Web Defacement: Threat Analysis and Modelling

C M Mishra[*] and R D Singh[#]
c_mmishra@gmail.com[*], singh_rd@gmail.com[#]

***Abstract*- With the proliferation of web applications and its usage in real life, validation of web content become an essential task. Attacker may tamper the contents of web pages through exploiting the attacks such as Spoofing, Tampering, Repudiation, Information disclosure, Elevation of privilege etc. In web defacement, Script-kiddie defaces the visual appearance of the webpage through tamping the text, images, and videos. This type of attack defames the reputation of organizations and misleads the web users. Analysis and modelling of these threats and attacks is a tedious and time consuming task for law enforcement agencies. Hence, there is requirement to identify threats and related vulnerabilities to rank the threatening activities.**

In this paper first threats related to the web defacement has been analyzed and modelled to identify smoothly. Subsequently, verify the integrity of web contents through five different techniques analyze the results of all five techniques. Proposed and developed threat model productively inspects the web defacement attacks and it would be helpful for web administrator to capture the web defacement cases easily.

Keywords: Web Defacement, Threat, Attack, Threat Modelling, Ranking Risk.

I. INTRODUCTION

In today's world internet has become an indispensable part of one's everyday life. Most of the routine transactions are available online, either it be information regarding a subject or other service like reservation, online shopping also known as e-shopping. Websites serve as the source of information. They also contain proprietary data which can be misused. It is the need of the hour to make websites and its services secure.

Security, in its simplest form, is concerned with making sure that meddlesome people cannot read, or worse yet, secretly modify messages intended for other recipients. These are the end users who try to access remote services so they are not allowed. It also addresses the issues of legitimate message to be captured and reproduced and the end user who try to deny that we have sent certain messages.

To develop a secure website threat should be understood. Security mechanisms are not systematic and are probably to put down huge section of the attack space uninvestigated. An attacker requires finding only one security vulnerability in web application to compromise the complete system.

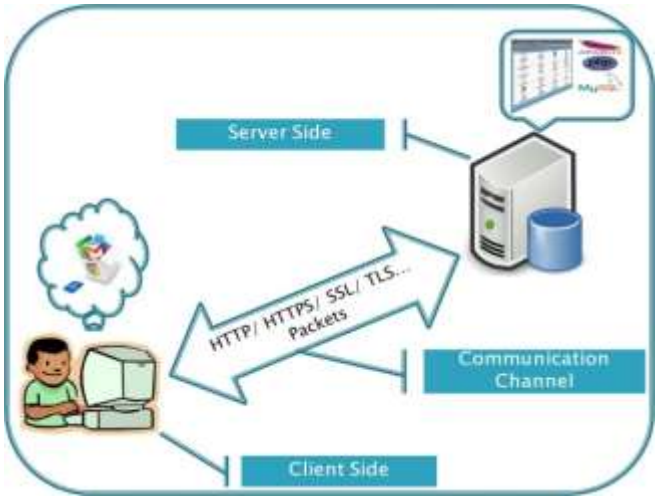

Figure 1: Possibilities of Attacks in Web Communication

Thus systematic study of attack is important. Figure 1 depicts the possibilities of attack in client-server communication. Attacks on web application may be occurred at client side, server side and at communication channel. Threat risk modelling is an essential process to provide web site security. It allows security experts to produce effective security implication and determine the correct controls over existing resources.

This paper is organized as follows. Section II, analyzed the threat related to web defacement as per the threat model and understand the security risk. Section III discus the results of prevention techniques discussed in this paper. Section IV concludes the work done in this paper.

II. Threat Analysis for Web Defacement

According to the ISO 7498-2 'Threat is a potential violation of security and attack is defined as a well-defined set of actions that, would result in either damage to asset or undesirable operation. Threats can attack the resources exploiting one or more vulnerabilities. Threat modelling is a technique that is used to understand relative level of threats and possible harm against one's system. It is an approach that directs one through the process of defining system components, entry and exit points, and key security components and mechanisms. In so doing, one gets a clear architectural system overview and get an idea that how I can exploit those threats. Following are the steps to build scalable threat modelling process:

1. Characterize the system architecture
2. Identifying assets
3. Identifying threats

4. Identify Vulnerability
5. Rank threats by risk
6. Develop method to mitigate threats

A. *Characterize the Web architecture*

The web consists of web server, client machine and a communication channel. Sometime web server referred as hardware system (i.e. computer) or sometime referred as software application (computer software). It supports to deliver web content that can be accessed through the Internet.

