenting the known treats by keeping an intelligent attacker in mind to shape the application design to meet security objectives, reduce the risks arising during development and deployment. While designing web application, it is essential to design threat risk assessed controls to make application assets more hack resilient at the design time rather than the deployment stage. But it is not possible to document all the possible threats a web application faces as the application development is dynamic process in nature. So the option would be conducting a brain storming session with development people, testers, architecture designers, and professionals etc. to identify the maximum threats at the design time itself. Then the process of documenting the threats in a hierarchical mode that defines core set of attributes to capture for each threat. It is important to rate the threats to prioritize the most frequently occurring possible threats, and which can cause maximum risk to the application. The rating methods depend on different parameters and generally calculated with probability of occurrence and the damage potential that threat could cause.

A. Threat risk models

Over the last five years, threat risk modeling became important mitigation development in the web application security environment [18]. Different process models exist for identifying, documenting and rating the threats such as Microsoft Framework, OWASP model, Trike, CVSS, AS 4360 and OCTAVE model [19]. It is up to the security specialist to choose the model according to the suitability of risk assessing method and the technology being used in the application. It is always best practice to adopt one of the risk models to reduce the business risk to the application. This study adopts the basic Microsoft Threat Modeling methodology for implementing threat risk modeling both at design and implementation stages.

IV. Geospatial Weather Information System: A Threat modeling approach

Geospatial Weather Information System (GWIS) is a web based tool for capturing, storing, retrieving and visualization of the weather climatic data. The GWIS contains historical climatic data for nearly hundreds of land stations country wide. The database is provided with both climatic daily and monthly data. Daily data has been nearly for 150 ground stations country wide and covering temperature, rainfall, humidity details. The climatic monthly data has for wide range of land stations around 3000 countrywide. Daily data is being captured from different sources after then arranged in GWIS format for storing in the database. The source for monthly data is Global Historical Climatology Network (GHCN). It is used operationally by National Climatic Data Centre (NCDC) to monitor long-term trends in temperature and precipitation. The mission of GWIS is to integrate the weather related information from different available sources and organize the data in structured GWIS format. The application tool is designed to cater the research needs of various application scientists working on different themes.

Microsoft provides a thereat-modeling methodology for .NET technologies. The process starts from identifying threats, defining architecture overview, decomposing the application, identifying the threats, document the threats and rating the threats. More emphasis has been given to the detailed architecture design describing composition and structure of the application including the sub systems addressing the technologies being used in the web application. As the Microsoft always emphasizes on holistic approach methodology, it again adopts holistic approach in identifying the threats [20].

A. Identifying threats

Threats are generally point to network, host and application layers. Identifying network threats is mainly concerned with understanding the network topology, the flow of data packets and the connecting network devices such as router, firewall, and switch. The most frequently occurring network threats are IP Spoofing, Session hijacking, open port policies, open protocols and any weak authenticated network device.Host threats mainly concerned with the security settings of operating system. Possible host vulnerabilities are unpatched servers which can be exploited by viruses, systems with nonessential ports, weak authentication, social engineering etc. Application threat is a big area compared to any other domain of web application. Since the web application includes combination of multiple technologies, there is always a chance for the technology gap between any two. Hence it is always important to evaluate the application vulnerability categories. The major application vulnerability categories are authorization, input validation, cryptography, configuration management, and exception handling. The mentioned areas are normal known threats in the web application environment. But there may be many more number of unknown threats in specific area. However, there are some other approaches to document potential threats using attack trees and attack patterns.

B. Attack trees and Attack pattern

As web application often includes the client / server technology with dynamic process of application development, it is very difficult to document all the possible threats. Attack Trees and Attack Patterns are special tools that most of security professionals use for identifying potential threats in the application. They refine information about the attacks by identifying the compromise of enterprise security or survivability as the root of the tree. Each tree represents an event that could significantly harm the asset. Each path through an attack tree represents a unique attack of the asset. Typically threat tree imparts lot more information in shorter time for the reader but takes longer to construct, and attack pattern is much easier way to write but takes longer for the impact of the threats to become obvious. Attack trees provide a formal way of describing the security of systems, based on varying attacks. It represents attacks against a system in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes. Fig. 3 and 4 represents attack tree and attack pattern of GWIS respectively. Attack trees are represented in a tree structure by decomposing a node of an attack tree either as,

- a set of attack sub-goals, all of which must be achieved for the attack to succeed, that are represented as an AND-decomposition.
- a set of attack sub-goals, any one of which must be achieved for the attack to succeed, they are represented as an OR-decomposition.

