Proposed Secure Hypertext Model in Web Engineering

Madhuri N. Gedam* and Bandu B. Meshram

Department of Computer Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, India
E-mail: madhuri.gedam@gmail.com; bbmeshram@vjti.org.in
**Corresponding Author*

Received 09 March 2023; Accepted 21 May 2023;
Publication 24 October 2023

Abstract

Secure web application development is one of the prime challenges for the software industry. In the last decade, web applications have rapidly developed but web engineering methods have some limitations while designing web applications. The extensive literature survey explores various concepts like web engineering, hypertext modelling, web applications hypertext modelling methods, attacks on web applications, same origin policy (SOP) and cross origin resource sharing (CORS). The complexity of web pages is a major concern for security. The proposed secure hypertext model (SHM) provides hypertext modelling of web applications and helps in the identification of attacks on hypertext links. It provides security stereotypes and precisely specifies vulnerability defences in web application design. This standardized attack vector and defence mechanism will help developers to build more secure applications.

Keywords: Web engineering methods, WebML, UWE, UML, IFML, web applications, hypertext modelling.

Journal of Web Engineering, Vol. 22_4, 575–596.
doi: 10.13052/jwe1540-9589.2241

1 Introduction

Web applications can be built with superior quality with the help of processes used in web engineering apart from software engineering. Mainly, the focus of software engineering is on developing a conventional desktop application whereas web engineering focuses on web applications. There has been rapid growth in web applications in recent years. The development methodologies used in two different types of applications such as web applications and software applications differ in several ways [1–3, 9]. Since web engineering mainly uses the principles and management activities found in software engineering processes, it is the extension of software engineering [11, 12].

The main purpose of web applications is to publish and maintain huge amounts of data over the internet. Web engineering includes methods and tools required for the development of web systems in a smooth manner covering the navigation structure, security aspects, and faster roll out of required changes than other software [35]. The developments in web engineering takes place in three stages: requirements, analysis/design, and implementation. As every web engineering method is designed with a specific purpose, any single web engineering method does not cover the entire life cycle in web engineering. The selection of a particular methodology considering the dynamic characteristic features of web systems and complexity of web applications poses a major challenge for the design of web applications.

Web software systems can be designed using different types of modelling like content, hypertext and presentation modelling. Hypertext modelling specifies the navigation through the content of web applications. It carries out a logical composition of a site such as web pages into nodes and the navigation structure into internal, external, and inbound and outbound links in modern web applications. The modelling can be done using various web methods like WebML, UWE, OOHDM, HDM, Secure UML and the like. Many web engineering methods are available with their unique features but a complete development life cycle is not covered by any one method as a whole [4]. The security of links while designing web software systems is of prime concern to avoid attacks on hypertext links and web spoofing. This paper is focused on the security of hypertext links using various stereotypes and notations in the proposed secure hypertext modelling.

The paper is organized as follows. Section 2 describes a detailed literature survey of hypertext modelling, web modelling methods, web-based client server communication, and security of software in a web environment. Section 3 discusses the proposed secure hypertext model. Section 4 concludes the results.

2 Literature Survey

This section explores various concepts on web engineering methods, attacks on links, same origin policy (SOP) and cross origin resource sharing (CORS).

2.1 Web Engineering

While designing web software systems, different types of modelling are used such as:

- **Content Modelling.** The information and functional requirements gathered during the requirement engineering phase is transferred to a class model and state transition diagram.
- **Hypertext Modelling.** This specifies the navigation through the content of web applications. Site navigations are specified through links. It is also known as navigation modelling and does navigation of a site using links through the content of web applications [2–4].
- **Presentation Modelling.** This specifies the look and feel of the web applications.

Various web engineering methods are available for modelling web applications. This paper mainly focusses on hypertext modelling.

2.2 Hypertext Modelling

This is also known as navigation modelling. The most important property of non-linearity is to be considered while modelling web application. The contents are structured into nodes such as web pages and links are used to establish the relationship between these nodes. It does logical composition of web pages and the navigation structure. It specifies the navigability through the content of the web application [18, 26]. The hypertext modelling generates the following.

1. Hypertext structure model

This is based on the concept of hypertext, i.e., node and the link between these nodes. It is also known as navigation structure model and used to define navigation throughout nodes but not orientation. It defines the structure of the hypertext covering navigation among classes of the content model. It transforms classes and objects from a content model into nodes in the hypertext. The node is also sometimes called the navigational view due to the capability of selecting one or more object from the content model [25].

Hyperlinks are directional links between a source and a target web page for navigation purposes on the same page or different sections of the same web page. It is done with the help of text strings, buttons, graphics or video being activated on a mouse click. Navigation in the hypertext modelling can be done in following ways.

Intra-page navigation occurs in a single web page within different sections of the same web page using hyperlinks.

Inter-page navigation involves calling another target web page using hyperlinks.

Frame-based navigation involves navigation between different frames in a single web page calling separate web pages. The called web pages may contain again frames.

Navigation of multiple windows in a browser differs from frames. They can be created upon navigation or closed with the help of an operating systems windows manager [23].

2. Access Model

A hypertext structure model is not sufficient alone to describe navigation between the nodes; the users need navigation and orientation aids through access structures refining the hypertext structure model. It provides the flexibility of linking of class content with hyperlinks in a non-linear fashion. The following access structure is used in access models [26]. The following elements are used to serve the purpose in access model:

<<index>> User is allowed to select objects of the same type
<<menu>> User is allowed to access heterogeneous nodes or submenus
<<guided tour>> Sequential links through number of nodes
<<query>> User is allowed to search for nodes
<<home>> Directs to the home page
<<landmark>> Directs to a node that can be reached from within all nodes.

Hypertext modelling can be done using various web engineering methods. They are classified according to their purpose like data oriented, hypertext oriented, object oriented, and software oriented methods. Many issues are within web engineering methods. The entire development life cycle is not covered by any single method in depth. Hence an appropriate hypertext model and method is required to communicate between heterogeneous members of hypermedia projects [12].

2.3 Web Application Hypertext Modelling Methods

The following are the web modelling methods for web development purposes [2, 3].

