AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ
AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

Fərid Fazil oğlu Qarayev
Onur Zahir oğlu Novruzlu

Əlyazması hüququnda

“VEB TƏTBİQLƏRİN TƏHLÜKƏSİZLİK DİZAYN NÜMUNƏLƏRİNİN QİYMƏTLƏNDİRİLMƏSİ” mövzusunda

MAGİSTRLİK DİSSERTASİYASI

60509 - “Kompüter elmləri”

“Dövlət informasiya sistemlərinin təhlükəsizliyi”

Elmi rəhbər: **H.R.Paşayev**

BAKI-2024

MÜNDƏRİCAT

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

YÜKSƏK TƏHSİL İNSTİTUTU

MAGİSTRANTIN ANDI

"Veb tətbiqlərin təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi" mövzusunda təqdim etdiyimiz magistrlik dissertasiyasını elmi əxlaq normalarına və istinad qaydalarına tam riayət etməklə və istifadə etdiyim bütün mənbələri ədəbiyyat siyahısında əks etdirməklə yazdığıma and içirik və magistrlik dissertasiyasının AzTU Kitabxana İnformasiya Mərkəzində saxlanılması, həmin mərkəz tərəfindən AzTU Rəqəmsal Repozitoriyasına daxil edilərək repozitoriyanın veb saytında yerləşdirilməsinə icazə veririk.

Fərid Qarayev _____

Onur Novruzlu _____

Tarix:

İXTİSARLAR VƏ İŞARƏLƏR

API Tətbiq Proqramlaşdırma İnterfeysləri (Application Programming Interface)

AI Süni İntellekt (Artificial Intelligence)

ML Maşın Öyrənməsi (Machine Learning)

HTTP Hipermətn ötürmə protokolu (Hypertext Transfer Protocol)

HTTPS Hipermətn ötürmə protokolu təhlükəsizdir (Hypertext Transfer Protocol Secure)

DAST Dinamik Tətbiq Təhlükəsizlik Testi (Dynamic Application Security Testing)

SAST Statik Tətbiq Təhlükəsizliyi Testi (Static Application Security Test)

RASP İşləmə Zamanı Tətbiqin Özünü Mühaafizəsi (Runtime Application Self-Protection)

XSS Saytlararası Skriptlər (Cross-Site Scripting)

CSRF Saytlararası Sorğu Saxtakarlığı (Cross-Site Request Forgery)

CWE Ümumi Zəifliklərin Sadalanması (Common Weakness Enumeration)

CVSS Ümumi Zəiflik Qiymətləndirmə Sistemi (Common Vulnerability Scoring System)

CSP Məzmun Təhlükəsizliyi Siyasəti (Content Security Policy)

RBAC Rol Əsaslı Giriş Nəzarəti (Role-Based Cccess Control)

IDOR Təhlükəli Birbaşa Obyekt İstinadları (Insecure Direct Object References)

PoLP Ən Az İmtiyaz Prinsipi (Principles of Least Privilege)

CDN Məzmun Çatdırma Şəbəkələri (Content Delivery Network)

CA Sertifikat Təşkilatı (Certificate Authority)

XXE XML Xarici Təşkilat hücumu (XML External Entity attack)

SSRF Server Tərəfində Sorğu Saxtakarlığı (Server-Side Request Forgery)

BSIMM Yetkinlik Modelində Təhlükəsizlik Binası(Building Security In Maturity Model)

PCI DSS Ödəniş Kartı Sənayesi Məlumat Təhlükəsizliyi Standartı(Payment Card Industry Data Security Standard)

GİRİŞ

Mövzunun aktuallığı. Veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi mövzusu bir neçə səbəbə görə çox aktualdır. Kibertəhlükələrin artan yayılması və təkmilləşməsi ilə veb proqramları təcavüzkarlar üçün əsas hədəflərdir. Dizayn nümunələri ümumi təhlükəsizlik zəifliklərinin aradan qaldırılması üçün standartlaşdırılmış yanaşmalar təqdim etməklə bu təhlükələrin azaldılmasında mühüm rol oynayır. Veb proqramları tez-tez şəxsi məlumatlar, maliyyə detalları və özəl biznes məlumatları kimi həssas istifadəçi məlumatlarını idarə edir. Təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi bu məlumatların icazəsiz giriş, açıqlama və saxtakarlıqdan adekvat şəkildə qorunmasını təmin edir. GDPR, HIPAA və PCI DSS kimi tənzimləyici tələblər istifadəçi məlumatlarını qorumaq üçün xüsusi təhlükəsizlik tədbirlərini tələb edir. Müvafiq təhlükəsizlik dizayn nümunələrinin tətbiqi təşkilatlara bu qaydalara əməl etməyə və hüquqi təsirlərdən qaçmağa kömək edir. Təhlükəsizlik pozuntuları təşkilatın reputasiyasına ciddi ziyan vura və müştərilərin etibarını sarsıda bilər. Effektiv təhlükəsizlik dizayn nümunələrini daxil etməklə, müəssisələr müştəri məlumatlarını qorumaq və təhlükəsiz onlayn mühiti saxlamaq öhdəliyini nümayiş etdirə bilər. Təhlükəsizlik pozuntuları bərpa xərcləri, hüquqi ödənişlər, tənzimləyici cərimələr və gəlir itkisi daxil olmaqla əhəmiyyətli maliyyə xərclərinə səbəb ola bilər. Güclü təhlükəsizlik dizayn nümunələrinin proaktiv şəkildə qiymətləndirilməsi və həyata keçirilməsi təhlükəsizlik insidentləri riskini və onlarla əlaqəli xərcləri minimuma endirməyə kömək edə bilər. Kibertəhlükələr daim inkişaf edir və təşkilatlardan ayıq-sayıq olmağı və təhlükəsizlik tədbirlərini buna uyğun uyğunlaşdırmağı tələb edir. Təhlükəsizlik dizayn nümunələrinin müntəzəm olaraq qiymətləndirilməsi təşkilatlara zəif tərəfləri müəyyən etməyə, təkmilləşdirmələri həyata keçirməyə və yaranan təhlükələrdən qabaqda qalmağa imkan verir. Ümumilikdə, veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi onlayn sistemlərin məxfiliyini, bütövlüyünü və əlçatanlığını təmin etmək, həssas məlumatları qorumaq, qaydalara riayət etmək, işgüzar nüfuzu qorumaq və təhlükəsizlik insidentləri ilə bağlı maliyyə risklərini azaltmaq üçün vacibdir.

