Rancang Bangun Aplikasi Surat Izin Sekolah Berbasis Web Menggunakan Metode WDLC

Hermawan Setiawan[1)], Ismail Sofyan Tsany[2)]
1) Politeknik Siber dan Sandi Negara, hermawan.setiawan@poltekssn.ac.id
2) Badan Siber dan Sandi Negara, ismail.sofyan@bssn.go.id

Abstrak

Saat ini dengan populernya aplikasi berbasis web membawa dampak negatif berupa maraknya serangan dan kerentanan aplikasi web. Pembangunan aplikasi dengan pendekatan yang tepat dan identifikasi ancaman dapat meningkatkan keamanan aplikasi yang dibuat. Pada penelitian ini akan dibangun sistem informasi yaitu Surat Izin Online berbasis web yang dibangun menggunakan metode Web Development Lifecycle (WDLC) dengan menambahkan threat modeling dengan metode Attack Tree pada tahap analysis dan mitigasinya pada tahap development. Aplikasi kemudian diuji dengan menggunakan OWASP ZAP untuk mengetahui kerentanan aplikasi. Hasil dari penelitian ini adalah dengan menerapkan metode WDLC dan threat modeling aplikasi yang dibangun bisa mengatasi ancaman yang sebelumnya sudah dijabarkan.

Kata kunci: Aplikasi web, WDLC, Attack Tree, Pemodelan ancaman, OWASP ZAP

Abstract

Currently the popularity of web-based applications has a negative impact in the form of the rise of attacks and vulnerabilities on web applications. Application development with the right approach and identification of threats can improve the security of applications made. In this research, an information system that will be built is a web-based online permit that is built using the Web Development Lifecycle (WDLC) method by adding threat modeling to the Attack Tree method at the analysis stage and mitigation at the development stage. The application is then tested using OWASP ZAP to find out the application vulnerability. The results of this study are by applying the WDLC method and the threat modeling applications that are built can overcome the threats that have been previously described.

Key word: Web application, WDLC, Attack Tree, Threat modelling, OWASP ZAP

1. PENDAHULUAN

Dengan meningkatnya penggunaan media digital, pengunaan web pun mengalami peningkatan. Platform web memang menjadi favorit karena dapat mencakup semua informasi dan fitur yang ingin disampaikan. Pengguna pun dapat dengan mudah mengakses halaman web, sehingga informasi bisa tersampaikan dengan baik. Oleh karena itu penggunaan platform berbasis web masih sangat digandrungi.

Salah satu sektor yang menggunakan platform web adalah sektor pendidikan. Kegiatan administrasi seperti kehadiran saat ini sudah banyak beralih menjadi digital [1] [2]. Akan tetapi perizinan sekolah masih menggunakan surat biasa. Hal ini tentunya dapat dimaksimalkan dengan mengalihkannya ke bentuk digital.

Dari pernyataan di atas diperlukan aplikasi yang dapat digunakan sebagai sarana komunikasi perizinan resmi yang dibangun dengan standar keamanan yang baik. Penggunaan siklus pengembangan perangkat lunak yang aman bisa menjadi salah satu langkah untuk mengurangi kerawanan yang ditimbulkan ketika membangun sebuah sistem informasi (SI), salah satunya adalah menggunakan metode *Web Development Lifecycle* (WDLC). Metode ini salah satu standar dalam membangun aplikasi berbasis web.

Penelitian ini mencoba untuk menjawab pengaruh penggunaan kerangka pembangunan aplikasi web menggunakan metode WDLC. Selain itu juga akan diterapkan *threat modeling Attack Tree* pada tahap *analysis* dan dilakukan mitigasinya pada tahap *development*. Kemudian pada tahap *testing* akan dilakukan pemindaian keamanan menggunakan OWASP *Zed Attack Proxy* (ZAP).

2. LANDASAN TEORI

2.1 *Web-based Issues*

Web saat ini menjadi bagian dari kehidupan manusia seiring dengan banyak informasi dan kegiatan yang mulai difokuskan pada web. Akan tetapi hal ini menjadikan web menjadi sangat rawan akan serangan-serangan yang dampaknya bisa berupa kerugian ekonomi sampai pada penyalahgunaan informasi pribadi [3]. Hal ini diperburuk dengan upaya pengamanan aplikasi web terhitung sulit karena banyak dan beragamnya komponen dan teknologi yang digunakan [3].