1. WebML

WebML is used to design web applications with huge relational database management systems. It is used to design the web application's structure and presentations of contents with hypertext [32]. It specifies contextual links carrying context information, non-contextual links having no related context information, intra-page links to navigate on the same web page, and inter-page links to navigate to other web pages [6, 10].

2. HDM

HDM is used as a method for modelling high level specifications of new or existing web applications. The smallest collection of information called units have standard qualities of hypertext nodes. The grouping of units together forms components. The components are subset of entities having sizable structures of information chunks making an application [26, 36]. Hypertext links have a representational role and a navigational role. It specifies three types of links like perspective links to connect various views of a node in relation to each other, structural links to connect components belonging to the same entity, and application links represent domain dependent relationships among entities, or their components.

3. UWE

UWE is a visual modelling method based on standards and an extension of UML providing a CASE tool which supports UML notations to model design and automatic creation of web applications [18, 30]. It covers the whole life cycle of web application development. It uses UML stereotypes for access structures such as <<menu>>, <<index>>, <<query>> and <<guided tour>>. Stereotypes <<navigation>> and <<web process>> are used to model navigation and process aspects respectively [22]. Stereotypes <<navigation class>> and <<process class>> are used to represent nodes. Stereotypes <<menu>>, <<navigation link>> and <<process

link>> are used for selection of a navigation path and the navigation paths between nodes.

4. OOH

The object-oriented hypermedia (OOH) method has an object-oriented focus providing semantics and notations necessary for the new web application development as well as linking with existing web applications. This method defines five types of link such as internal link (I-links) for navigation among the nodes within boundaries, traversal-links (T-links) point to nodes covering other navigational requirements, requirement links (R-links) point to a start of a navigational path, external links (X-links) point to external nodes and service links (S-link) point to services [24].

5. UMLSec

UMLSec is an extension of UML in which security requirements are fulfilled using 21 numbers of stereotypes, 9 types of constraints and 7 types of tags [16, 17]. The stereotypes are used for representing fair exchange of data, non-repudiation of data, role-based access control, secure communication link, confidentiality, integrity and authenticity. These security features are used in use case diagrams, class diagrams, state charts, sequence diagrams, activity diagrams, and deployment diagrams [40].

2.4 Hypertext Modelling for a College Management System

A college management system (CMS) is a private internal web-based application that can be accessed inside an institution or a specified college by anyone, anywhere and from any device. The basic principle in the implementation of CMS is to make supervision of the college easier and to explore all the academic activities inside the college on a single click. It manages the details of students, faculties, class details, course details, etc. This system serves as a tool for better communication among students, faculties and administration staff for all kind of activities like curricular, extracurricular and social.

A CMS web application has a common home page for all with login, contact us and about us classes. The classes are called through navigation links without any security stereotypes, as shown in Figure 1. After successful login with valid credentials, the user will be able to navigate to different classes like add, update and view details using navigation links. The user will be able to log out of the application and return to home page using log

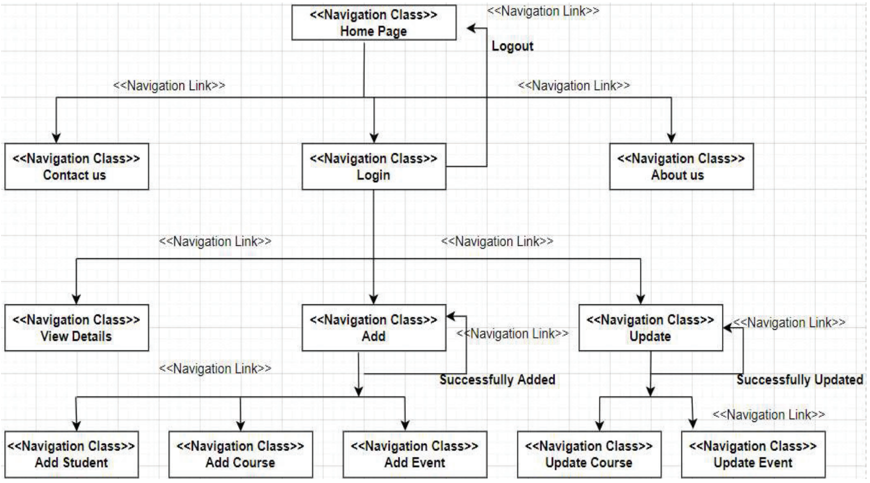

Figure 1 Hypertext model of a college management system.

out navigation link. The complete hypertext model is designed with absence of security stereotypes.

2.5 Attacks on Web Applications

The web applications being hosted in the public domain possess threats to the different components of web systems such as web server, application server, database server, browsers, network, etc. It has become known that more than 80% of websites in the public domain have a minimum of one potential vulnerability [8]. Web based communication takes place as shown in Figure 2.

As shown in Figure 2, the client-side browser request is sent to the DNS for domain name resolution and receipt of an IP address to make a TCP connection with the client. The HTTP request sent by the client is sent to the web server using TCP/IP. The HTTP request is served by an operating system on the web server. The HTTP software running on the web server transfers this HTTP response to the TCP/IP software running on the web server which breaks the HTTP response into packets and sends it over the TCP connection to the client. The TCP/IP software on the client computer checks the correctness of the packets and reassembles them to form the original web page in HTML format. It informs the HTTP software on the server that the page was received correctly [8]. HTTP protocol being stateless, the session state can be maintained at either the client-side using cookie,