Tədqiqatın məqsəd və vəzifələri. Tədqiqatın məqsədi veb tətbiqlərin təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsinin tədqiqidir.

Bu məqsədə çatmaq üçün aşağıdakı vəzifələri yerinə yetirmək lazımdır:

- Veb tətbiqini araşdırmaq və komponentlərini müəyyən etmək;
- Veb tətbiqinin təhlükəsizliyi dizayn nümunələrini müəyyən etmək;

Tədqiqatın obyekti və predmeti. Tədqiqatın obyekti veb tətbiqləridir, predmeti isə bu tətbiqlərin təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsidir.

Tədqiqatın nəzəri və metodoloji əsası. Tədqiqatın nəzəri əsasını müasir dövrdə veb tətbiqinə dair xarici alimlərin əsərlərində işlənmiş müasir veb tətbiqinin nəzəriyyəsi təşkil etmişdir. İnstrumental-metodoloji aparata öz struktur və funksional tədqiqat metodları ilə tarixi-məntiqi, sistemli və situasiya yanaşmaları, təhlil, habelə dialektik prinsiplər və ziddiyyətlər, sadədən mürəkkəbə yüksəliş, induksiya və situasiya metodlarından istifadə edilmişdir.

Tədqiqatın elmi yeniliyi. Təklif olunan tədqiqat həm nəzəri, həm də praktiki aspektləri nəzərə alaraq, əsas veb tətbiqi framevorklərini qabaqcıl təhlükəsizlik dizayn nümunələri ilə birləşdirərək sahəni inkişaf etdirir. O, hərtərəfli təhlil vasitəsilə SQL inyeksiya və saytlararası skript kimi ümumi təhlükələrin azaldılmasında təhlükəsiz dizayn nümunələrinin kritik əhəmiyyətini vurğulayır. Bundan əlavə, tədqiqat beynəlxalq standartlardan istifadə edərək vahid qiymətləndirmə təmin edən ən müasir təhlükəsizlik aşkarlama texnologiyalarını araşdırır. Güclü qiymətləndirmə üsulları yaratmaq və davamlı təkmilləşdirməni təşviq etməklə, bu tədqiqat veb tətbiqi təhlükəsizliyində yeni meyarlar müəyyən etmək, inkişaf edən kibertəhlükələrə qarşı dayanıqlığı artırmaq və daha təhlükəsiz veb proqramların inkişafına töhfə vermək məqsədi daşıyır.

Tədqiqatın praktiki və nəzəri əhəmiyyəti. Tədqiqat ümumi təhlükələrin azaldılmasında veb tətbiqi təhlükəsizlik prinsiplərinin və müxtəlif dizayn nümunələrinin effektivliyinin başa düşülməsinə kömək edir. Bu, kibertəhlükəsizlik və proqram mühəndisliyi fənləri üzrə biliklər toplusuna töhfə verir. Dissertasiyanın praktik əhəmiyyəti ondan ibarətdir ki, tədqiqat zamanı əldə edilmiş nəzəri nəticələr və praktik tövsiyələr universitetlərdə istifadə oluna bilər. Bundan əlavə, əsas elmi

materiallardan və dissertasiya tədqiqatlarının nəticələrindən tədris prosesində istifadə edilə bilər.

Tədqiqatın həcmi və quruluşu: Dissertasiyanın strukturu giriş, üç fəsil, doqquz paraqraf, nəticə və 68 istifadə olunmuş ədəbiyyat siyahısından ibarətdir.