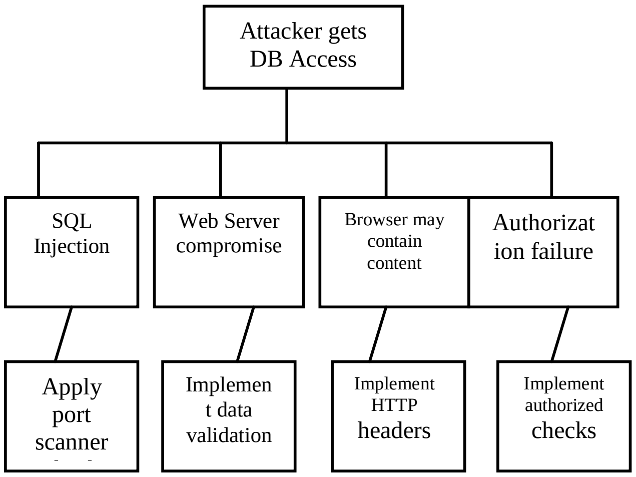

Fig. 3 Attack tree representation of GWIS

Attack patterns are generic representations of commonly occurring attacks that can occur in a variety of contexts. The pattern defines the goal of the attack as well as the conditions that must exist for the attack to occur, the steps that are required to perform the attack, and the results of the attack [21].

Threat #1:The attacker will learn the structure of the SQL query, and then use this knowledge to thwart the query by injecting data that changes the query syntax into performing differently than indented.

1.1 SQL Vulnerabilities
1.1.1 Block SQL Injection, Blind SQL Injection, Cross Site Scripting, HTTP response splitting etc.

1.1.1.1 By verifying that user input does not contain hazardous characters, it is possible to prevent malicious users from causing your application to execute unintended operations, such as launch arbitrary SQL queries, embed javascript code to be executed on the client side, run various operating system commands

Fig. 4 Attack paattern representation of GWIS

B. *Document the threat*

Documenting the possible known threats of GWIS application gives the great edge to deal with the vulnerabilities. Sometimes it is very difficult to document the unknown threats. But documenting the known threats with the help of third party vulnerability assessment tools will give great knowledge to the developer / administrator to reduce the risks. GWIS application has been scanned thoroughly to perform vulnerability testing to find out the vulnerabilities in the application. For this type of application assessment, single type of vulnerability scanner is not sufficient for scanning the application. So larger sites may require multiple vulnerability scanners to support the assessment needs. The reason is the specific tools are effective in some of the areas and may not be good at other functional areas. For this reason, the GWIS application has been scanned with multiple scanners namely AppScan, CENZIC, and Nessus tools. The consolidated list of vulnerabilities observed is shown in Table 1.

TABLE I
VULNERABILITIES BY PATTERNS

Vulnerability Patterns	**No.of Instances**
Blind SQL Injection	2
Login page SQL Injection	2
Unencrypted login request	1
Application Error	1
Inadequate account lockout	1
Permanent cookie contains sensitive session information	1
Session information not updated	1
Unencrypted password parameter	3
Unencrypted viewstate parameter	7

The vulnerabilities are documented in the threat list as per the Microsoft threat template. Threat list generates the application threat document with the details of threat target, attack techniques, risk and possible countermeasures that are required to address the threat.

C. *Rating the Risk*

The rating process measures the probability of threat against the damage that could result to an application. This process will generate the list of priority with the overall rating of threats. This allows addressing the most risk generated threat first on priority with proper countermeasures to mitigate the risk. The risk can be calculated from a simple formula [22].