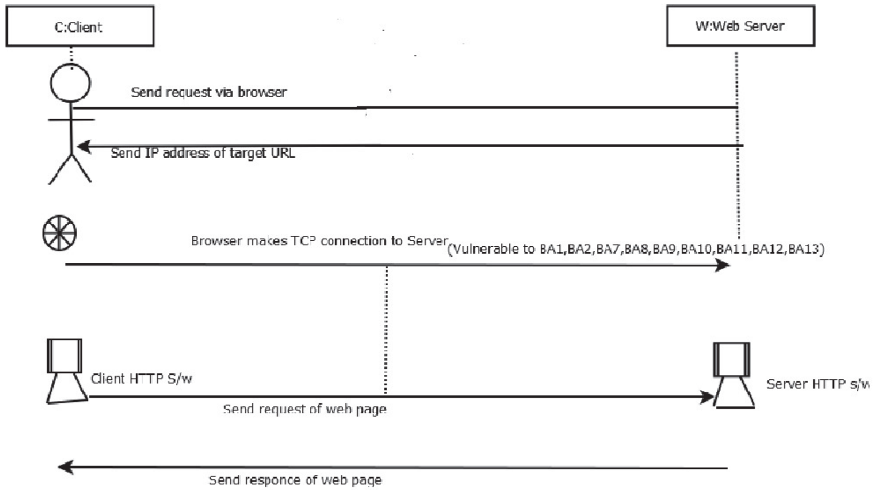
C:Client
W:Web Server
Send request via browser
Send IP address of target URL
Browser makes TCP connection to Server(Vulnerable to BA1,BA2,BA7,BA8,BA9,BA10,BA11,BA12,BA13)
Client HTTP S/w
Server HTTP s/w
Send request of web page
Send responce of web page

Figure 2 Web based communication.

hidden form or URL rewriting, or at the server side. However, the client plays an important role in maintaining the states of a web application.

The web applications are vulnerable to internal and external attacks like browser based, operating system level and database level due to its presence in the internet [19]. These attacks make the hypertext model insecure, as shown in Table 1.

2.6 Same Origin Policy and Cross-origin Resource Sharing

The same origin policy (SOP) is an important browser level security mechanism to restrict access of a document from another origin maintaining confidentiality and integrity. It allows scripts running on pages originating from the same site which can be a combination of the protocol, domain and port. SOP restrictions can be imposed using iframe to confine untrusted domain data to their boundaries, parseJason() to check for non-JSON strings, using the HttpOnly cookie attribute, captcha, cryptographic tokens, etc.

The SOP is able to be bypassed with the use of JavaScript, Ajax, web services, mashups and steal passwords, cookies, log keystrokes and alter information. This has led to attacks such as cross-site scripting (XSS), cross-site request forgery (CSRF), web cache poisoning. Presently, SOP doesn't support the separation of privilege and least privilege access control principles. Ajax proxy and JavaScript object notation with padding (JSONP) script tag injection are able to bypass SOP and fetch the data from other domains.

Table 1 Web application attacks

1. Browser based attacks (BAi): Attackers collect information from a web browser to target browser based attacks due to traces left behind by a user on a computer during web browsing. These footprints are usually traced out in the browser history, cookies, cache, downloaded files, temporary files, etc.	
BA1	Browser cache poisoning: Collection of critical information from the browser cache.
BA2	Man-in-the-middle attack: The attacker sniffs the communication between victim and the intended web application.
BA3	Passwords in browser memory: Attacker can steal the stored login details of a user in a browser memory and escalate his privilege.
BA4	Back & refresh attack: Collecting sensitive and important details by the use of the back button and the browser's refresh feature.
BA5	Autocomplete: The attacker can easily get hold of a saved password in the browser by unaware users.
BA6	Browser history: Leakage of sensitive information through the URL from the browser's history.
BA7	Cross-site scripting: malicious script injection by an attacker into a web page leading to a target.
BA8	CSRF: An attacker tricks a victim into performing actions on their behalf.
BA9	Eavesdropping attack: The attacker sniffs the unprotected network between victim and intended web application and monitors HTTP requests, DNS queries, etc. to prepare for an attack.
BA10	UI redressing – The attacker redresses a target application to confuse the user. It is also known as clickjacking or tap jacking.
BA11	Session Hijacking: The attacker steals the authenticated session from the victim's browser.
BA12	Session fixation: The attacker makes the user use the browser's existing session forcefully which is stolen by the attacker.
BA13	Social engineering attacks: The attackers use manipulation techniques to make people prey to their attack vectors and disclose confidential information.
2. Operating system attacks (OSi): Operating system (OS) security is the process of ensuring confidentiality, integrity and availability of the OS. It refers to the particular action taken to protect the operating system from viruses, worms, threats, malware, remote hacker intrusions [36].	
OA1	Virus: Designed to spread from host to host and has the ability to replicate itself on a user's click.
OA2	Worms: Standalone malware program replicating itself to infect other systems using a computer network on its own.
OA3	Trojan horse: Affects data or network from remote location by damaging, disrupting, stealing, or doing some harmful action [35].
OA4	Backdoor: Acts as an entry point to perform malfunctioning and perform actual attack [37].

(*Continued*)

Table 1 Continued

OA5	Botnet: These are robots over the network in the form of computer programs to perform data theft, perform DDoS, flooding with spam mails and illegitimate access to the device.
OA6	Port scanning: Attackers used to find vulnerable entry points in the networks such as open ports to launch attack on the system [37].
OA7	Denial of service: Generally prevents actual authorised users to access the system [37, 38].
3. Database attacks (DAi): Database security refers to the complete protection of data from tampering and unauthorized intrusion due to its most critical nature and being vulnerable to attacks demands highest level of protection.	
DA1	Excessive privileges: Authorized users with unwanted database privileges may tamper with data [38, 39].
DA2	SQL injections: Entering malicious queries into the input fields of web forms.
DA3	NoSQL injection: Entering malicious queries into big data components like Hive, MapReduce [39].
DA4	Weak audit trail: Weak auditing of database activities [38, 39].
DA5	Backup exposure: Physical security breaches like theft of database backup disks and tapes.
DA6	Weak authentication: No provision of multiple factor authentication enabling attackers to access the database easily.
DA7	Database vulnerabilities and misconfiguration: Un-patched databases, default accounts, default configuration parameters.
OWASP top 10 attacks 2021	
A1	Broken access control: Leads to unauthorized disclosure of information, manipulation or destruction of data and harming integrity of data.
A2	Cryptographic failures: Leads to sensitive data exposure.
A3	Injection: Entering malicious queries into the input fields such as SQL injection, NoSQL injection, etc.
A4	Insecure design: A broad category representing different weaknesses due to missing or ineffective control design.
A5	Security misconfiguration: Default account username and their passwords, improper error handling exposing system information, latest security patches not applied, etc.
A6	Vulnerable and outdated components: Increases the susceptibility to attacks.
A7	Identification and authentication failures: Takes place due to improper implementation of user identity and authentication function in a web application.
A8	Software and data integrity failures: Use of unauthenticated third-party tools, plug-ins, patches auto-updated lead to integrity failures.
A9	security logging and monitoring failures: Lack of security monitoring like integrity monitoring, log monitoring, root check, and process monitoring leads to the website failure.
A10	Server-side request forgery: Attacker sends manipulated requests from the back-end server of a vulnerable application abusing the server functionality in order to access or modify sensitive information.