I FƏSİL. VEB TƏTBİQLƏRİNİN KONSEPTUAL NƏZƏRİ ƏSASLARI VƏ VEB TƏTBİQİNİN TƏHLÜKƏSİZLİYİ DİZAYN NÜMUNƏLƏRİNƏ GİRİŞ

1.1. Veb tətbiqlərinin konseptual nəzəri əsasları

Veb tətbiqləri bir şəbəkə, adətən internet üzərindən veb brauzer vasitəsilə əldə edilən proqramlardır. Onlar istifadəçi interfeysi üçün HTML, CSS və JavaScript kimi veb texnologiyalarından və funksionallığı təmin etmək üçün verilənlər bazası, serverlər və proqram framevorkləri kimi arxa texnologiyalardan istifadə edirlər. Müştəri-server arxitekturası, verilənlərin emalı və saxlanmasının baş verdiyi müştərilər, adətən veb-brauzerlər və serverlər arasında əlaqə və qarşılıqlı əlaqəni təyin edən veb proqramlar sahəsində təməl strukturdur. Bu arxitektura tapşırıqların bölüşdürülməsini asanlaşdırır, səmərəli kommunikasiya və resursların idarə olunmasına imkan verir. Müştəri-server arxitekturasında tapşırıqlar müştərilər və serverlər arasında bölüşdürülür. Müştərilər, adətən veb brauzerlər, istifadəçi qarşılıqlı əlaqəsini və təqdimat məntiqini idarə edir, serverlər isə məlumatların saxlanmasını, biznes məntiqini və emalını idarə edir (Chapple, Seid, 2020.). Müştərilər və serverlər arasında əlaqə sorğu-cavab nümunəsinə uyğundur. Müştərilər sorğuları serverlərə göndərir, sonra bu sorğuları emal edir və tələb olunan məlumatları ehtiva edən cavabları qaytarır və ya tələb olunan hərəkətləri yerinə yetirir. Müştəri-server arxitekturası miqyaslana bilən həlləri təmin edir. Serverlər yükü paylamaq üçün daha çox server əlavə etməklə üfüqi və ya avadanlıqları təkmilləşdirməklə şaquli olaraq ölçülə bilər. Bu miqyaslılıq proqramların performansını itirmədən artan sayda istifadəçi və məlumatları idarə edə bilməsini təmin edir. Serverlər məlumatları saxlayır və idarə edir, çoxsaylı müştərilər arasında ardıcıllıq və bütövlüyü təmin edir. Mərkəzləşdirilmiş məlumat idarəetməsi məlumatlara girişi və texniki xidməti asanlaşdırır, məlumat uyğunsuzluğu və ya itkisi riskini azaldır. Mərkəzləşdirilmiş serverlər autentifikasiya, avtorizasiya və şifrələmə kimi təhlükəsizlik tədbirlərinin həyata keçirilməsini təmin edir. Bu, həssas məlumatları qorumaq və icazəsiz girişin qarşısını almaqla sistemin ümumi təhlükəsizliyini artırır. Müştərilər və serverlər müxtəlif platformalarda və texnologiyalarda işləyə bilər. Bu

çeviklik müxtəlif istifadəçi seçimlərinə və cihaz imkanlarına cavab verən çarpaz platforma proqramlarının inkişafına imkan verir. Müştəri-server arxitekturası tətbiq vəziyyətini idarə etmək üçün müxtəlif üsulları dəstəkləyir. Sessiyalar, kukilər və tokenlər istifadəçi seanslarını saxlamaq və çoxsaylı qarşılıqlı əlaqədə vəziyyəti qorumaq üçün ümumi istifadə olunan mexanizmlərdir. Müştərilər və serverlər arasında tapşırıqları paylamaqla, müştəri-server arxitekturası xətaya dözümlülüyünü yaxşılaşdırır. Bir server uğursuz olarsa, digərləri tətbiqin funksionallığına minimal pozulma təmin etməklə işləməyə davam edə bilər. Müştəri-server arxitekturası performansı yaxşılaşdırmaq üçün optimallaşdırmalara imkan verir. Gecikməni azaltmaq və istifadəçi təcrübəsini artırmaq üçün keşləmə, yük balanslaşdırma və CDN kimi üsullardan istifadə edilə bilər. Müştəri-server arxitekturasının modul xarakteri çevikliyi və genişlənməni asanlaşdırır. Komponentlər müstəqil olaraq əlavə edilə, çıxarıla və ya yenilənə bilər ki, bu da asan təmir və gələcək təkmilləşdirmələrə imkan verir. Veb proqramların müştəri-server arxitekturası müasir rəqəmsal mühitlərin tələblərinə cavab verən, genişlənə bilən, təhlükəsiz və səmərəli sistemlərin qurulması üçün güclü çərçivə təmin edir.

Veb proqramlarının backend komponenti məlumatların işlənməsi, saxlanması və biznes məntiqini idarə edən əsas kimi xidmət edir. O, frontend istifadəçi sorğuları və verilənlər bazası arasında qarşılıqlı əlaqənin idarə edilməsinə cavabdehdir. Backendin əsasını müştərilərdən (brauzerlər, mobil proqramlar və.s.) daxil olan HTTP sorğularının idarə edilməsinə cavabdeh olan server təşkil edir. Serverlər Node.js, Python (Django və ya Flask kimi framevorklərlə), Ruby on Rails, Java (Spring Boot ilə) və ya PHP (Laravel kimi framevorklərlə) kimi müxtəlif texnologiyalardan istifadə etməklə həyata keçirilə bilər (Dhillon, Backhouse, 2020). Marşrutlaşdırma daxil olan sorğuların idarə olunması üçün müvafiq koda necə yönəldildiyini müəyyən edir. O, tətbiqin son nöqtələrini müəyyənləşdirir və onları xüsusi funksiyalara və ya nəzarətçilərə uyğunlaşdırır. Marşrutlaşdırma mexanizmləri veb framevorklər tərəfindən təmin edilir və bu, backend kod bazasının təşkili üçün çox vacibdir. Backend komponentləri verilənlərin saxlanması, əldə edilməsi və dəyişdirilməsi üçün verilənlər bazası ilə qarşılıqlı əlaqədə olur. Bu qarşılıqlı əlaqə sorğuların icrasını, əməliyyatları