Risk = Probability x Damage potential

Where risk posed by a particular threat is equal to probability of threat occurring multiplied by the damage potential. With this formula risk can be categorized into high, medium and low by calculating in the scale of 1-10. Most of the security professionals may not agree up on the simple rating system in calculating the risk of the application because of the equal distribution of the assets. To resolve this issue, Microsoft came up with a modeling formula called DREAD which is used to calculate risk. On the basis of these parameters, values can be calculated for the given threats and then can be categorized as high risk, medium risk and low risk

D. Rating risk with DREAD approach

DREAD methodology is used to calculate the risk. For each threat the risk rating is calculated by assessing damage potential, reproducibility of attack, exploitability of hte vulnerability, discoverbility of vulnerability and finally total risk points of the application.

D: Damage potential – The loss if the vulnerability is exploited
R: Reproducibility - How easy is it to reproduce the attack
E: Exploitability - How easy to attack the assets
A: Affected users - Average affected users in enterprise
D: Discoverability – How easy to find out the vulnerabilities
T: Total - Total calculated risk points

The threat is rated with high value, if it poses significant risk to the application and needs to be addressed immediately. Table 2 shows the risk rating value of GWIS application using DREAD approach.The scoring system does not consider more than one vulnerability, if the application has more than one number of similar types of vulnerability. For example, GWIS consists of two instances of blind SQL injections and seven unencrypted view state parameters. But finally the scoring has given only for one blind SQL injection, and one unencrypted view state parameter, as the type of vulnerability is same. But when particular type of vulnerability is addressed, total number of instances is taken care. This is because of the reason that, each vulnerability provides equal chance of opportunity for exploiting the application. Once the risk rating is obtained, the threat is documented and with full information of threat target, risk rating, attack technique and necessary countermeasure as shown in Table 3. This template is quite useful for the administrators and application designers for understanding the risk they are dealing with.

V. Experimental Results

The GWIS application has been scanned thoroughly for the vulnerabilities across the presentation, business, and database layers of GWIS. Nine vulnerability patterns are found including total 20 instances.

TABLE II
DREAD SCORES OF

Threat	D	R	E	A	D	T	Average	Rating
Blind SQL Injection	9	6	8	9	6	38	7.60	High
Login page SQL Injection	9	6	8	9	6	38	7.60	High
Unencrypted login request	6	4	6	5	5	26	5.2	Medium
Application Error	2	1	3	2	3	11	2.2	Low
Inadequate account lockout	2	1	3	2	3	11	2.2	Low
Permanent cookie contains sesnsitive session information	2	1	3	2	3	11	2.2	Low
Session information not updated	2	1	3	2	3	11	2.2	Low
Unencrypted password parameter	2	1	3	2	3	11	2.2	Low
Unencrypted viewstate parameter	2	1	3	2	3	11	2.2	Low

The DREAD scores are calculated against each vulnerability of the application, and the final scores are derived as per the risk catagories. In order to experiment with DREAD model, the study has been chosen the GWIS application to implement security assessment. During the assessment phase, the application flaws are completely assessed with variety of tools for finding out vulnerabilities of the application. The found vulnerabilities are billed with DREAD factors. Fig. 5 shows the DREAD severity gauze of the GWIS. The exploitability factor is maximum for the application, which shows that the vulnerabilities present in the applicaiton are easy to exploit. The damanage potential also is more if the vulnerability is exploited by the attacker. Hence fromthe business point of view the risk is medium. Affected user of hte applicaion, discoverability of the applicaiton is medium when applicaiton is explited. But the reproducibility of hte attack is very less for GWIS. So from the technical point of view the risk is less. Now these DREAD scores are combined together to get final severity risk rating for the GWIS.