Secara tradisional, keamanan web berkutat dengan dua jenis serangan yaitu serangan terhadap web dan serangan terhadap jaringan [3]. Jika serangan web berfokus pada injeksi konten berbahaya melalui *request* HTTP(S) yang diterima oleh salah satu server maka serangan jaringan akan berfokus pada hal yang

lebih luas yaitu komunikasi *traffic* di antara dua *network endpoints.*

Dampak yang dihasilkan oleh serangan web tidak main-main. Jika yang diserang adalah akun perbankan seperti PayPal, penyerang bisa saja membobol habis tabungan yang dimiliki nasabah dari bank tersebut. Contoh lainnya adalah serangan yang ditujukan pada SI rumah sakit, penyerang bisa saja membocorkan data riwayat penyakit pasien beserta informasi lain kemudian menjualnya. Hal-hal seperti ini tidak bisa dianggap remeh karena seperti kita ketahui bersama bahwa sebagian besar kegiatan manusia sudah beralih ke sistem digital dan platform web merupakan salah satu platform yang banyak digunakan.

2.2 *Web Development Lifecycle*

Pada tahun 1989, Rothi et.al [4] memperkenalkan metode dalam membangun suatu perangkat lunak. *Software Development Lifecycle* (SDLC) tradisional yang dibawakan ditekankan untuk membangun perangkat lunak menggunakan pendekatan yang sesuai dan berilmu dalam melakukan analisis dan desain perangkat lunak. Pada tahun 2013 kemudian hadir SDLC yang disesuaikan dengan platform web, WDLC, yang dikenalkan oleh Kamatchi et al. [5].

Metode WDLC yang dikenalkan Kamatchi et.al terdiri dari lima tahap: *planning, analysis, design and development, testing, dan implementation and maintenance* [5]. Pada tahap *planning* dilakukan proses identifikasi tujuan dan web, menentukan siapa yang akan menggunakan web, teknologi yang digunakan web, menentukan pemilik dan perawat dari web, dan menentukan apa dan dimana informasi akan diletakkan pada web. Pada tahap kedua dilakukan identifikasi fungsi dan fitur web, menentukan proses yang diperlukan untuk menjalankan web, dan analisis. Pada tahap ketiga, *design and development*, mulai dilakukan penggambaran diagram yang mewakili alur logika dan *physical artifact* yang dibangun selama proses *development.* Desain ini kemudian didokumentasikan dan web mulai dibangun. Tahap keempat yaitu *testing* adalah tahap dimana web didemonstrasikan untuk menguji fitur dan tampilan sudah sesuai dengan ekspektasi. Terakhir, *implementation and maintenance*, berisikan kegiatan berupa instalasi web pada komputer atau server dan perawatan web seperti *update* dan lainnya.

2.3 *Threat Modeling: Attack Tree*

Perkembangan teknologi saat ini tentu banyak mengubah pola hidup manusia, bahkan dari segi kejahatan. Kejahatan siber kini menjadi perhatian dunia luas. Banyaknya jenis dan jumlah serangan siber menjadikan sulit untuk melakukan evaluasi secara manual.

Threat modeling menyediakan solusi bagi pengembangan aplikasi yang aman dan evaluasi sistem keamanan [6]. Tujuannya adalah untuk lebih proaktif dan mempersulit penyerang untuk melancarkan serangannya. Metodenya pun saat ini sudah banyak seperti *STRIDE, Attack Tree, VAST, DREAD, Microsoft SDL*, dan lain-lain.

Pada penelitian ini metode *threat modeling* yang akan digunakan adalah *Attack Tree*. *Attack Tree* digunakan untuk mefokuskan penelitan masalah web dan jalur-jalur yang menuju pada masalah tersebut sehingga lebih mudah dipahami dan fokus pada serangan [7]. Pada *Attack Tree* ancaman dijabarkan sesuai dengan tujuan utamanya atau disebut dengan *root*, kemudian digambarkan *intermediate goal* untuk mencapai tujuan utama dan terakhir dijabarkan *leaf/subgoal* yaitu implementasi untuk mencapai *intermediate.*

2.4 OWASP ZAP

OWASP ZAP merupakan *security tools* yang berfungsi untuk melakukan pemindaian keamanan web [8]. *Tools* ini dikembangkan dalam naungan OWASP Foundation dan menjadi salah satu proyek yang paling aktif. Ketika digunakan sebagai *proxy server*, pengguna dapat memanipulasi semua *traffic* yang melewatinya, termasuk HTTP(S). *Tools* ini juga memungkinkan untuk berjalan secara *daemon* yang dikontrol melalui REST API.