idarə etməyi və məlumatların bütövlüyünü təmin etməyi əhatə edir. Populyar verilənlər bazalarına PostgreSQL, MySQL və SQLite kimi SQL əsaslı verilənlər bazası, həmçinin MongoDB və Redis kimi NoSQL verilənlər bazaları daxildir. Təhlükəsiz veb proqramlar istifadəçinin autentifikasiyası (istifadəçinin şəxsiyyətinin yoxlanması) və avtorizasiya (istifadəçilərə hansı hərəkətləri yerinə yetirməyə icazə verildiyini müəyyən etmək) üçün mexanizmlər tələb edir. Bu, sessiyalar, tokenlər (JWT), OAuth və şəxsiyyət təminatçıları ilə inteqrasiya kimi üsulları əhatə edir. Backend komponentləri tez-tez API-ləri ərsəyə gətirir ki, bu da xarici xidmətlərə və ya frontend proqramlarına onlarla əlaqə saxlamağa imkan verir. API-lər mövcud son nöqtələri, sorğu/cavab formatlarını və autentifikasiya mexanizmlərini müəyyən etməklə, backend və xarici qurumlar arasında müqaviləni müəyyən edir. Middleware proqram funksiyaları daxil olan sorğuları və cavabları ələ keçirərək əlavə emal etməyə imkan verir. Ümumi middleware proqram tapşırıqlarına giriş, səhvlərin idarə edilməsi, sorğunun təhlili və təhlükəsizlik təkmilləşdirmələri daxildir. Bütövlükdə, veb proqramların backend komponenti bütün sistemin funksionallığının, təhlükəsizliyinin və performansının təmin edilməsində mühüm rol oynayır. Onun dizaynı və tətbiqi istifadəçi təcrübəsinə və tətbiqin genişlənməsinə əhəmiyyətli dərəcədə təsir göstərir.

Veb proqramlarının frontend komponenti məlumatların təqdim edilməsi və istifadəçilərlə qarşılıqlı əlaqə üçün cavabdeh olan istifadəçinin gördüyü hissədir. O, veb brauzerdə vizual və interaktiv təcrübə yaradan müxtəlif texnologiyalar və elementləri əhatə edir. HTML (Hypertext Markup Language) başlıqlar, paraqraflar, şəkillər, düymələr və formalar kimi elementləri müəyyən etməklə veb səhifələrin strukturunu formalaşdırır. O, digər texnologiyaların üzərində qurulduğu əsas skleti təmin edir (Pfleeger, Pfleeger, Margulies, 2019). CSS (Cascading Style Sheets) HTML elementlərinin üslubu, tərtibata, tipoqrafiyaya, rənglərə və vizual effektlərə nəzarət etmək üçün istifadə olunur. O, tərtibatçılara veb səhifələrin görünüşünü fərdiləşdirməyə və müxtəlif cihazlar və brauzerlərdə ardıcıllığı təmin etməyə imkan verir. JavaScript veb səhifələrə interaktivlik və davranış əlavə edən dinamik proqramlaşdırma dilidir. O, HTML və CSS ilə manipulyasiya etməyə, istifadəçi daxiletmələrini idarə etməyə, serverlərə asinxron sorğular etməyə (AJAX) və səhifəni

yenidən yükləmədən məzmunu dinamik olaraq yeniləməyə imkan verir (Tək Səhifə Tətbiqləri və ya SPA). React.js, Angular, Vue.js və Svelte kimi frontend framevorkləri və kitabxanaları inkişafı asanlaşdırmaq və kodun davamlılığını artırmaq üçün əvvəlcədən qurulmuş komponentlər, marşrutlaşdırma mexanizmləri təmin edir. Onlar mürəkkəb istifadəçi interfeyslərinin yaradılmasını asanlaşdırır və tətbiqin vəziyyətini effektiv idarə etməyə kömək edir. Fərqli ekran ölçüləri və qətnamələri olan müxtəlif cihazların çoxalması ilə cavab verən dizayn üsulları veb proqramların uyğunlaşmasını və masaüstü kompüterlər, noutbuklar, planşetlər və smartfonlarda optimal istifadəçi təcrübəsini təmin etməsini təmin edir. Bu, tez-tez CSS media sorğularından və çevik tərtibat üsullarından istifadəni əhatə edir. Əlçatanlıq mülahizələri veb proqramların əlilliyi olan insanlar tərəfindən istifadə oluna biləcəyini təmin edir. Bu, şəkillər üçün alternativ mətnin təqdim edilməsini, məzmunun bütün istifadəçilər üçün qavranılan, işlək və başa düşülən olması üçün klaviatura naviqasiyasını, semantik HTML işarələməsini və digər təcrübələri təmin edir. Frontend tərtibatçıları fayl ölçülərini (HTML, CSS, JavaScript) minimuma endirməklə, HTTP sorğularının sayını, tənbəl yükləmə resurslarını azaltmaqla və keşləmə strategiyalarından istifadə etməklə sürət və performans üçün veb proqramlarını optimallaşdırır. Performansın optimallaşdırılması, xüsusən aşağı band genişliyi və ya mobil şəbəkələrdə problemsiz istifadəçi təcrübəsi təqdim etmək üçün çox vacibdir. Frontend komponentləri tez-tez verilənləri asinxron şəkildə əldə etmək və təqdim etmək üçün API vasitəsilə backend xidmətləri ilə əlaqə qurur (Gollmann, 2016). Bu qarşılıqlı əlaqə tam səhifəni yenidən yükləmədən dinamik məzmun yeniləmələrinə və real vaxt funksionallığına imkan verir. Veb proqramların frontend komponenti cəlbedici istifadəçi interfeysləri yaratmaq, interaktivliyi təmin etmək və müxtəlif cihazlar və platformalarda istifadə və əlçatanlığı təmin etmək üçün vacibdir. O, HTML, CSS, JavaScript, framevorklər və ən yaxşı təcrübələri birləşdirirərək, internetdə zəngin, həssas və effektiv istifadəçi təcrübələri təqdim edir.