Web servers are commonly used to host websites. Common feature of web server:

- *Virtual hosting:* Provide to serve a lot of web sites through single IP address.
- *Large file support:* Provide support to operate files having size greater than 2 GB on 32-bit or 64-bit operating system.
- *Bandwidth throttling:* It regulates the speed of responses in order to proficient the server to serve extra clients and manage the network saturation.
- *Server-side scripting:* It supports to generate dynamic web pages and responsive web design.

And a good client computer is proficient in following characteristics [11] Speed, Accuracy, Versatility, Reliability, Storage capacity, Diligence.

B. *Identifying assets*

Web is a two way network which composed of three components: Client-side components, Communication-side components, Server-side components. Assets are the components of websites which can be categorized into Front-End components, Back-End components and other components.

Front-End components are: The navigation structure, the page layout, Logo, Images, Text and Graphic Design. Back-End components are: Content Management System, E-Commerce, Shopping Cart, Site Search, Contact forms, Referral forms, Newsletter registration, Online databases, Password protected sections and Downloadable files. Other components are: Hosting, Domain Name and Online Promotion.

C. *Identifying threats:*

In general, threats can be classified into six classes based on their effect [11]:

- Spoofing- Attacker gain access of web services through victim's credentials selected from the accessible sets.
- Tampering- Modifying legal web contents in illegal way to exploit an attack.
- Repudiation- It happens while end user denied performing a process, although the target server has no other option.

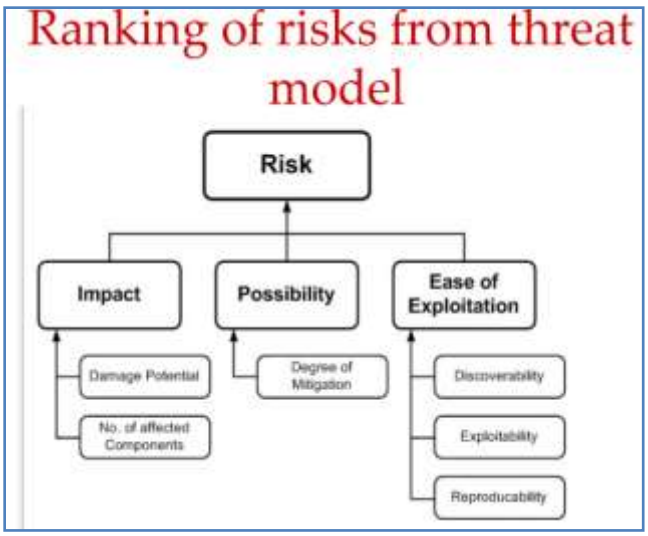

Figure 2: Risk Ranking Model

- Information disclosure- Unknowingly exposure of web information used by attacker to plan attack on system.
- Denial of service- Downgrade the availability web resources to valid end users.
- Elevation of privilege- Process to gain privileged access by unprivileged end user or attacker.

The major web based threats fall into the categories presented in the table 3-1. This also illustrates the vulnerabilities of threats and their control measures.

D. *Identify Vulnerability*

The major web based threats fall into the categories presented in the table 1. This also illustrates the vulnerabilities of threats and their control measures.

E. *Rank threats by risk*

Various factors should be identified in order to analyze risk, including:

- Event: possible changes in web pages.
- Probability: frequent occurrence of the events on web pages
- Impact: Degree of bad effect on website.
- Mitigation: possible reduction of the Probability.
- Contingency: possible reduction of the Impact.

Relation among reduction, mitigation and contingency is as follows:

Reduction = Mitigation * Contingency

Relation among exposure, risk and reduction is as follows:

Exposure = Risk – Reduction

After identification of all the factors, the result produced is called Exposure. This is the amount of risk one simply can't avoid. Exposure may also be referred to as Threat, Liability or Severity, but they mean the same thing.

Table 1: Threats and Related Vulnerabilities.