OWASP ZAP memiliki banyak fitur yang bermanfaat untuk melakukan penilaian risiko pada platform web. Fitur yang dimiliki OWASP ZAP antara lain: pemindaian otomatis, pemindaian pasif, *forced browsing, fuzzer*, mendukung *websocket, scripting language*, mendukung *plug-n-hack*, menyela *proxy server*, dan *traditonal and AJAX web crawler*.

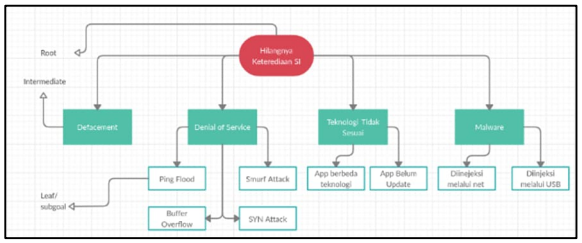
3. METODE PENELITIAN

Pada penelitian ini akan dibangun sebuah SI Surat Izin *Online* berbasis web yang dikembangkan dengan metodologi WDLC. Pada metodologi WDLC terdapat lima tahap. Kelimanya yaitu *planning, analysis, design and development, testing,* dan *implementation and maintenance* [5]. Pada tahap *planning* ditentukan tujuan dari SI yakni sebagai sarana untuk perizinan siswa sekolah secara resmi, yang kemudian bisa digunakan oleh siswa, staf administrasi, dan wali kelas dari sekolah tersebut. Selain itu pada tahap *planning* juga ditentukan bahwa SI ini menggunakan *database* untuk menyimpan data akun dan surat masuk, dimiliki sekolah dan dirawat oleh bagian administrasi sekolah. Pada tahap *analysis* ditentukan bahwa SI ini memiliki beberapa fitur yaitu kirim terima surat dan CRUD akun dan juga memiliki dua *role* yaitu siswa sebagai pengguna dan wali kelas atau staf administrasi sebagai pengelola. Sebagai tambahan pada tahap *design and development* dan *testing* dilakukan *threat modeling* menggunakan metode *Attack Tree*.

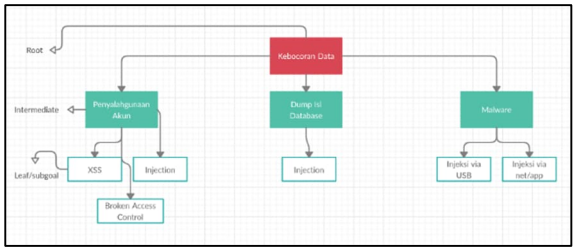
3.1 *Threat Modeling* Sistem Informasi

Pada penelitian ini *threat modeling* yang digunakan adalah *Attack Tree* karena merupakan

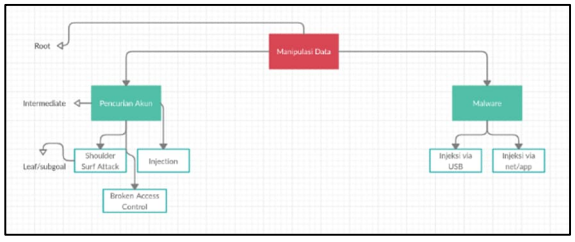
metode yang sederhana dan banyak diterapkan di metode *threat model* lainnya sebagai dasar. Dari bermacam-macam kerawanan dan ancaman dihasilkan tiga buah tujuan utama yang menjadi acuan seorang penyerang yaitu manipulasi data, kebocoran data, dan hilangnya ketersediaan SI. Tiga tujuan ini didapatkan dari konsep *CIA triad* pada keamanan informasi yang juga dipakai untuk identifikasi masalah yang ada. Berikut merupakan *Attack Tree* dari SI yang dibangun.