Veb proqramlarının verilənlər bazası xüsusiyyəti məlumatların saxlanmasına, axtarışına və manipulyasiyasına imkan verən əsas funksiyanı yerinə yetirir və bununla da dinamik məzmunun çatdırılmasını və istifadəçi ilə qarşılıqlı əlaqəni asanlaşdırır. Bu

xüsusiyyət veb proqramlar daxilində məlumatların səmərəli idarə edilməsi üçün vacib olan müxtəlif komponentləri və funksiyaları əhatə edir. Verilənlər bazaları istifadəçi məlumatı, məzmun, konfiqurasiya parametrləri və.s. daxil olmaqla çoxlu strukturlaşdırılmış və ya strukturlaşdırılmamış məlumatları saxlayır. Veb tətbiqlərində istifadə olunan ümumi verilənlər bazası növlərinə əlaqəli verilənlər bazaları (məsələn, MySQL, PostgreSQL) və NoSQL verilənlər bazaları (məsələn, MongoDB, Cassandra) daxildir, hər biri müxtəlif məlumat modelləri və genişlənmə tələbləri üçün uyğundur. Veb proqramları istifadəçi sorğularına uyğunlaşdırılmış dinamik məzmun yaratmaq üçün verilənlər bazalarından məlumatları tez-tez alır. Sorğular, əlaqəli verilənlər bazaları üçün SQL (Structured Query Language) və ya NoSQL verilənlər bazaları üçün API-lər kimi sorğu dillərindən istifadə etməklə qurulur ki, bu da tərtibatçılara istifadəçi girişi və ya əvvəlcədən müəyyən edilmiş meyarlar əsasında xüsusi məlumat alt dəstlərini əldə etməyə imkan verir. Verilənlər bazası xüsusiyyətləri tərtibatçılara daxil etmək, silmək, yeniləmək və əldə etmək kimi müxtəlif məlumatların manipulyasiya əməliyyatlarını yerinə yetirmək imkanı verir. Bu əməliyyatlar məlumatların bütövlüyünü qorumaq, istifadəçi qarşılıqlı əlaqələrini idarə etmək və veb tətbiqləri daxilində biznes məntiqini həyata keçirmək üçün vacibdir. Çox istifadəçi mühitlərində paralelliyə nəzarət mexanizmləri paylaşılan məlumatlara eyni vaxtda girişi idarə etməklə məlumatların ardıcıllığını təmin edir. Kilidləmə, əməliyyatlar və optimist paralellik nəzarəti kimi üsullar münaqişələrin qarşısını alır və verilənlər bazasının bütövlüyünü qoruyur, istifadəçilər arasında qüsursuz əməkdaşlıq və qarşılıqlı əlaqəni təmin edir. Verilənlər bazası xüsusiyyətlərinə həssas məlumatları icazəsiz giriş, manipulyasiya və ya pozuntulardan qorumaq üçün güclü təhlükəsizlik tədbirləri daxildir (Pfleeger, Pfleeger, Margulies, 2019). RBAC, şifrələmə, parametrləşdirilmiş sorğular və məlumatların maskalanması verilənlər bazalarını qorumaq və təhlükəsizlik risklərini azaltmaq üçün həyata keçirilən ümumi təhlükəsizlik təcrübələridir. Verilənlər bazaları artan məlumat həcmini və artan istifadəçi trafikini təmin etmək üçün üfüqi və ya şaquli miqyasda nəzərdə tutulmuşdur. Parçalama, təkrarlama, keşləmə və indeksləşdirmə kimi xüsusiyyətlər verilənlər bazası performansını optimallaşdırır, hətta yüksək yükləmə şəraitində belə sürətli məlumatların axtarışını və cavab

reaksiyasını təmin edir. Məlumat itkisinin qarşısını almaq və işin davamlılığını təmin etmək üçün verilənlər bazası xüsusiyyətlərinə ehtiyat nüsxə və bərpa mexanizmləri daxildir. Avtomatlaşdırılmış ehtiyat nüsxə cədvəlləri, artan ehtiyat nüsxələri və fəlakətin bərpası planları hardware nasazlıqları, proqram xətaları və ya zərərli hücumlarla bağlı riskləri azaldır. Veb proqramların verilənlər bazası xüsusiyyəti tətbiqin həyat dövrü ərzində məlumatların idarə edilməsində mühüm rol oynayır. Verilənlər bazaları səmərəli saxlama, axtarış, manipulyasiya, təhlükəsizlik, miqyaslanma və davamlılıq imkanlarını təmin etməklə tərtibatçılara istifadəçi ehtiyaclarına və biznes tələblərinə uyğunlaşdırılmış möhkəm, interaktiv və məlumatlara əsaslanan veb proqramlar yaratmağa imkan verir.