THREAT TO WEB	VULNERABILITY	CONTROLS
URL Misinterpretation	(i) Web server unable to parse the URL properly. (ii) Mismatched resource mappings in the configuration.	(i) Usually requires a vendor supplied fix. (ii) Thorough inspection of web server configuration and bindings.
Directory Browsing	(i) Capability to acquire entire directory record on the web server. (ii) Commonly occurs while the default document within directory has missed. (iii) Improper configuration of web server.	(i) Properly configured Web server. (ii) Disable directory listing service. (iii) Restrict to show important information in error message through configuring user defined error messages.
Retrieving "non-web" Files	(i) Can be retrieve with guess work. • E.g. in directory such as /reports/, search a file named "report.zip".	(i) Take out careless presence of these types of files. (ii) Adjust change control procedures.
Reverse Proxying	(i) Usually intended to permit access external websites to end users within a network. (ii) You can reach HTTP proxy requests from the external world to the internal network.	(i) Ensure the web server proxy configuration thoroughly. (ii) Carefully create mapping of URLs with internal servers.
JAVA De-compilation	(i) De-compilation of JAVA byte-code is quite effective. (ii) It unveil susceptible data like path of application, passwords, etc.	(i) Exclusion of sensitive configuration information within byte-code. (ii) Exclusion of excessive files contained by .jar format.
Input Validation	(i) Interfere with hidden fields. (ii) Bypassing client side checking (e.g. javascript).	(i) No easy fix. (ii) There is no countermeasure but proper coding practices.
Session Hijacking	(i) Poor mechanisms of state tracking. (ii) Reverse engineering of the session ID manages the retrieving of end users' data.	(i) Use of session ID tracking at server side. (ii) Test connections according to the predicates like time stamps, IP addresses, etc.
Buffer Overflows	(i) Poor bound checking. (ii) Can cause: Denial of service and remote command execution	(i) Bound checking within applications. (ii) Source code reviews.
SQL Query Poisoning	(i) Parameters presents in input fields or URL become useful to fire SQL queries. (ii) Execution of stored procedures. (iii) May even load to back-end database server compromise.	(i) Again, no easy fix. (ii) Through source code review. (iii) Follow the principles of least privilege for database application. (iv) Eliminate unnecessary users of database and stored procedures.

Table 2: Calculated Results of Analyzed Techniques

No. of Web Pages	PSNR		SSIM		CRC 32		MD 5		SHA 512	
	tp + tn	Accuracy	tp + tn	Accuracy	tp + tn	Accuracy	tp + tn	Accuracy	tp + tn	Accuracy
30	25	83.33	24	80	28	93.33	29	96.66	29	96.66
60	53	88.33	51	85	55	91.66	59	98.33	59	98.33
90	78	86.66	76	84.44	81	90	85	94.44	87	96.66
120	103	85.83	100	83.33	107	89.16	113	94.16	115	95.83
150	130	86.66	127	84.66	135	90	141	94	143	95.33

III. RESULT ANALYSIS AND DISCUSSSION

Effectiveness of evaluated integrity techniques has been evaluated through calculating the accuracy on the basis of True Positives (TP), True Negatives (TN), False Positives (FP), and false negatives (FN). Terms positive and negative refer to the actual output and the terms true and false refer to identified outcomes.

- TP: It represents correctly identified. In this work it defined as actually web contents has been defaced and also detected as defaced.
- FP: It represents incorrectly identified that described in this work as web content actually has not been defaced but detected as defaced.
- TN: It represents correctly rejected. Here, it defines image or text actually has defaced but detected as not defaced.
- FN: It represents incorrectly rejected. Here it defines image or text actually has not defaced and also detected as not defaced.

Training data set has been applied to calculate the count of tp, tn, fp and fn (i.e. expected and observed results) during the experiment for evaluated techniques. For the evaluation of result Accuracy, Precision and Recall of the framework has been calculated which are discussed as follows:

A. Accuracy

Equation (5.1) is the formula to calculate Accuracy of the system which is given by:

$$Accuracy = \frac{tp + tn}{(tp + tn + fp + fn)}$$

B. Precision

Equation (5.2) is the formula to calculate Precision of the system which is given by:

$$Precision = \frac{tp}{Observed\ Defaced\ Component} = \frac{tp}{(tp + fp)}$$

C. Recall

Equation (5.3) is the formula to calculate Recall of the system which is given by:

$$\text{Recall} = \frac{\text{tp}}{\text{Actual Defaced Website Component}} = \frac{\text{tp}}{(\text{tp} + \text{fn})}$$

IV. CONCLUSION

To accurate scrutinizing and checking the integrity of web content is still the challenge for law enforcement agencies. To handle the web defacement cases an effective prototype system has been developed which successfully point out the suspicious activity. The integrity of web contents is checked through CRC32, MD5, SHA512, PSNR and SSIM techniques.

It is found that CRC32 is not fully secure as it cannot prevent intentional alteration of data. As there is no authentication, the data can be altered. MD5 suffers from collision problem, as it provides the same hash value for two web pages with complexity 230. PSNR and SSIM do not support text integrity. The best result is obtained by enforcing SHA512 technique. The work presented in this thesis is focused on effective extraction and matching of web content.

REFERENCES

[1] Yu, W.D.; Nargundkar, S. and Tiruthani, N.; "A phishing vulnerability analysis of web based systems," in Computers and Communication, 2008. ISCC 2008. IEEE Symposium., Marrakech, 2008, pp. 326 - 331

[2] Shar, L.K. and Hee Beng Kuan Tan, "Defeating SQL Injection," in IEEE Computer, Singapore, 2013, Vol. 46, Issue: 3, pp. 68-77.