Gambar 1. *Attack Tree* 1



Gambar 2. *Attack Tree 2*

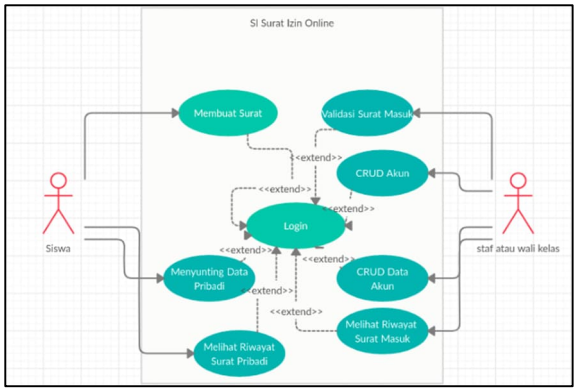


Gambar 3. *Attack Tree 3*

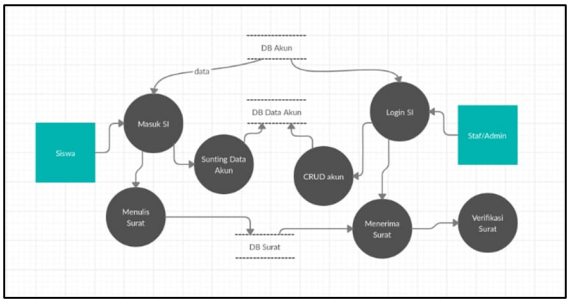
Pada diagram *Attack Tree* di atas terdapat tiga jenis kotak, yaitu kotak merah yang artinya *root* atau tujuan utama, kotak tosca yang artinya *intermediate* atau *goal* yang bisa membawa ke *goal* utama, dan kotak putih yang artinya *leaf/subgoal* yang merupakan implementasi yang bisa dilakukan penyerang untuk mencapai *goal* pada level *intermediate*. Ancaman yang perlu diperhatikan oleh pengembang aplikasi adalah ancaman pada level *leaf/subgoal* karena merupakan implementasi langsung dari penyerang. Untuk penjabaran lebih jelas mengenai ancamannya perhatikan Tabel 1.

3.2 *Design, Development, and Mitigation*

Pada tahap ketiga, *design and development*, mulai dirancang alur kerja dari SI dan pengkodingan. Untuk membantu memahami alur kerja dari SI yang dibangun dibuat dua buah diagram, yaitu *usecase* diagram dan *data-flow diagram* (DFD) sebagai berikut:



Gambar 4*: Usecase Diagram*



Gambar 5. DFD

Tabel 1. Identifikasi Ancaman Aplikasi

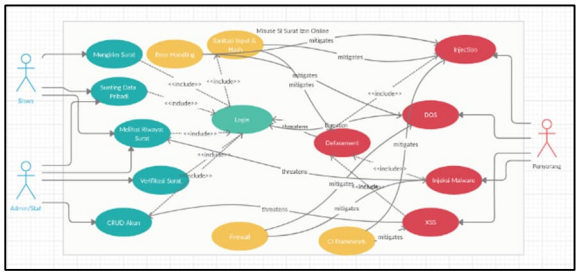
Root	*Intermediate*	*Leaf/Subgoal*
Hilangnya Ketersediaan SI	*Defacement*	-
	Denial of Service	*Ping Flood, Smurf Attack, Buffer Overflow, SYN Attack*
	Teknologi Tidak Sesuai	Perbedaan Teknologi Aplikasi, Aplikasi Belum Diperbaharui
	Malware	Injeksi via USB, Injeksi via net/app
Kebocoran Data	Penyalahgunaan Akun	*XSS, Injection, Broken Access Control*
	Dump Isi *Database*	*Injection*
	Malware	Injeksi via USB, Injeksi via net/app
Manipulasi Data	Pencurian Akun	*Shoulder Surf Attack, Injection, Broken Access Control*
	Malware	Injeksi via USB, Injeksi via net/app

Pada tahap ini juga akan dilakukan mitigasi dari ancaman yang sudah dijabarkan pada Tabel 1. Adapun pada penelitian ini, mitigasi yang dilakukan berada pada tingkat kode, jadi beberapa ancaman yang disebutkan pada Tabel 1 tidak akan diperhitungkan skenarionya. Mitigasi yang dilakukan akan dijabarkan melalui Tabel 2 di bawah ini.

Tabel 2. Ancaman Level Kode

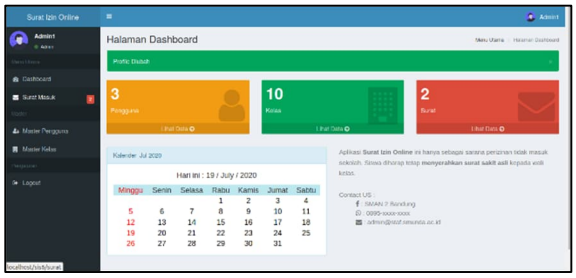
No	Ancaman	Mitigasi
1.	*Defacement*	*Framework, Error Handling*
2.	*Buffer Overflow*	*Error Handling, sanitasi input*
3.	*Injection*	*Error Handling, sanitasi input*
4.	*XSS*	*Framework*
5.	*Broken Access Control*	*Role based auth*

Ancaman pada Tabel 2 merupakan ancaman yang dapat diatasi pada level kode. Adapun ancaman lain seperti DOS atau *Malware* dapat diatasi dengan pendekatan lain yang lebih efektif seperti *firewall* atau *antivirus*. Maka dari ancaman dan mitigasi pada Tabel 2 dapat digambarkan pada diagram *misuse* di bawah.

