WASM — A Metric for Securing a Web Application

Rakesh Kumar
Professor, Department of Computer Science and Applications, Kurukshetra University Kurukshetra, India
Email: rsagwal@rediffmail.com

Gurvinder Kaur
Director, Guru Nanak Khalsa Institute of Technology and Management Studies, Yamuna Nagar, India
Email: allagh.gurvinder@gmail.com

Internet is a medium to connect millions of computers which share and access information all over the world. With the evolution of the web and its increased use in every aspect of life, the need for web security has become imperative. As websites opt for commercial viability, the threat of hackers, viruses, or annoyance attacks becomes more pronounced. Organizations face several security-related challenges. If organizational information is hacked either through the network or through other means, it could incur a heavy cost to the company. A failure in network security could also cost the organization in terms of its goodwill and reputation. This paper identified common threats on the web and classified these threats into various categories, such as accidental, malicious, authorization, application, privacy, and access control threats. This also highlights the three main areas in which web can be secured ie. client side threats, server side threats and network side threats. This paper discusses the primary goals and objectives of security contained within the CIA Triad: Confidentiality, Integrity and Availability. Different types of attackers which are responsible for security of web are also depicted. This paper shows different attacks related to client side, server side and network side threats. Client-side Security threats are classified into: Cross Site Scripting, Cross Site Request Forgery, Broken Authentication and Session Management, Security Misconfiguration and Failure to Restrict URL Access. Server-side Security consists of Structured Query Language (SQL) Injection, Malicious File Execution, Insecure Direct Object Reference, Insecure Cryptographic Storage and Unvalidated Redirects and Forwards. The network threats highlighted are Denial of Service (DoS), Insufficient Transport Layer Protection, Eavesdropping, Data Modification, IP Address Spoofing, Sniffer attacks*, Man-in-the-Middle Attack, Phishing,* Brute force attack *and TCP Session Hijacking. The paper shows the causes of each of the attacks and the web application metrics which were earlier defined are also highlighted. A metric named Web Application Security Metric (WASM) is proposed in this regard to make the web page secure. This metric calculates the sum of the weight of the categories like: Input validation, Authentication, Authorization, Configuration management, Sensitive data, Session management, Cryptography, Parameter manipulation, Exception management and Auditing and logging.*

ACM Classification: *H.5.3*

Keywords: *Client, Network, Server, Web Security*

1. Introduction

A number and variety of threats to IT systems multiplying in daily routine cause a lot of risk for the software companies. Businesses that trade electronically are particularly vulnerable to risks such as

Manuscript received: 1 October 2012
Communicating Editor: Francisco Garcia-Sanchez

fraud or breaches of confidentiality, causing loss of assets and damage to their reputation. Web applications have become increasingly vulnerable to different forms of hacker attacks. According to a Gartner Report, 75% of attacks today occur at the application level. According to Forrester "people are now attacking through applications, because it's easier than through the network layer". Symantec documented 56 vulnerabilities in the second half of 2006, 66% of which affected web applications. Due to this industry the stress is laid on the security of the web applications along with the security of the underlying computer network and operating systems. The vulnerabilities in a web application are due to inadvertent flaws left behind during development, security issues in the underlying environment and misconfigurations in one or more components like database, web server etc.

Web security or web application security or webappsec is a set of procedures, practices, and technologies assuring the reliable, predictable operation of web servers, web browsers, other programs that communicate with web servers, and the surrounding internet infrastructure. Web security comprises of three parts – a) how the objects and resources can be named securely b) how secure authenticated connections can be established c) what happens when a website sends a client a piece of executable code?

1.1 Web Security Attacks

Due to web environment there are three areas which need to be secured. They are: (i) client, (ii) server and (iii) network. The client sends a request to the server by entering a URL in a web browser. The web server sends the requested web page to the client. The browser then displays the web page on the screen. This interaction between the client and the server is facilitated by several networks that link all the computers on the net. Designing and building a secure web application requires the knowledge of threats. A threat may cause an unauthorized disclosure, maneuvering, disruption, or destruction of information and systems.