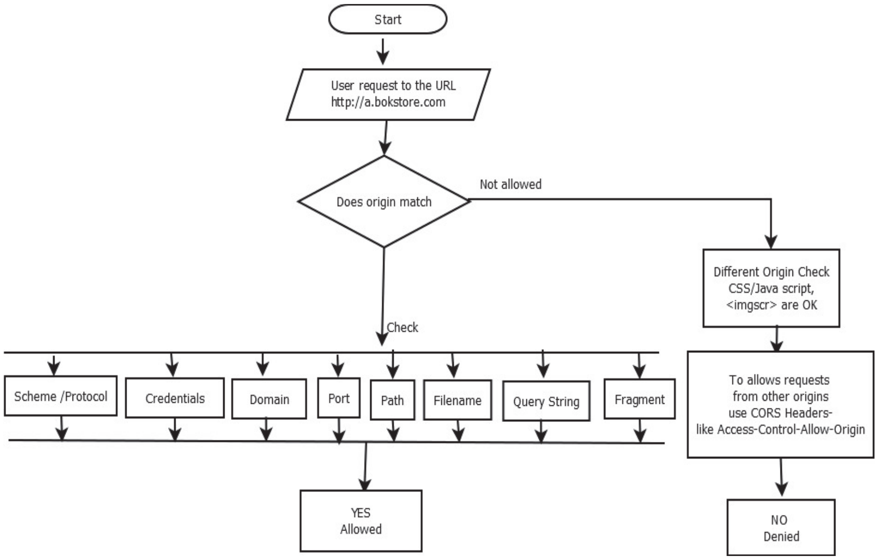

Figure 3 Relationship between SOP and CORS.

Web browser plug-ins like Flash Player, Adobe Flash, Silverlight, Windows Media Player, VLC, etc. are executable files located inside the browser with elevated privilege to communicate with OS directly violating SOP [6]. Extensions or add-ons are third party APIs with elevated privileges to bypass SOP and access other web applications' data, cookies, browsing history, bookmarks, etc. They have access to DOM of web pages [35].

Cross-origin resource sharing (CORS) was implemented in the web browsers due to the restrictions imposed by SOP to exchange data between different domains (Figure 3). It is a part of HTTP that lets servers specify any other hosts from which a browser should permit loading of content. A HTTP CORS request is triggered when a domain attempts to load resource from a different domain. It adds new HTTP headers to allow the local server to keep a list of allowed origins. Simple CORS involves browser requests asking data from another domain and do not need a pre-flight check. In the case of an HTTP request involving a method other than GET/POST or custom headers are set or the request body has a MIME type other than text/plain, the browser will initiate a pre-flight request to validate whether they have permission to perform an action. It contains two main headers access-control-allow-origin (ACAO) and access-control-allow-credentials (ACAC) and if misconfigured can pose a major threat to any web application.

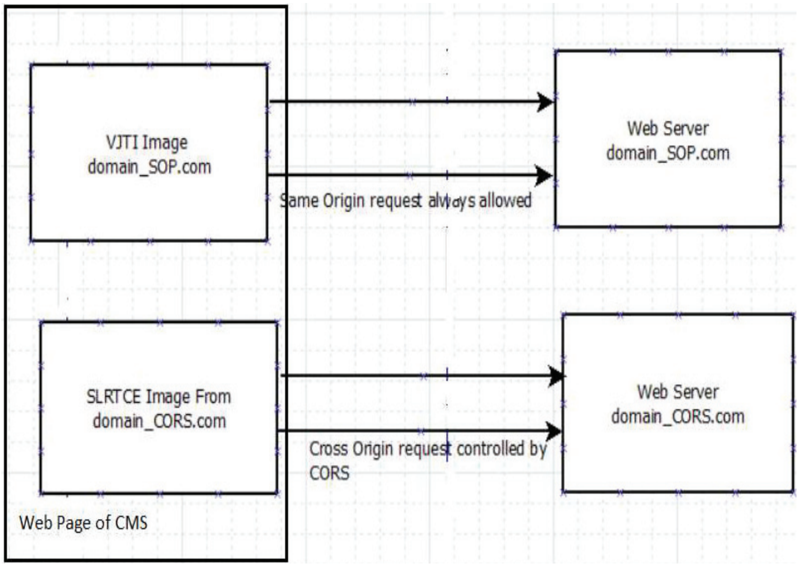

Figure 4 Example of SOP and CORS.

In Figure 4, the web page of the CMS is hosted on a domain_SOP.com domain. The first image points to the same web server following SOP whereas another image in web page points to another domain, namely domain_CORS.com if CORS is allowed on both websites. The security restrictions imposed by SOP are made flexible with the help of CORS in modern day websites.

3 Proposed Secure Hypertext Model (SHM)

In this section the proposed hypertext links, secure stereotypes for the particular attacks, same origin policy (SOP) and cross origin resources sharing (CORS) is discussed to secure the transactions of users in hypertext modelling.

3.1 Proposed Hypertext Links for Web Development

The following are the links in the proposed hypertext modelling.

1. Internal link: A link from a page on a site/domain to another page on the same site/domain. They are also commonly found on the main navigation, footer, and under images.

There are four types of internal links: (i) contextual links, (ii) navigational links, (iii) footer links, (iv) image links.