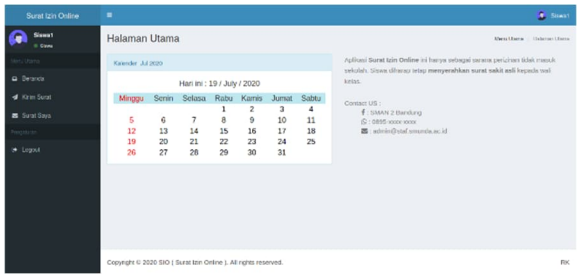


Gambar 6. *Misuse Diagram*

Dari semua diagram di atas, SI kemudian mulai ke tahap pengkodingan. Untuk membangun SI berbasis web ini, peneliti menggunakan bahasa pemograman PHP dan memanfaatkan *framework* Code Ignitor. Berikut merupakan tampilan SI saat dijalankan dengan fitur-fitur di dalamnya.



Gambar 7. Tampilan *Dashboard* Staf



Gambar 7. Tampilan *Dashboard* Siswa

3.3 *Testing* dan Pemindaian Kerawanan

Setelah dilakukan identifikasi ancaman dan dilakukan mitigasi, SI akan dipindai kemanannya. Proses pemindaian akan menggunakan *automated tools* OWASP ZAP dengan server localhost dan tipe *spider*, *Firefox Ajax Spider*. Dari hasil pemindaian otomatis ditemukan 10 *alerts* dengan rincian dua *medium flags*, empat *low flags*, dan empat *information flags.*

Gambar 9. Tampilan Halaman Login

3.4 *Implementation and Maintenance*

Pada tahap terakhir, SI diimplementasikan pada server localhost. Pada tahap ini juga SI yang dihasilkan melalui serangkaian tahap WDLC ini siap digunakan.

4. HASIL DAN PEMBAHASAN

Dari pemindaian yang dilakukan dengan menggunakan *automated tools* OWASP ZAP, didapatkan 10 *alerts* dengan rincian dua *medium alerts*, empat *low alerts*, dan empat *informational alerts.* Akan tetapi dari 10 *alerts* yang didapat beberapa *alerts* nyatanya berasal dari kerentanan pada server *localhost* yang dipakai, bukan dari SI, maka tidak semua *alerts* yang didapatkan melalui pemindaian OWASP ZAP dapat diterima karena pada penerapannya nanti SI Surat Izin *Online* ini akan menggunakan *hosting* berbayar. Adapun kerawanan yang berasal dari pembangunan SI dapat dilihat dari Matrik Risiko [9] Tabel 3 berikut ini.

Tabel 3. Hasil Pemindaian Kerawanan OWASP ZAP

Alerts	*Risk*	Jumlah	Keterangan
Directory Browsing	*Medium*	18	Daftar direktori dimungkinkan untuk dilihat
X-Frame-Options Header Not Set	*Medium*	3	*X-Frame-Options Header* yang melindungi dari serangan *ClickJacking* tidak termasuk dalam respon HTTP
Absence of Anti-CSRF Tokens	*Low*	3	Tidak terdapat token Anti-CSRF pada *form submission*
Cross-Domain JavaScript Source File Inclusion	*Low*	6	Halaman terdiri dari satu atau lebih *script* dari *third-party domain*
Content-Type Header Missing	*Information al-medium*	2	*Content-Type header* hilang atau kosong
Information Disclosure - Suspicious Comments	*Information al-low*	1	*Respons* yang ditampilkan mengandung pesan informatif
Loosely Scoped Cookie	*Information al-low*	2	*Cookie* dapat diambil oleh domain atau *path*

Dari hasil pemindaian didapatkan tujuh *alerts* secara total dengan rincian dua *alerts medium*, dua *alerts low*, dan tiga *alerts informational*. Dari tingkat level bahaya dan jumlah kerentanan yang ditemukan, kerentanan *Directory Browsing* merupakan kerentanan yang paling perlu diperhatikan. Ini menandakan bahwa walaupun sudah dilakukan *threat modeling* dan mitigasi ancamannya, aplikasi masih menyimpan kerawanan lain yang tidak dijabarkan melalui *threat modeling*. Walaupun begitu ancaman yang sudah dijabarkan melalui *threat modeling* sebelumnya sudah bisa dimitigasi sehingga tidak menimbulkan ancaman lagi.