2. Client-side Security Threats

Client-side vulnerability is a form of un-patched software on a workstation, desktop or laptop. Client-side Security threats comprise the following threats:

1. **Cross-Site Scripting (XSS)** – This is the most prevalent web application security flaw that helps the attackers to insert client-side script into web pages viewed by other users. XSS flaws comprise three types:
   1) *Stored or Persistent XSS Attacks* – Stored attacks are those where the injected code is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim can recover the malicious script from the server when requested.
   2) *Reflected XSS Attacks* – These attacks are those where the injected code is reflected off the web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request.
   3) *Document Object Model (DOM) Based XSS* – Here the attack payload is executed as a result of modifying the DOM "environment" in the victim's browser used by the original client side script to make the client side code run in an "unexpected" manner (Klein, 2005).

   There are certain causes leading to XSS attack. They are:

   1) HTTP header injection created due to escaping problems on HTTP protocol level thus,

enabling attacks such as HTTP response splitting (Adobe Systems, 2006).

2) It occurs when an application allows data that is not validated or escaped properly to be sent to a web browser.
3) It can occur when malicious scripts are injected into the benign and trusted websites.
4) XSS is caused by the failure of a web based application to validate user supplied input before returning it to the client system.
5) XSS is also caused by missing input validation. Input validation refers to the process of validating all the input to an application before using it.

2. **Cross-Site Request Forgery (CSRF) –** Cross-Site Request Forgery also known as one-click attack or session riding attack is a malicious exploit of a website where unauthorized commands are transmitted from a user that the website trusts (Ristic, 2005).

   Cross-site request forgery occurs due to the following reasons:

   1) It is carried out from the user's IP address (Ristic, 2005).
   2) This attack occurs when there is no authorization check for vulnerable actions.
   3) CSRF attacks occur when a malicious website causes a user's web browser to perform an unwanted action on a trusted site. These attacks are called the "sleeping giant" of web-based vulnerabilities (Grossman, 2006).
   4) It also occurs when a web application uses session cookies.
   5) It occurs when the application acts on an HTTP request without verifying that the request was made with the user's consent.

3. **Broken Authentication and Session Management –** Authentication and session management handles authentication of user and manages the active sessions. User authentication on the web makes use of a userid and password. Web applications establish sessions to keep track of the stream of requests from each user. This includes solid authentication mechanisms which can be undermined by flawed credential management functions, including password change, forgot my password; remember my password, account update, and other related functions.

   The causes of authentication and session management are:

   1) Session management vulnerabilities occur when user sensitive information like user names, passwords, and session tokens are not protected by the developers.
   2) Broken authentication vulnerabilities occur when developers fail to use authentication methods that have been adequately tested and rely on their own, often flawed, method for authenticating users.
   3) These attacks occur when a session ID is visible to others, timeouts are not properly set, SSL/TLS is not used, or any other flaw in the authentication scheme is detected.

4. **Security Misconfiguration –** Security misconfiguration, or poorly configured security controls allows the malicious users to change the website, obtain unauthorized access, compromise files, or perform other unintended actions.

   Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, framework, and custom code.

5. **Failure to Restrict URL Access –** This is the attack when a page doesn't have the correct access control policy in place and unauthorized users can view the content that they shouldn't have the ability to view.

The main cause of this attack is that it takes place when an authorized user can simply change URL to access a privileged page.

3. Server-side Security threats

The essential part of the web that needs to be secured is the host. Host security refers to the security of computers on which severs are running. It is important to secure internet servers because they host all essential services being accessed from different parts of the world. Servers are the single connecting point for millions of people across the world. If an attacker manages to get control over any server on the internet, it may result in the leakage of crucial information, disruption of services, and malicious content sent to users. Server-side security comprises such threats as:

1. **Structured Query Language (SQL) Injection –** SQL Injection being one of the most common application layer attack techniques used is an attack where malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed.