[3] Yusra A. Y. Al-Najjar and Der Chen Soong, "Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI," in International Journal of Scientific & Engineering Research, Vol. 3, Issue 8, August 2012.

[4] Alain Horé and Djemel Ziou, "Image quality metrics: PSNR vs. SSIM," in Pattern Recognition (ICPR), 2010 20th International Conference, 2010, pp. 2366 - 2369

[5] "Chinese websites 'defaced in Anonymous attack' ", [Online], Available: http://www.bbc.co.uk/news/ technology-17623939, April 5, 2012.

[6] Rick Burgess, "Hackers continue Israel attack, deface website, more credit cards leaked", Available: http://www.techspot.com/news/47046-hackers-continue-israel-attack-deface-website-more-credit-cards-leaked.html, January 13, 2012.

[7] "DOTC site defaced by 'Turkish' hackers", Available: http://www.gmanetwork.com/news/story/277286/scitech/technology/dotc-site-defaced-by-turkish-hackers, October, 2012.

[8] Bartoli, A.; Davanzo, G., and Medvet, E.; "A Framework for Large-Scale Detection of Web Site Defacements", ACM Transaction on Internet Technology (TOIT), Vol. 10, No. 3, Oct. 2010, Article 10.

[9] Kanti, T.; Richariya, V. and Richariya, V.; "Implementing a Web Browser with Web Defacement Detection Techniques", World of Computer Science and Information Technology Journal (WCSIT), Vol. 1, No. 7, 2011, pp. 307-310.

[10] Pan Shi; Xu, H. and Zhang, X. (Luke); "Informing Security Indicator Design in Web Browsers," in proceeding iConference '11 Proceedings of the 2011 iConference, Feb 11, 2011, pp. 569-575.

[11] Staikos, G. 2005. Web Browser Developers Work Together on Security. http://dot.kde.org/1132619164/, Nov. 2005.

[12] Moore, T. and Clayton, R.; "Examining the impact of website take-down on phishing," in Proceedings of the anti-phishing working groups' 2nd annual eCrime researchers summit, Pittsburgh, Pennsylvania, 2007, pp. 1-13.

[13] Peterson, W. W. and Brown, D. T. (January 1961). "Cyclic Codes for Error Detection," in Proceedings of the IRE 49 (1), 1961, pp. 228–235.

[14] Teng Lv and Ping Yan, "A Web Security Solution based on XML Technology," in Communication Technology, 2006. ICCT '06. Int. Conf., 2006, pp. 1 – 4.

[15] Suresh, R.M. and Padmajavalli, R., "An Overview of Data Preprocessing in Data and Web Usage Mining," in Digital Information Management, 2006 1st International Conference, Bangalore, India, 2006, pp. 193 – 198.

[16] Stigge, M. et al., "Reversing CRC – Theory and Practice" Berlin: Humboldt University Berlin Public Rep., May 2006.

[17] Rivest, R.; "The MD5 Message-Digest Algorithm," RFC 1321, Available: http://tools.ietf.org/html/rfc1321.

[18] Stevens, M.M.J.; "On Collisions for MD5", Master's thesis, Mathematics and Computing Science Department, Eindhoven University of Technology, Eindhoven, June 2007.

[19] U.S. National Security Agency, "Secure Hash Standard," in Federal Information Processing Standards Publication 180-2, Aug 2002.

[20] Ndajah, P. et al, "SSIM Image Quality Metric for Denoised Images," in Proc. 3rd WSEAS International Conference on Visualization, imaging and simulation, Japan, September 3, 2010, pp. 53-57.

[21] Wang, Z.; Simoncelli, E.P. and Bovik, A. C.; "Multi-Scalestructural Similarity For Image Quality Assessment," in Proc. 37th IEEE Asilomar Conference on Signals, System and Computation, Pacific Grove , CA , Nov 2009-10.

[22] Huynh-Thu, Q.; Ghanbari, M.; "Scope of validity of PSNR in image/video quality assessment," in Electronics Letters (Volume: 44, Issue: 13), June 19 2008, pp. 800 – 801.

[23] MIT.edu, Available: http://people.xiph.org/~xiphmont/ demo/theora/demo7.html.

[24] Oriani, Emanuele. "qpsnr: A quick PSNR/SSIM analyzer for Linux", April 2011, Available: http:// qpsnr. youlink.org/.

[25] "Opinion Poll", Available: http://en.wikipedia.org/wiki/ Opinion_poll, April 17, 2013.

[26] "The Characteristics of Computers" Available: http://compusystem-tech.blogspot.in/2012/09/the-characteristics-of-computers.html, September 20, 2012]