5. KESIMPULAN

WDLC merupakan metode untuk membangun suatu aplikasi berbasis web. Pada dasarnya tahap-tahap pada WDLC sama dengan SDLC namun pada setiap tahapnya lebih disesuaikan dengan teknologi dan keperluan aplikasi web. Pada penelitian ini aplikasi web yang akan dibangun berupa SI Surat Izin *Online* yang bisa digunakan oleh siswa dan wali kelas/staf administrasi. Pada penelitian ini, tahapan *design* dan *testing* pada WDLC ditambahkan proses *threat modeling* dan mitigasinya. Setelah dilakukan mitigasi, hasil pemindaian oleh OWASP ZAP membuktikan bahwa ancaman yang dijabarkan *threat modeling Attack Tree* dapat diatasi. Sehingga dapat disimpulkan bahwa penggunaan *threat modeling* pada tahapan *design and development* dan *testing* dapat menghasilkan aplikasi web yang lebih aman.

Adapun penelitian ini dapat dilanjutkan dengan menerapkan metode *phased development* SDLC yang dikombinasikan dengan WDLC. Ini didasarkan pada hasil pemindaian pada penelitian ini masih ditemukan kerawanan yang masih bisa dimitigasi. Dengan menerapkan *phased development* hal ini dapat diatasi. Selain itu bisa dicobakan *threat model* lain seperti STRIDE atau DREAD yang mungkin bisa menjabarkan ancaman lebih detail.

REFERENSI

[1] A. H. K. Yuen, N. Law, and K. C. Wong, "ICT implementation and school leadership: Case studies of ICT integration in teaching and learning," *Journal of Educational Administration*, vol. 41, no. 2, pp. 158–170, Apr. 2003, doi: 10.1108/09578230310464666. Available: https://www.emerald.com/insight/content/doi/10.1108/09578230310464666/full/html.

[2] Hendra and Y. Arifin, "Web-based Usability Measurement for Student Grading Information System," *Procedia Computer Science*, vol. 135, pp. 238–247, 2018, doi: 10.1016/j.procs.2018.08.171. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050918314601.

[3] M. Bugliesi, S. Calzavara, and R. Focardi, "Formal methods for web security," *Journal of Logical and Algebraic Methods in Programming*, vol. 87, pp. 110–126, Feb. 2017, doi: 10.1016/j.jlamp.2016.08.006. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352220816301055.

[4] J. Rothi and D. (Chi-C. Yen, "System Analysis and Design in End User Developed Applications," *Journal of Information Systems Education*, vol. 2, no. 1, pp. 11–17, Dec. 1989, Available: https://aisel.aisnet.org/jise/vol2/iss1/2

[5] R.Kamatchi, J. Iyer, and S. Singh, "Software Engineering:Web Development Life Cycle," *International Journal of Engineering Research & Technology*, vol. 2, no. 3, Mar. 2013, doi: 10.17577/IJERTV2IS3438. Available: https://www.ijert.org/research/software-engineeringweb-development-life-cycle-IJERTV2IS3438.pdf, https://www.ijert.org/software-engineeringweb-development-life-cycle.

[6] W. Xiong and R. Lagerström, "Threat modeling – A systematic literature review," *Computers & Security*, vol. 84, pp. 53–69, Jul. 2019, doi: 10.1016/j.cose.2019.03.010. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404818307478.

[7] H. Mantel and C. W. Probst, "On the Meaning and Purpose of Attack Trees," in *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, Hoboken, NJ, USA: IEEE, Jun. 2019, pp. 184–18415. doi: 10.1109/CSF.2019.00020. Available: https://ieeexplore.ieee.org/document/8823696/.

[8] A. Mohammed, J. Alkhathami, H. Alsuwat, and E. Alsuwat, "Security of Web Applications: Threats, Vulnerabilities, and Protection Methods," *International Journal of Computer Science and Network Security*, vol. 21, no. 8, pp. 167–176, Aug. 2021, doi: 10.22937/IJCSNS.2021.21.8.22. Available: https://doi.org/10.22937/IJCSNS.2021.21.8.22.

[9] The MITRE Corporation, "SAMPLE - Safety Management System Risk Matrix." Available: https://www.mitrecaasd.org/SMS/doc/Sample_Risk_Matrix.pdf. [Accessed: Aug. 23, 2024]