For example:

<a href="# Chapter1">Chapter 1</a> link can be referred as <a id="Chapter1">Introduction of Chapter 1</a>

i. Contextual links: These are internal links that share in the context of web content. These links carry contexts directly from sentences or can be added at the end [37].

ii. Navigational links: These links are interlinked on the navigational bar to make it easy for site visitors to move from page to page.

iii. Image links: Images can also be used as links like button images, charts, and infographics, etc. This type of link can be beneficial to readers if there is a page that would assist someone who wants additional information about that image [38].

For example:

<a href = "http://www.example .com" target = "_self"> <img src = "/images/logo.png" alt = "Example Website" border = "0"/> </a>

iv. Text links: Text links are hyperlinked words or phrases within your content. The text link is frequently structured in blue text and may include an underline.

For example:

<a href = "https://www.tutorialspoint.com" target = "_self">Example Website</a>

v. Footer links: Links in the footer section are footer links [35].

For example:

<footer><p>@Copyright 2022, All rights reserved. </p> </footer>

2. External links

These links are used to go to other pages of a same website.

I-links, T-links, X-Links of OOH methods are same as internal links, navigational links and external links of the proposed secure hypertext model.

3. Inbound links come from other websites or a different domain name.

4. Outbound links are used to link to other websites with a different domain name [38].

Intra-page links and inter-page links of WebML methods are same as inbound links and outbound links of the proposed secure hypertext model.

Table 2 Security hyperlink between client and server

Name of Link	Source	Destination	Attack	Stereotype
Contextual link	http://www.example.com	http://www.example.com	BA2, BA7, BA8, BA11	StCSRF, StfilterXSS
Navigational links	http://www.example.com/home.html	http: //www.example.com/about.html	BA9, BA11, BA12	StPwdStrength, StSOP
Image links	<img src="/images/logo.png" alt= "Example Website" border = "0"/>	<a href="http://www.example.com" target="_self"></a>	BA2, BA7, BA8, BA11	StCSRF, StfilterXSS
Text links	Example website	<a href="http://www.example.com" target="_self"></a>	BA2, BA7, BA8, BA11	StCSRF, StfilterXSS
Footer links	@Copyright 2022, All rights reserved.		BA9, BA11, BA12	StPwdStrength, StSOP
External links	http://www.example.com/page1.html	http: //www.example.com/page2.html	BA2, BA7, BA8, BA11	StCSRF, StfilterXSS
Inbound links	http://www.tutorial.com/home.html	http: //www.example.com/page1.html	BA2, BA7, BA8, BA11	StCSRF, StfilterXSS
Outbound links	http://www.example.com/page1.html	http: //www.tutorial.com/home.html	BA2, BA7, BA8, BA11	StCSRF, StfilterXSS

Various security hyper links between client and server, the link attack labels and stereotypes labels are shown in Table 2 and Figure 5.

The above table shows different types of links when visiting from source to destination, different types of attacks are possible, which are mentioned in the table to mitigate these attacks if developer has to incorporate these stereotypes in the coding phase to develop a secure web application.

3.2 Proposed Secure Stereotypes for Hypertext Modelling

This section proposes securing links in hypertext modelling: (1) Secure transfer link S(T), (2) S.<<filter XSS>>, (3) S.<<Chk Action>>, (4) S.<<Same origin policy >>, (5) S.<<filter URL>>, (6) S.<<HTTP Authentication >>, (7) S.<<False Password Change request>>, (8) S. <<Authentication Cookie>>.

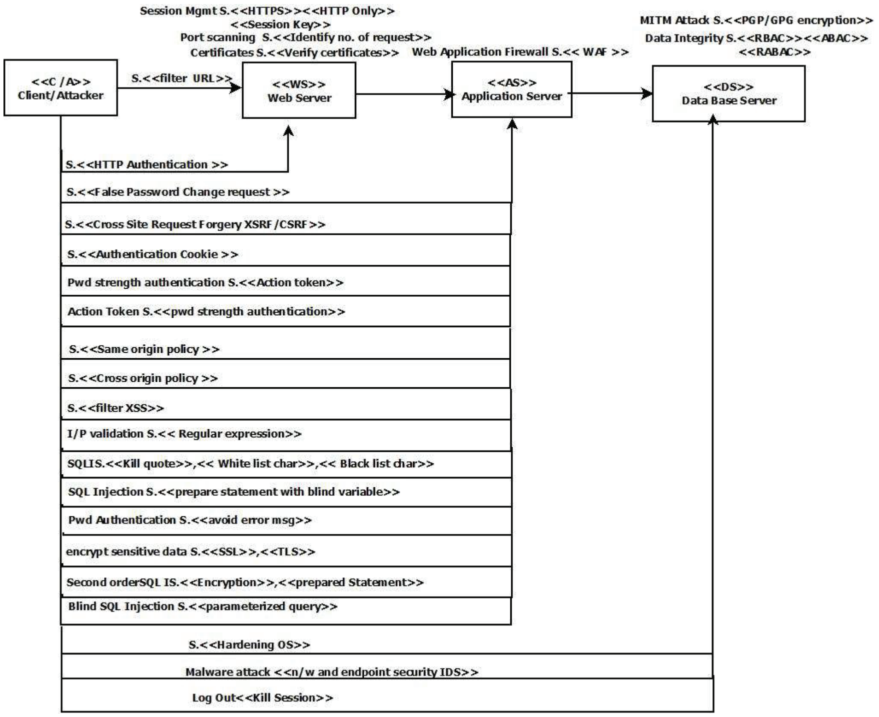
Session Mgmt S.<<HTTPS>><<HTTP Only>>
<<Session Key>>
Port scanning S.<<Identify no. of request>>
Certificates S.<<Verify certificates>>
Web Application Firewall S.<< WAF >>
MITM Attack S.<<PGP/GPG encryption>>
Data Integrity S.<<RBAC>><<ABAC>>
<<RABAC>>
<<C /A>>
Client/Attacker
S.<<filter URL>>
<<WS>>
Web Server
<<AS>>
Application Server
<<DS>>
Data Base Server
S.<<HTTP Authentication >>
S.<<False Password Change request >>
S.<<Cross Site Request Forgery XSRF /CSRF >>
S.<<Authentication Cookie >>
Pwd strength authentication S.<<Action token>>
Action Token S.<<pwd strength authentication>>
S.<<Same origin policy >>
S.<<Cross origin policy >>
S.<<filter XSS>>
I/P validation S.<< Regular expression>>
SQLIS.<<Kill quote>>,<< White list char>>,<< Black list char>>
SQL Injection S.<<prepare statement with blind variable>>
Pwd Authentication S.<<avoid error msg>>
encrypt sensitive data S.<<SSL>>,<<TLS>>
Second orderSQL IS.<<Encryption>>,<<prepared Statement>>
Blind SQL Injection S.<<parameterized query>>
S.<<Hardening OS>>
Malware attack <<n/w and endpoint security IDS>>
Log Out<<Kill Session>>

Figure 5 Secure hypertext model of a web application.