   The causes of SQL Injection are:

   1) *Incorrectly filtered escape characters* – This form of SQL injection occurs when user input is not filtered for escape characters and is then passed into an SQL statement.
   2) *Incorrect type handling* – This form of SQL injection occurs when a user supplied field is not strongly typed or is not checked for type constraints.
   3) *Blind SQL injection* – Blind SQL Injection is used when a web application is vulnerable to an SQL injection but the results of the injection are not visible to the attacker.
   4) These attacks occur when untrusted data, such as a query, command or argument, is sent to an interpreter.

2. **Malicious file execution –** Uploaded files or other data feeds don't appear like they should These files can contain a malicious payload, thus, should not be stored in web accessible locations. Malicious file execution vulnerabilities can be found in many applications.

   The occurrence of malicious file execution depends upon:

   1) It occurs when all web application frameworks accept filenames or files from the user.
   2) It is caused when hostile data is uploaded to session files, log data, and via image uploads.
   3) It is also caused when compression or audio streams are used which do not inspect the internal PHP URL flag and thus allow access to remote resources even if allow_url_fopen or allow_url_include is disabled.

3. **Insecure Direct Object References –** Insecure Direct Object Reference is when a web application exposes an internal implementation object to the user like database records, URLs, or files.

   Attack occurs when an authorized user can change a parameter value that refers to a system object that they are not authorized for.

4. **Insecure Cryptographic Storage –** Web applications frequently use cryptographic functions to protect information and credentials. Protecting sensitive data with cryptography has become a key part of most web applications.

   The main causes of Insecure Cryptographic Storage are:

   1) It occurs when an application doesn't securely encrypt its sensitive data when it is stored

into a database.

2) This attack can result from the poor use of encryption algorithms such as using home grown algorithms, insecure use of strong algorithms or the continued use of proven weak algorithms.

3) The use of weak or unsalted hashes to protect passwords is another common flaw that leads to this risk.

5. **Unvalidated Redirects and Forwards –** Applications frequently redirect users to other pages, or use internal forwards in a similar manner. The target page is specified in an unvalidated parameter which allows the attackers to choose the destination page.

   Unvalidated redirects and forwards occur when attackers redirect users to other pages and websites such as phishing or malware sites or use forwards to access unauthorized pages.

4. Network-side Security threats

A network attack can be defined as any method, process or means used to maliciously attempt to compromise the security of the network.

1. **Denial of Service (DoS) –** DoS or distributed denial-of-service attack (DDoS attack) is a threat that prevents legitimate users getting access to the information or resource that they need, when they need it. There are two general forms of DoS attacks: (a) those that crash services and (b) those that flood services (Erikson, 2008). The five basic types of attack are:

   (i) Consumption of computational resources, such as bandwidth, disk space, or processor time.

   (ii) Disruption of configuration information, such as routing information.

   (iii) Disruption of state information, such as unsolicited resetting of TCP sessions.

   (iv) Disruption of physical network components.

   (v) Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

   The reasons behind the occurrence of this attack are:

   1) A distributed denial of service attack (DDoS) occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers (Wikipedia, 2010).

   2) *Non-intentional Causes* – Causes can be accidental as well as deliberate.

   3) *Deliberate Causes* – The deliberate threat can be further categorized into 'deliberate-authorized' and 'deliberate-unauthorized' incidents. The latter category includes the kinds of attacks most commonly associated with DoS.

2. **Insufficient Transport Layer Protection –** Insufficient transport layer protection allows communication to be exposed to untrusted third-parties, providing an attack vector to compromise a web application and/or steal sensitive information.

   This attack occurs when a site does not use SSL/TLS for pages that require authentication where an attacker can monitor network traffic to steal an authenticated user's session cookie.

3. **Eavesdropping –** Network Eavesdropping or network sniffing is a network layer attack consisting of capturing packets from the network transmitted by others' computers and reading the data content in search of sensitive information like passwords, session tokens, or any kind of confidential information.

This type of network attack occurs when an attacker monitors or listens to network traffic in transit, and then interprets all unprotected data.