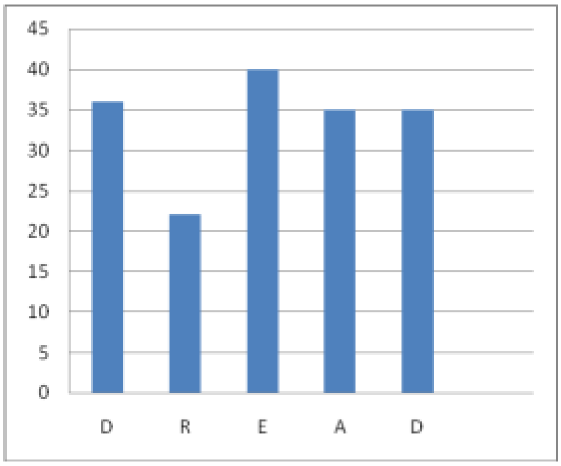

Fig. 5 DREAD severtity gauze

TABLE III
THREAT RISK DOCUMENTATION TEMPLATE OF GWIS

Threat Description	***Blind SQL Injection***	***Login page SQL Injection***	***Unencrypted login request***	***Application Error***	***Inadequate account lockout***	***Permanent cookie contains sensitive session information***	***Session information not updated***	***Unencrypted password parameter***	***Unencrypted viewstate parameter***
Threat Target	Data access component	Data access component	Data access component	Application and Data access component	Application and Data access component	Application	Application	Application	Application
Risk	High	High	Medium	Medium	Medium	Medium	Medium	Medium	Low
Attack technique	The attacker will learn the structure of the SQL query, and then use this knowledge to thwart the query by injecting data that changes the query syntax.	In this case an attacker will inject malicious data, which when incorporate into an SQL query, changes the original syntax .	Information sent to server as a clear text, may be stolen and used later for identify theft user impersonation.	The attacker can gain useful information from the application's response to the request.	Attacker gains the access to the application by sending large number of possible user account by hit and trail method.	During the application test, sensitive information such as user credential or session information are stored in a permanent cookie on client computer.	Session fixation is attack technique that forces user session ID to an explicit value.	Input parameter of the type “ password” is sent unencrypted to the server.	The ASP contains property called view state an is sent to client and back as hidden variables.
Countermeasures	By verifying that user input does not contain hazardous characters,	Do not use the special character	Make sure that sensitive information such as username password , etc. is always sent encrypted to the server.	Check incoming request for the presence of all expected parameters and values.	Fix the number of login accounts to be attempted. Make sure that if number of logi–n account exceeds, the account is locked .	Make sure that sensitive information such as user credentials session tokens will always be stored in a non-permanent cookies.	Always generate a new session to a user by strong userid/password authentication. Prevent user ability to manipulate session id.	Make sure that sensitive information such as username password , data id, lat, long, sid, location id etc. is always sent encypted to the server.	If a property is not persisted in view state, it's a good practice to return its default value on post back.

As shown in Table 3, probability of occurring the threat in GWIS is MEDIUM, and the damage potential is MEDIUM and hence severity level is medium. SQL injection attracts high sever scores, unencrypted login request carries medium severity scores. Rest of the vulnerabilities are of low sever levels. So from pure business point of view, the risk factor is LOW. But on the whole, probability and damage potential levels of GWIS are MEDIUM and MEDIUM respectively. Therefore the overall severity of the risk is MEDIUM. To minimize the risk levels of the GWIS, it is crucial to fix the most sever risk generating vulnerabilities first such as, blind SQL injection and login page SQL injection vulnerabilities in the GWIS. Similarly the other vulnerabilities also should be fixed to further reduce the risk of GWIS.

VI .CONCLUSIONS

Web based application should be addressed by threat modeling process to identify the potential threats, attacks, vulnerabilities and countermeasures. It is basically a software engineering approach to see the application is meeting the company's security objective and to mitigate the risk at maximum level. It helps to identify the vulnerabilities in the application context. The paper has discussed Microsoft's framework DREAD approach to evaluate the risk of GWIS, and the remediation levels for the vulnerabilities. The output of threat modeling process is standard document of the security aspects about the architecture of application and list of rated threats. This document helps as reference to the designers, developers and testers to make secure design choices, writing code to mitigate the risks and to write the test cases against the vulnerable areas identified in the document.

REFERENCES

[1] WASC, 2008, Web Application SecurityConsortium, Web Hacking Statistics. Accessed from http://www.webappsec.org/projects/whid/statistics.shtml. Accessed on 14.06.2007.

[2] K Mandeep, 2008, Cenzic Application Security Trends Report - Q4, 2007, Cenzic Inc. Whitepaper. Accessed from http://www.Cenzic.com Accessed on 12.02.2008.