Figure 5 describes how secure communication takes place between client, web server, application server and database server using the stereotypes shown in Table 3 while coding developer will make use of this stereotypes to develop secure application.

As shown in Figure 1, a hypertext model of the CMS does not have any security measures over the navigation links while accessing different classes. Hence, we have proposed the security stereotypes as a security measure in hypertext structure model as shown in Figure 6.

The security stereotypes used in Figure 6 of proposed secure hypertext model of a college management system (CMS) are described in detail in Table 4.

All methods do not describe dependencies between the content model and hypertext model exactly. However, the proposed secure hypertext model shows the relationship between secure content model and secure hypertext model with the use of security stereotypes Secure transfer link S(T), Secure link to prevent XSS attack S.<<filter XSS>>, S.<<Chk Action>>,

Table 3 Proposed stereotypes in hypertext modelling to secure web applications

Stereotype Name	Description
StUrl	S.<<filter URL>>
StHttpAuth	S.<<HTTP Authentication >>
StFalsePwd	S.<<False Password Change request >>
StCSRF	S.<<Cross Site Request Forgery XSRF/CSRF>>
StAuthCookie	S.<<Authentication Cookie >>
StActionTtoken	Pwd strength authentication S.<<Action token>>
StPwdStrength	Action Token S.<<pwd strength authentication>>
StSOP	S.<<Same origin policy >>
StCORS	S.<<Cross origin policy >>
StfilterXSS	S.<<filter XSS>>
StRegularExp	I/P validation S.<< Regular expression>>
StSQLClear	SQLIS.<<Kill quote>>,<< White list char>>, <<Black list char>>
StPrepareStmt	SQL Injection S.<<prepare statement with blind variable>>
StPwdAuth	Pwd Authentication S.<<avoid error msg>>
StEncrypt	encrypt sensitive data S.<<SSL>>,<<TLS>>
StEncryptStmt	Second orderSQL IS.<<Encryption>>, <<prepared Statement>>
StParamQuery	Blind SQL Injection S.<<parameterized query>>
StOSHarden	S.<<Hardening OS>>
StSessionMgmt	Session Mgmt S.<<HTTPS>>,<<HTTP Only>>, <<Session Key>>
StRequest	Port scanning S.<<Identify no. of request>>
StVerifyCert	Certificates S.<<Verify certificates>>
StWAF	Web Application Firewall S.<< WAF >>
StDataInteg	Data Integrity S.<<RBAC>><<ABAC >> <<RABAC>>
StMITM	Man-In-The Middle Attack S.<<PGP/GPG encryption>>
StVirtualHost	Virtual Hosting <<get Request>>,<<host name>>, <<Server Name>><<SNI>>
StMalwareSec	Malware attack <<n/w and endpoint security IDS>>
StKillSession	Log Out<<Kill Session>>

S.<<Same origin policy>>, S.<<filter URL>>, S.<<HTTP Authentication >>, S.<<False Password Change request >>, S.<<Authentication Cookie >> and the like, as shown in Table 3.

3.3 Comparison of Web Methods

UML does not provide hypertext modelling for the analysis of the web information system and appropriate concepts for the specification of hyperlinks,

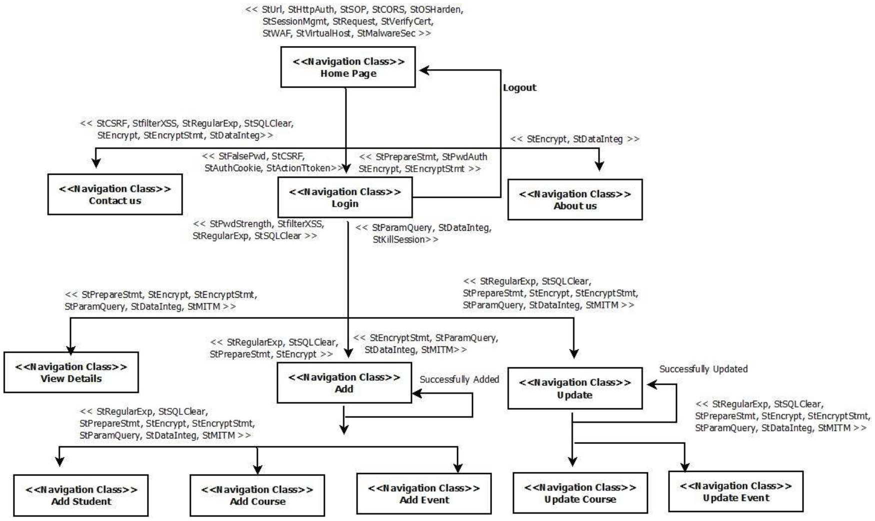
<< StUrl, StHttpAuth, StSOP, StCORS, StOSHarden, StSessionMgmt, StRequest, StVerifyCert, StWAF, StVirtualHost, StMalwareSec >>
<<Navigation Class>> Home Page
Logout
<< StCSRF, StfilterXSS, StRegularExp, StSQLClear, StEncrypt, StEncryptStmt, StDataInteg>>
<< StEncrypt, StDataInteg >>
<< StFalsePwd, StCSRF, StAuthCookie, StActionTtoken>>
<< StPrepareStmt, StPwdAuth StEncrypt, StEncryptStmt >>
<<Navigation Class>> Contact us
<<Navigation Class>> Login
<<Navigation Class>> About us
<< StPwdStrength, StfilterXSS, StRegularExp, StSQLClear >>
<< StParamQuery, StDataInteg, StKillSession>>
<< StPrepareStmt, StEncrypt, StEncryptStmt, StParamQuery, StDataInteg, StMITM >>
<< StRegularExp, StSQLClear, StPrepareStmt, StEncrypt, StEncryptStmt, StParamQuery, StDataInteg, StMITM >>
<< StRegularExp, StSQLClear, StPrepareStmt, StEncrypt >>
<< StEncryptStmt, StParamQuery, StDataInteg, StMITM>>
<<Navigation Class>> View Details
<<Navigation Class>> Add
Successfully Added
<<Navigation Class>> Update
Successfully Updated
<< StRegularExp, StSQLClear, StPrepareStmt, StEncrypt, StEncryptStmt, StParamQuery, StDataInteg, StMITM >>
<< StRegularExp, StSQLClear, StPrepareStmt, StEncrypt, StEncryptStmt, StParamQuery, StDataInteg, StMITM >>
<<Navigation Class>> Add Student
<<Navigation Class>> Add Course
<<Navigation Class>> Add Event
<<Navigation Class>> Update Course
<<Navigation Class>> Update Event