Veb proqramların API (Application Programming Interface) xüsusiyyəti müxtəlif proqram təminatı tətbiqləri arasında körpü rolunu oynayır, onlara ünsiyyət qurmağa, məlumat mübadiləsinə və funksionallıqlara maneəsiz daxil olmağa imkan verir. API-lər bir neçə əsas xüsusiyyətləri təklif edərək veb proqramlarında qarşılıqlı fəaliyyət, inteqrasiya və genişlənmə imkanı verən ayrılmaz komponentlərdir. API-lər fərqli sistemlərin, platformaların və cihazların bir-biri ilə effektiv şəkildə qarşılıqlı əlaqədə olmasına imkan verən standartlaşdırılmış rabitə protokollarını və məlumat formatlarını müəyyən edir. API-lər vasitəsilə veb proqramları üçüncü tərəf xidmətləri, verilənlər bazası və xarici sistemlərlə inteqrasiya edə bilər, məlumat mübadiləsini və funksionallığın təkrar istifadəsini asanlaşdırır. API-lər veb proqramlar tərəfindən təklif olunan xüsusi funksiyaları və ya xidmətləri ərsəyə gətirir, tərtibatçılara onlardan proqramlı şəkildə daxil olmaq və istifadə etmək imkanı verir. API-lər vasitəsilə təqdim edilən xidmətlərə autentifikasiya, məlumatların axtarışı, məlumatların manipulyasiyası, bildirişlər, ödənişlərin işlənməsi və.s. daxil ola bilər ki, bu da tərtibatçılara müxtəlif və xüsusiyyətlərlə zəngin proqramlar yaratmaq imkanı verir (Stuart, Joel, George, 2020). API-lər məlumatların axtarışı, təqdim edilməsi və manipulyasiyası üçün son nöqtələr və üsullar təmin etməklə veb proqramlar və xarici sistemlər arasında problemsiz məlumat mübadiləsini asanlaşdırır. API-lər vasitəsilə mübadilə edilən məlumatlar uyğunluğu və inteqrasiya asanlığını təmin edərək, JSON (JavaScript Object Notation) və ya XML (Extensible Markup Language) kimi müxtəlif

formatlarda ola bilər. Veb proqramları tez-tez tərtibatçılar və ya üçüncü tərəf provayderlər tərəfindən fərdiləşdirmə və genişləndirilməsini təmin etmək üçün API-ləri ərsəyə gətirir. Bu API-lər tərtibatçılara tətbiqin funksionallığını genişləndirməyə, əlavə funksiyaları birləşdirməyə və ya tamamlayıcı xidmətlər qurmağa, innovasiya və ekosistemin böyüməsini təşviq etməyə imkan verir. API xüsusiyyətlərinə müxtəlif API versiyaları arasında problemsiz keçidi təmin etmək üçün versiya mexanizmləri daxildir. Versiyalaşdırma tərtibatçılara mövcud inteqrasiyaları pozmadan və ya köhnə API versiyalarına əsaslanan tətbiqləri pozmadan təkmilləşdirmələr təqdim etməyə, problemləri həll etməyə və ya köhnəlmiş funksiyaları ləğv etməyə imkan verir. API-lər həssas məlumatları qorumaq və icazəsiz giriş və ya sui-istifadənin qarşısını almaq üçün autentifikasiya, avtorizasiya, sürətin məhdudlaşdırılması və şifrələmə kimi təhlükəsizlik tədbirlərini həyata keçirir. OAuth, JWT (JSON Web Tokens), API açarları və HTTPS API son nöqtələrini qorumaq və məlumatların məxfiliyini və bütövlüyünü təmin etmək üçün istifadə edilən ümumi təhlükəsizlik protokollarıdır. Hərtərəfli sənədlər və tərtibatçı dəstəyi API-lərin əsas xüsusiyyətləridir və tərtibatçıları API-nin inteqrasiyası və səmərəli istifadəsi üçün aydın təlimatlar, nümunələr və resurslarla təmin edir. Yaxşı sənədləşdirilmiş API-lər tərtibatçı təcrübəsini təkmilləşdirir, inteqrasiya səylərini sürətləndirir və tərtibatçı icması daxilində əməkdaşlığı gücləndirir. Veb proqramların API xüsusiyyəti qarşılıqlı fəaliyyətin, inteqrasiyanın, genişlənmənin və fərdiləşdirmənin təmin edilməsində mühüm rol oynayır. Standartlaşdırılmış interfeyslər, xidmətə giriş, məlumat mübadiləsi imkanları, təhlükəsizlik tədbirləri, versiya dəstəyi və tərtibatçı resursları təmin etməklə, API-lər tərtibatçılara müxtəlif istifadə hallarına və tələblərinə uyğunlaşdırılmış bir-biri ilə əlaqəli, miqyaslana bilən və innovativ veb proqramlar yaratmaq imkanı verir (Thomas, 2020).

Veb tətbiqlərinin miqyaslılıq xüsusiyyəti, performansı, etibarlılığı və cavab vermə qabiliyyətini qoruyarkən tətbiqin artan yükləri və artan istifadəçi bazalarını idarə etmək qabiliyyətinə malikdir. Miqyaslılıq veb proqramlarının xidmət keyfiyyətində fasilə və ya problem yaşamadan trafikdə, məlumat həcmində və istifadəçi qarşılıqlı fəaliyyətindəki dalğalanmaları qəbul edə bilməsini təmin etmək üçün çox vacibdir.