4. **Data Modification –** Data modification or data manipulation pertains to a network attack where confidential company data is interpreted, deleted, or modified.

   This type of attack occurs when an unauthorized party intercepts data in transit, alters it, and retransmits it.

5. **IP Address Spoofing –** IP address spoofing attack identifies computers on a network.

   IP address spoofing occurs when an attacker assumes the source Internet Protocol (IP) address of IP packets to make it appear as though the packet originated from a valid IP address.

6. **Sniffer attacks –** A sniffer is an application or device that can read, monitor, and capture network data exchanges and read network packets. If the packets are not encrypted, a sniffer provides a full view of the data inside the packet.

   Network Sniffing attacks occur when an individual is observing network traffic. Typically, any system on a network sharing a transmission medium has the ability to view other system traffic.

7. **Man-in-the-Middle Attack –** Man-in-the-middle attack (MITM) or bucket-brigade attack, or Janus attack, is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

   For a man-in-the-middle (MITM) attack to be successful, the following sequence of events has to occur:

   1) The hacker must be able to obtain access to the communication session to capture traffic when the receiver and sender establish the secure communication session.
   2) The hacker must be able to capture the messages being sent between the parties and then send messages so that the session remains active.
   3) A man-in-the-middle attack can succeed only when the attacker can impersonate each end-point to the satisfaction of the other – it is an attack on mutual authentication.
   4) A man-in-the-middle (MITM) attack occurs when a hacker eavesdrops on a secure communication session and monitors, captures and controls the data being sent between the two parties communicating.

8. **Phishing –** Phishing is a way of attempting to acquire sensitive information such as usernames, passwords and credit card details by masquerading as a trustworthy entity in an electronic communication.

   A phishing attack occurs through a process outside the control of the spoofed company.

9. **Brute force attack –** Brute force attacks simply decode a cipher by trying each possible key to find the correct one. This type of network attack systematically uses all possible alpha, numeric, and special character key combinations to find a password that is valid for a user account.

   A brute-force attack occurs when a potentially malicious user tries all possible keys and guesses what the encryption key is.

10. **TCP Session Hijacking –** TCP hijacking is one of the most popular techniques for intruders to gain unauthorized access to internet servers. This is a technique that involves intercepting a

TCP session initiated between two machines in order to hijack it.

TCP session hijacking is when a hacker takes over a TCP session between two machines. As most of the authentications occur at the start of a TCP session the hacker can gain access to a machine.

5. Metrics for Security of Web Application

Ten metrics were defined according to NIST 800-55 standard (Swanson *et al*, 2010) that were used to measure 17 Information Technology (IT) security issues:

1. Percentage of systems that had formal risk assessments performed and documented;
2. Percentage of total systems for which security controls have been tested and evaluated in the past year;
3. Percentage of total systems that have the costs of their security controls integrated into the life cycle of the system;
4. Percentage of total systems that have been authorized for processing following certification and accreditation;
5. Percentage of current security plans;
6. Percentage of systems that have a contingency plan;
7. Percentage of systems for which contingency plans have been tested in the past year;
8. Percentage of employees with significant security responsibilities who have received specialized training;
9. Percentage of agency components with incident handling and response capability;
10. Number of incidents reported externally to law enforcement.

Different metrics were also defined which comprised of few areas like:

1. Users
2. Cookies
3. Services (Email, FTP and others),
4. Communication,
5. Input validation and
6. System devices.

Manadhata and Wing (2005) also proposed a metric: a) An Attack Surface Metric defined in terms of the system's attackability and b) Three abstract dimensions: method, data, and channel.

Jaquith (2004) also defined five metrics such as:

Metric 1: Baseline Defenses Coverage (Antivirus, Antispyware, Firewall, and so on) – This helped in protecting the enterprise against the most basic information security threats.

Metric 2: Patch Latency – Patch latency is the time between a patch's release and successful deployment of that patch.

Metric 3: Password Strength – This metric offers simple risk reduction by sifting out bad passwords and making them harder to break, and finding potential weak spots where key systems use default passwords.