Figure 6 Proposed secure hypertext model of a college management system (CMS).

Table 4 Comparison between hypertext methods to develop a secure web application

Sr. No	Web Application Features	WebML	UWE	SHM
1	a. Design RIA server	FS	FS	FS
	b. Design the structure of rich user interface	PS	PS	FS
	c. Generate browser-oriented rich client	FS	FS	FS
	d. Generate plug-in oriented rich client	FS	X	FS
2	a. Access anytime	FS	FS	FS
	b. Access anywhere	X	X	FS
	c. Access from any media	FS	X	FS
3	a. Web mining	NA	NA	NA
	b. Intelligent agent	NA	NA	FS
	c. Web personalization	NA	NA	FS
4	Security features with stereotypes	NA	NA	FS

FS: fully supported; PS: partially supported; NA: not applicable; X: not supported.

authoring systems, and security from malicious links. UMLSec also does not provide security stereotypes for hypertext modelling of web applications. Hence we propose a secure hypertext model with stereotypes to provide security features like a secure transfer link S(T), a secure link to prevent XSS attack S.<<filter XSS>>, S.<<Chk Action>>, S.<<Same origin policy >>, S.<<filter URL>>, S.<<HTTP Authentication>>, S.<<False Password Change request >>, S.<<Authentication Cookie>>.

4 Conclusion

An extensive literature survey was performed on web engineering which deals with content modeling for class and state transition modelling, hypertext modeling for site navigation and presentation modelling for look and feel. The hypertext modelling generates hypertext structure model and access model. hypertext structure model deals with intra-page navigation, inter-page navigation, frame-based navigation and navigation of multiple windows. Access model presents the various access structure such as <<index>>, <<menu>>, <<guided tour>>, <<query>>, <<home>> and <<landmark>>.

The various web modelling methods like WebML, HDM, UWE, OOH and UMLSec for web development are described to show their hypertext concepts. To define the various stereotypes for various use case attacks like browser-based attacks, operating system attacks, database attacks, OWASP top 10 attacks are explored for the design of the proposed hypertext modelling. Same origin policy (SOP) and cross origin resource sharing (CORS) are also discussed.

This proposed secure hypertext model (SHM) can identify the attack vectors on hyperlinks in a more precise manner and provides defences through security stereotypes to build more secure applications.

References

[1] R. Cao, and X. Liu, *IFML-Based Application Modeling*. Elsevier, 2020.

[2] K. Wakil and D. N. A. Jawawi, “A new adaptive model for web engineering methods to develop modern web applications”, ICSIM2018, ACM, 2018.

[3] D. Ingle and B. B. Meshram, “Hybrid analysis and design model for building web information system”, *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 1694–0814, July 2012.

[4] K. Wakil and D. N. A. Jawawi, “Combining web engineering methods to cover lifecycle”, *Computer Modelling & New Technologies*, 2017.

[5] D. Granada, J. M. Vara, M. Brambilla, V. Bollati, E. Marcos, *Analysing the Cognitive Effectiveness of the Webml Visual Notation*. Springer, 2015.

[6] Q. Wang and Z. Qin, *KUBERA: A Security Model for Web Applications*. IEEE, 2010.

[7] K. Wakil and D. N. A. Jawawi, "Comparison between web engineering methods to develop multi web applications", *Journal of Software,* vol. 12, no. 10, October, 2017

[8] S. Ceri, P. Fraternali, A. Bongio, "Web modeling language (WebML): A modeling language for designing web sites", *Computer Networks*, vol. 33, pp. 137–157, 2000.

[9] R. Sharma, S. R. Kumar, *Strategies for Web Application Development Methodologies*. IEEE, 2016.

[10] T. Margari, C. Winkler, C. Kubczak, B. Steffen, "The Sws mediator With Webml/Webratio And Jabc/Jeti: A comparison", *International Conference on Enterprise Information Systems*, 2016.

[11] N. Choudhury, "World wide web and its journey from Web 1.0 to Web 4.0", *(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, 2014.

[12] Y. Deshpande, S. Murugesan, A. Ginige, S. Hansen, D. Schwabe, M. Gaedke, B. White, "Web engineering", *Journal of Web Engineering*, vol. 1, no. 1, pp. 003–017, 2002.

[13] R. Acerbis, A. Bongio, M. Brambilla, S. Butti, S. Ceri, Piero Fraternali, "Web applications design and development with WebML and WebRatio 5.0", *TOOLS EUROPE 2008, LNBIP 11*, pp. 392–411, Springer, 2008.

[14] M. Gedam and B. B. Meshram, "Proposed secure 3 use-case diagram", *International Journal of Systems and Software Security and Protection*, IGI Global, 2022.

[15] M. Brambilla, I. Celino, S. Ceri, D. Cerizza, E. Della Valle, F. Michele Facca, "A software engineering approach to design and development of semantic web service applications", *ISWC 2006, LNCS 4273*, pp. 172–186, Springer, 2006.