Veb proqramları yükü çoxsaylı serverlər və ya instansiyalar arasında paylayaraq üfüqi miqyaslılığa nail olur. Bu yanaşma artan trafik və iş yükünü idarə etmək üçün əlavə serverlər və ya bulud nümunələri kimi daha çox hesablama resurslarının əlavə edilməsini nəzərdə tutur. Yük balanslaşdırıcıları daxil olan sorğuları bu resurslar arasında bərabər paylayır, optimal istifadə və performansı təmin edir. Şaquli miqyaslılıq artan tələbi idarə etmək üçün CPU, yaddaş və ya saxlama tutumu kimi fərdi server resurslarının təkmilləşdirilməsini nəzərdə tutur. Şaquli miqyaslama dərhal performans təkmilləşdirmələrini təmin edə bilsə də, üfüqi miqyaslama ilə müqayisədə iqtisadi səmərəlilik və maksimum genişlənmə potensialı baxımından məhdudiyyətlərə malik ola bilər. REST (REpresentational State Transfer) və mikroservislər kimi vətəndaşlığı olmayan(stateless) dizayn nümunələri sessiya məlumatlarının saxlanması və ya müştəri sorğuları arasında vəziyyətə uyğun əlaqənin saxlanması ehtiyacını aradan qaldırmaqla miqyaslılığı təşviq edir. Vətəndaşlığı olmayan proqramlar sorğuları serverlər arasında daha asan paylaya bilər, çünki hər bir sorğu emal üçün bütün lazımi məlumatları ehtiva edir və daha yaxşı üfüqi miqyaslılığa imkan verir. Tez-tez əldə edilən məlumatların və ya hesablanmış nəticələrin müxtəlif səviyyələrdə (məsələn, müştəri, server, verilənlər bazası tərəfi) keşləşdirilməsi təkrar hesablamalara və verilənlər bazası sorğularına ehtiyacı azaldır, tətbiqin performansını və miqyasını yaxşılaşdırır. CDN statik aktivləri keşləyir və onlara istifadəçilərə daha yaxın olan kənar serverlərdən xidmət edir, gecikməni azaldır və mənbə serverlərindən trafiki boşaltır (Schneier, 2020). Verilənlər Bazasının miqyası artan məlumat həcmini və paralel əməliyyatları idarə etmək üçün verilənlər bazası performansını və tutumu optimallaşdırmağı nəzərdə tutur. Parçalanma, təkrarlama, bölmə və paylanmış verilənlər bazaları kimi texnikalar verilənlər bazalarının üfüqi miqyasını genişləndirməyə imkan verir və onlara artan məlumat dəstlərini və istifadəçi fəaliyyətini uyğunlaşdırmağa imkan verir. Resurs tələb edən və ya çox vaxt aparan tapşırıqların fon proseslərinə və ya asinxron növbələrə yüklənməsi tətbiqin cavab vermə qabiliyyətini və miqyaslılığını artırır. Mesaj növbələri, tapşırıq növbələri və hadisəyə əsaslanan arxitekturalar komponentləri ayırır və tapşırıqların paralel işlənməsini təmin edərək ötürmə qabiliyyətini və miqyaslılığı yaxşılaşdırır. Avtomatik

miqyaslama xüsusiyyətləri CPU istifadəsi, sorğu gecikməsi və ya daxil olan trafik kimi əvvəlcədən təyin edilmiş ölçülərə əsasən hesablama resurslarını avtomatik tənzimləyir. Bulud platformaları avtomatik miqyaslama imkanları təklif edir, veb tətbiqlərə tələb dəyişkənliyinə cavab olaraq resursları dinamik şəkildə böyütməyə və ya azaltmağa imkan verir, optimal performansı və qənaətcilliyi təmin edir. Bu miqyaslılıq xüsusiyyətlərini tətbiq etməklə veb tətbiqləri performansı, etibarlılığı və istifadəçi təcrübəsini qoruyarkən artan iş yüklərini, istifadəçi fəaliyyətini və məlumat həcmlərini effektiv şəkildə idarə edə bilər. Miqyaslılıq veb proqramlarının bugünkü dinamik rəqəmsal mənzərədə dəyişən tələblərə və inkişaf imkanlarına cavab verən, əlçatan və uyğunlaşa bilən qalmasını təmin edir.

1.2. Veb tətbiqinin təhlükəsizliyi dizayn nümunələrinə giriş

Veb tətbiqi təhlükəsizlik dizayn nümunələri ümumi təhlükəsizlik problemlərini və zəiflikləri həll etməklə tərtibatçılara təhlükəsiz veb proqramlar yaratmağa kömək edən əsas planlardır. Bu nümunələr inyeksiya hücumları, XSS, CSRF və bir çox başqaları kimi müxtəlif təhlükələrdən qorunmaq üçün təhlükəsizlik nəzarətlərinin və ən yaxşı təcrübələrin necə həyata keçirilməsinə dair təlimat verir. Veb tətbiqi təhlükəsizliyi dizaynı müxtəlif kibertəhlükələrdən qorunmaq üçün çox vacibdir və girişin doğrulanması bu müdafiənin əsas aspektidir. Daxiletmənin yoxlanılması veb tətbiqinə daxil edilmiş məlumatların emal edilməzdən və ya saxlanmazdan əvvəl müəyyən edilmiş meyarlara cavab verməsini təmin edir və bununla da inyeksiya hücumları, məlumatların pozulması və digər təhlükəsizlik zəiflikləri kimi riskləri azaldır. Sosial media platforması üçün veb-əsaslı qeydiyyat forması kontekstində daxiletmənin təsdiqlənməsi nümunəsini nəzərdən keçirək. Qeydiyyat forması adətən istifadəçi adı, e-poçt ünvanı, parol, doğum tarixi və profil şəkli kimi məlumatları toplayır (Darril, 2020). İstifadəçi adı sahəsi uzunluq məhdudiyyətləri (məsələn, 4 və 20 simvol arasında), icazə verilən simvollar (məsələn, əlifba simvolları və alt xətt) və unikallıq (məsələn, istifadəçi adı) kimi müəyyən meyarlara cavab verməsi üçün təsdiqlənməlidir. (dublikat istifadəçi adlarının qarşısını almaq üçün). E-poçt ünvanları onların düzgün formata (məsələn, `example@email.com`) uyğun olmasına və sistem