Metric 4: Platform Compliance Scores – Widely available tools, such as the Center for Internet Security

(CIS) scoring toolset, can run tests against systems to find out if the hardware meets best-practice standards such as those set by CIS.

Metric 5: Legitimate Email Traffic Analysis – Legitimate email traffic analysis is a family of metrics including incoming and outgoing traffic volume, incoming and outgoing traffic size, and traffic flow between the company and others.

Subramanian (2011) also defined metrics in two categories:

1. The first set of metrics is for incidents and vulnerabilities and
2. The second set is for the application security program itself.

According to Young, there are three major types of security/privacy metrics (Young, 2012):

a) Compliance Metrics – It measures compliance with current security and privacy regulations and standards,

b) Resilience Metrics – It measures the resilience of controls relating to physical security, personnel security, IT security, and operational security both before and after a product, system or network is deployed and

c) Return on investment (ROI) Metrics – It measures the ROI in physical, personnel, IT, and operational security controls to guide capital investment.

6. Proposed Web Application Security Metric (WASM)

The factors affecting the security of web applications have been categorized into the following 10 categories as shown in the table given below:

1. Input validation
2. Authentication
3. Authorization
4. Configuration management
5. Sensitive data
6. Session management
7. Cryptography
8. Parameter manipulation
9. Exception management and
10. Auditing and logging.

Every category is assigned the maximum weight 1 (Although the weight may vary from 0 to 1 depending upon the factors listed in the Table below).

WASM is the sum of the weight (Ci) of all the listed 10 categories. The maximum weight of a category can be 1.

$$WASM = \Sigma Ci$$

Where Ci is the product of Category Weight Adjustment Factor where every CWAF can be assigned a number within the range .5 to 1 depending upon the fact that the implementation of the factor is rated very low, low, nominal, high, very high or extra high. If any CWAF is present, it is assigned minimum value .5 and if not present then the value will be .1. The objective of the designer should be to maximize WASM value.

Category	Guidelines
Input Validation	a) Assume all input is malicious b) Consider centralized input validation c) Do not rely on client-side validation d) Be careful with canonicalization issues e) Constrain, reject, and sanitize input f) Validate for type, length, format, and range
Authentication	a) Use strong passwords b) Support password expiration periods and account disablement c) Do not store passwords in user stores d) Encrypt communication channels to protect authentication tokens e) Pass Forms authentication cookies only over HTTPS connections f) Do not send passwords over the wire in plaintext
Authorization	a) Use least privileged accounts b) Consider authorization granularity c) Enforce separation of privileges d) Restrict user access to system-level resources e) Use multiple gatekeepers
Configuration Management	a) Use least privileged process and service accounts b) Do not store credentials in plaintext c) Use strong authentication and authorization on administration interfaces d) Do not use the Local Security Authority (LSA) e) Secure the communication channel for remote administration. f) Avoid storing sensitive data in the Web space.
Sensitive Data	a) Avoid storing secrets b) Encrypt sensitive data over the wire c) Secure the communication channel d) Provide strong access controls on sensitive data stores e) Do not store sensitive data in persistent cookies f) Do not pass sensitive data using the HTTP-GET protocol
Session Management	a) Limit the session lifetime b) Secure the channel c) Encrypt the contents of authentication cookies d) Protect session state from unauthorized access

Table continued on following page

Category	Guidelines
Cryptography	a) Do not develop own cryptography b) Use tried and tested platform features c) Keep unencrypted data close to the algorithm d) Use the right algorithm and key size e) Avoid key management f) Use Data Protection API (DPAPI) g) Cycle the keys periodically h) Store keys in a restricted location
Parameter Manipulation	a) Encrypt sensitive cookie state b) Do not trust fields that the client can manipulate such as query strings, form fields, cookies, or HTTP headers c) Validate all values sent from the client
Exception Management	a) Use structured exception handling b) Do not reveal sensitive application implementation details c) Do not log private data such as passwords. Consider a centralized exception management framework.
Auditing and Logging	a) Identify malicious behavior b) Audit and log activity through all of the application tiers c) Secure access to log files d) Back up and regularly analyze log files