[16] M. Brambilla and F. M. Facca, *Building Semantic Web Portals with WebML*, L. Baresi, P. Fraternali, and G.-J. Houben (eds.). ICWE 2007, LNCS 4607, pp. 312–327, Springer, 2007.

[17] F. M. Facca, M. Brambilla, "Extending WebML towards Semantic Web", *WWW 2007, May 8–12, 2007, Banff, Alberta, Canada*.

[18] N. Moreno, P. Fraternali, A. Vallecillo, "WebML modelling in UML", *The Institution of Engineering and Technology*, 2007. doi:10.1049/iet-sen:20060067.

[19] R. Cao and X. Liu, "IFML-based web application modeling", *3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019)*, Elsevier.

[20] U. Sabir, F. Azam, S. Ul Haq, M. Waseem Anwar, W. Haider Butt, A. Amjad, “A model driven reverse engineering framework for generating high level UML models from Java source code”, *IEEE Access*, 2019.

[21] S. Ceri, F. Daniel, F. M. Facca, M. Matera, “Model-driven engineering of active context-awareness”, *World Wide Web*. Springer, 2007.

[22] A. Kraus, A. Knapp, N. Koch, “Model-Driven Generation of Web Applications in UWE”.

[23] Karl R.P.H. Leung, Lucas C.K. Hui, S.M. Yiu, Ricky W.M. Tang, “Modeling Web Navigation by Statechart”, IEEE, 2000.

[24] Cristina Cachero , Nora Koch, “Navigation Analysis and Navigation Design in OO-H and UWE”, Available at: https://ceur-ws.org/Vol-261/paper03.pdf.

[25] M. Brambilla, S. Comai, P. Fraternali, M. Matera, “Designing web applications with WebML and WebRatio”, Web Engineering: Modelling and Implementing Web Applications. 2008, ch. 9.

[26] G. Kappel, W. Schwinger, N. Koch, *Modeling Web Applications*, March 31, 2006.

[27] M. Brambilla, S. Ceri, P. Fraternali, “Process modeling in web applications”, ACM *Transactions on Software Engineering and Methodology*, vol. 15, no. 4, October, 2006.

[28] S. Ceri, M. Brambilla, P. Fraternali, “The history of WebML lessons learned from 10 years of model-driven development of web applications”, A. T. Borgida et al. (Eds.), *Mylopoulos Festschrift, LNCS 5600*, pp. 273–292, Springer, 2009.

[29] M. Zaremba, T. Vitvar, M. Moran, “Towards semantic interoperabilty in-depth comparison of two approaches to solving semantic web service challenge mediation tasks”. *ICEIS 2007 – Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume SAIC, Funchal, Madeira, Portugal, June 12–16, 2007.*

[30] N. Moreno, P. Fraternalli, A. Vallecillo, “A UML 2.0 Profile for WebML Modeling”, *ICWE’06 Workshops, July 10-14, 2006, Palo Alto, CA*.

[31] M. J. Escalona , G Aragón, “NDT. A Model-Driven Approach for Web Requirements,” *IEEE Transactions on Software Engineering*, vol. 34, no. 3, pp. 377–390, May–June 2008.

[32] A. Bongio, S. Ceri, P. Fraternali, A. Maurino, “Modeling data entry and operations in WebML”. in: G. Goos, J. Hartmanis, J. van Leeuwen, D. Suciu, G. Vossen, G. (Eds) *The World Wide Web and Databases. WebDB 2000. Lecture Notes in Computer Science, vol. 1997.* Springer, 2001.

[33] R. Vdovjak, F. Frasincar, G.-J. Houben, P. Barna, "Engineering semantic web information systems in Hera", *Journal of Web Engineering*, vol. 2, no. 1, pp. 3–26.

[34] N. Koch, H. Baumeister, L. M. Hennicker, *Extending UML to Model Navigation and Presentation in Web Applications*, 2000.

[35] D. F. Som, "EmPoWeb: Empowering web applications with browser extensions", *IEEE Symposium on Security and Privacy*, 2019.

[36] https://www.accuranker.com/learn-seo/beginner/guide-to-external-links-for-seo.

[37] https://yoast.com/internal-linking-for-seo-why-and-how/.

[38] https://friospops.com/blog/the-difference-between-inbound-outbound-and-internal-seo-links/.

[39] A. Poniszewska-Maranda, "UML representation of extended role-based access control model with the use of usage control concept", *Multidisciplinary Research and Practice for Information Systems. CD-ARES 2012. Lecture Notes in Computer Science*, vol. 7465. Springer, 2012.

[40] M. Mohsin and M. U. Khan, *UML-SR: A Novel Security Requirements Specification Language*. IEEE, 2019.

[41] MDN contributors, Cross-Origin Resource Sharing (CORS), (May 21, 2022), https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Biographies

Madhuri N. Gedam, a Research Scholar at Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga, Mumbai 400019, obtained her Bachelor's degree (Computer Engineering) from Nagpur University and M.Eng. (Computer Engineering) from Mumbai University. She is currently working as Assistant Professor in Information Technology Dept of SLRTCE, Mira Road, Thane. She has 12 years of teaching experience. She has authored around 30+ papers in national and international conferences and journals. She has received the “Best Technical Paper” award in a National Conference held at Nashik, India in 2018. Her area of specialization includes software engineering, database security, DevOps, cyber security and big data analytics.

Bandu B. Meshram, Professor and Former Head of Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga Mumbai 400019, obtained his Bachelor’s degree in (Computer Engineering), M.Eng. (Electronics Engineering) and Ph.D. in Computer Engineering and obtained a Bachelor’s degree (LLB) and LLM (Constitution Law). He is also a Computer Hacking Forensic Investigator (CHFI) EC-Council USA Certified. He has graduated 10 Ph.D. students and 7 students are now in research work. He has guided over 190^+ M.Tech. (Computer Engineering) students, authored over 400+ publications, filed 10 patents and authored 7 books, and is a member of the editor team for five journals. His research interests are databases, software engineering, cyber and cloud security and digital forensics. He is a proponent of concise scientific lawful thinking and truth communication. He believes that teaching can’t be just a profession, but has to be one’s ‘Dhamma’.