[3] Cenzic, 2008, The voice of IT leadership on Web security, Findings of survey conducted by CENZIC and Executive alliance, October 2007. Whitepaper. Accessed from http://www.Cenzic.com Accessed on 12.07.2006.

[4] IBM, 2008, Web application security management, "Understanding the web application security", Whitepaper. Accessed from http://www.ibm.com Accessed on 12.07.2006.

[5] M Curphey, D Endler, W Hau, S Taylor, T Smith, A Russel , M McKenna, R Parke, K McLaughlin, N Tranter, A Klien, D Groves, By-Gad, S Huseby, M Eizner, R Mcnamara, 2002, A guide to building secure web applications, The open security web application project, V.1.1.1. Whitepaper, Available from http://www.first.org/cvss/cvss-guide.html Accessed on 12.04.2007.

[6] J.D Meier, A Mackman, S Vasireddy, M Dunner, S Escamilla, Murukan A, , Improving web application security : Threats and Countermeasures. Microsoft Corporation, 2003, pp.3,.

[7] SoftwareUnlimited, 2005, Firewall Limitations, Irvine, CA. Whitepaper. Accessed fromhttp://www.softwareunlimited.com/securityfirewalltop10.htm. Accessed on 06.20.2005.

[8] 2001, "Understanding the IIS vulnerabilities: fix them, SANS Institute,Whitepaper.Accessed from http://www.sans.org/training/category.php?c=SEC, Accessed on 06.20.2005.

[9] F Ricca and P Tonella, 2000, Web site analysis: structure and evolution, in:roceedings of the IEEE international Conference on Software maintenance, San Jose, California.

[10] D Scott and R Sharp, Developing secure web applications, IEEE nternet Computing, 2002, vol.6.

[11] J.D Meier, A Mackman, S Vasireddy, M Dunner, S Escamilla, A Murukan, Improving web application security : Threats and Countermeasures. Microsoft Corporation, 2003,pp.3.

[12] A Langsford, , *OSI Management Model and Standards*. Chapter 4 in Network and Distributed Systems Management (Sloman, 1994ed), 1994, pp. 69-93.

[13] M. S Sloman, (1994b). *Policy Driven Management for Distributed Systems*. Journal of Network and Systems Management, vol. 2(4), pp. 333-360, December 1994.

[14] J.D Meier, , A Mackman, S Vasireddy, M Dunner, S Escamilla, A Murukan, 2003, Improving web application security : Threats and Countermeasures. Microsoft Corporation, pp.4.

[15] R Russ, D Ted, Greg Miles, Ed Fuller Greg, Security Assessment: Case studies for implementing the NSA IAM, 2007Syngress, Syngress Media, Inc,

[16] G Ygor, 2005, Practical Threat Analysis for the Software Industry, PTA Whitepaper.Available from http://www.securitydocs.com/library/2848.

[17] J.D Meier, A Mackman, S Vasireddy, M Dunner, S Escamilla, A Murukan, Improving web application security: Threats and Countermeasures. Microsoft Corporation, pp 45-49.

[18] A Wiseman, A Adrew, V.D Stock, C Mark, S Ray, 2003,.OWASP : Open Web Application Security Project, A Guide to build secure Web Applications and Web Services. Available from http://www.owasp.org. Accessed on 22.07.2006.

[19] K.Ram Mohan Rao, D. Pant,2006, A Multi-model approach to threat risk modeling of web based application, ICT 2007 Conference on "Communication and computational techniques: Current and future trends" organized at Dehradun February 10-11, 2007.

[20] J.D Meier, A Mackman, S Vasireddy, M Dunner, S Escamilla, A Murukan, Improving web application security : Threats and Countermeasures. Microsoft Corporation, pp.6-7.

[21] S Andrew, 2003, Using Vulnerability Assessment Tools to develop and Octave risk profile. Available from http://www.giac.org. Accessed on 23.08.2007.

[22] J.D Meier, A Mackman, S Vasireddy, M Dunner, S Escamilla, A Murukan, Improving web application security: Threats and Countermeasures. Microsoft Corporation, pp.63.