Table: Category Weight Adjustment Factor (CWAF)

7. Conclusion

Today, our lives and organizations revolve around computerized systems, such as payroll accounting, inventory control, and transaction-processing systems. Life across the world would come to a halt without computerized systems that store volumes of information in databases. What would happen if a person with malicious intentions accessed this information? As use of the internet spreads, the potential for misusing it has increased. A security breach incurs a cost for the organization in terms of money as well as goodwill. This paper also helps to identify the threats. Identification and illustration of various server/client/network side threats facing the network, host, and application layers are elaborated. A metric is proposed for the security of the web page. It is, therefore, important to secure and make the data available to an organization at all times. It is also advisable to use some kind of vulnerability scoring to be able to compare the security level of subsequent versions of the same application or even different applications.

8. References

ADOBE SYSTEMS (2006): Update available for potential HTTP header injection vulnerabilities in Adobe Flash Player. November 14.

Cross-Site Scripting, Web Application Security Consortium, February 23rd, 2004 http://www.webappsec.org/projects/threat/classes/cross-site_scripting.shtml

Cross Site Scripting (XSS) Flaws, The OWASP Foundation, updated 2004 http://www.owasp.org/documentation/topten/a4.html

ERIKSON, J. (2008): *Hacking the art of exploitation* (2nd edition Ed.). San Francisco: No Starch Press. ISBN 1-59327-144(1): 251.
GROSSMAN J. (2006): *CSRF, the sleeping giant*. http://jeremiahgrossman.blogspot.com/ 2006/09/csrf-sleeping-giant.html, September.
http://www.webappsec.org/projects/articles/071105.shtml
http://www.applabs.com/html/TheInternetSecurityRisks_748.html MANADHATA, P. and WING, J.M. (2005): *An Attack Surface Metric*, July.
JAQUITH, A. (2004): *A Few Good Information Security Metrics*.
KLEIN, A. (2005): *DOM Based Cross Site Scripting or XSS of the Third Kind* (WASC writeup), July.

MANADHATA, J.M. and WING, M. (2005): *An Attack Surface Metric*, July.
RISTIC, I. (2005): *Apache Security*. O'Reilly Media. ISBN 0-596-00724-8, 280.
SUBRAMANIAN, S, (2011): *Measure and Monitor Application Security*.
SWANSON, M., BARTOL, N., SABATO, J., HASH, J. and GRAFFO, L. (2010): Security Metrics Guide for Information Technology Systems, *NIST Special Publication (SP)* 800-55, accessed January.
WIKIPEDIA (2010): Denial-of-service attack – Wikipedia, the free encyclopedia (n.d.). *Wikipedia, the free encyclopedia*. Retrieved November 8, (2010).
YOUNG, B. (2012): CS378: *Information Assurance and Security Metrics for IA*, July 2.

Biographical Notes

Dr Rakesh Kumar *obtained his BSc degree, Master's degree – Gold Medalist (Master of Computer Applications) and PhD (Computer Science & Applications) from the Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, India, where he is currently a professor. He has 23 years of teaching experience. His research interests are in genetic algorithm, software testing, artificial intelligence, and networking. He has published approximately 70 research papers in various national/international journals and conferences. He is a senior member of the International Association of Computer Science and Information Technology (IACSIT).*


Dr Rakesh Kumar

Dr Gurvinder Kaur*, Director, Guru Nanak Khalsa Institute of Management Studies, Yamuna Nagar, Haryana, India, has 13 years of teaching experience. Her qualifications include BSc (Computer Science), MA (English), MSc (Information Technology), MCA, M.Phil (Computer Science) and PhD (Computer Science & Applications) from the Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, India. Her areas of interest are software engineering, mobile computing, software testing and web applications. She has published approximately 15 research papers in various national/international journals and conferences.*


Dr Gurvinder Kaur