daxilində unikal olmasına əmin olmaq üçün təsdiq edilməlidir. Bundan əlavə, domenin mövcudluğunu və e-poçt ünvanının birdəfəlik e-poçt provayderindən olmadığını yoxlamaq üçün bəzi əsas yoxlamalar aparıla bilər. Parollar minimum uzunluq tələbləri, həm böyük, həm də kiçik hərflərin, rəqəmlərin və xüsusi simvolların daxil edilməsi kimi təhlükəsizlik standartlarına cavab verməsi üçün təsdiqlənməlidir. Bundan əlavə, açıq mətnin saxlanmasının qarşısını almaq üçün parollar saxlanmazdan əvvəl heşləşdirilməlidir. İstifadəçilərin minimum yaş tələbinə (məsələn, COPPA uyğunluğu üçün 13 yaş) malik olmasını təmin etmək üçün doğum tarixi sahələri təsdiqlənməlidir. Bu, daxil edilmiş tarixi, cari tarixlə müqayisə etməklə və onun yaş meyarlarına uyğunluğunu yoxlamaqla edilə bilər. Əgər qeydiyyat formasında profil şəklini yükləmək seçimi varsa, yüklənmiş fayllar düzgün fayl növündə (məsələn, JPEG, PNG) və ölçü məhdudiyyətləri daxilində olduğundan əmin olmaq üçün təsdiq edilməlidir. Bundan əlavə, icra edilə bilən faylların və ya digər potensial zərərli məzmunun yüklənməsinin qarşısını almaq üçün tədbirlər görülməlidir. Güclü giriş yoxlama mexanizmlərinin veb proqram dizaynına daxil edilməsi SQL inyeksiya hücumları, XSS və komanda inyeksiya hücumları kimi ümumi zəifliklərin riskini azaltmağa kömək edir. Ciddi yoxlama qaydalarını tətbiq etməklə tərtibatçılar tətbiqin təhlükəsizlik vəziyyətini gücləndirə və istifadəçi məlumatlarını zərərli istismardan qoruya bilərlər.

Veb tətbiqi təhlükəsizliyi dizaynının mühüm aspekti, zərərli skriptlərin və ya məzmunun inyeksiyasının qarşısını almaq üçün istifadəçilərə təqdim olunan hər hansı məlumatın lazımi qaydada təmizləndiyini təmin edən çıxış kodlaşdırmasıdır.

Veb səhifəsində istifadəçi şərhlərini göstərən sadə bir veb tətbiqini nəzərdən keçirək. Müvafiq çıxış kodlaşdırması olmadan, HTML və ya JavaScript ehtiva edən istifadəçi girişi brauzer tərəfindən icra edilə bilər və bu, XSS hücumları kimi potensial təhlükəsizlik zəifliklərinə səbəb ola bilər (Harold, Micki, 2017). Burada funksiya(və ya metod) ilə `<`, `>`, `"` və `&` kimi xüsusi simvolları müvafiq HTML obyektlərinə çevirərək onları brauzer üçün zərərsiz etmək daha yaxşı təcrübə sayılır. İstifadəçi tərəfindən yaradılan məzmunu veb-səhifəyə çıxarmazdan əvvəl kodlaşdırmaqla tərtibatçılar XSS hücumlarının qarşısını ala və proqramın təhlükəsiz qalmasını təmin

edə bilərlər. Çıxış kodlaşdırmasının veb tətbiqinin dizaynına daxil edilməsi bir sıra təhlükəsizlik təhdidlərindən qorunmaq və istifadəçi məlumatlarının bütövlüyünü qorumaq üçün vacibdir.

Doğrulama və avtorizasiya veb proqram təhlükəsizlik dizaynının əsas komponentləridir və yalnız səlahiyyətli istifadəçilərin resurslara və funksiyalara giriş əldə etməsini təmin edir (Thomas, 2021). Bir veb tətbiqində autentifikasiya və avtorizasiyanın necə həyata keçirilə biləcəyini göstərən bir nümunəyə nəzər salaq:

İstifadəçilərə hesab qalıqlarına baxmaq və əməliyyatlar həyata keçirmək imkanı verən bank veb tətbiqini nəzərdən keçirək. Tətbiq istifadəçilərin şəxsiyyətini yoxlamaq üçün etibarlı autentifikasiya və həssas funksiyalara girişi idarə etmək üçün icazə tələb edir.

İdentifikasiyası:

İstifadəçi daxil olmağa cəhd edərkən istifadəçi adı/e-poçtu və parolunu təqdim edir.

Tətbiq parolu təhlükəsiz şəkildə hash edir və onu müvafiq istifadəçi üçün saxlanılan parol ilə müqayisə edir.

Əgər etimadnamələr uyğun gəlirsə, istifadəçi autentifikasiya olunur və sistemə giriş hüququ verilir. Əks halda, autentifikasiya uğursuz olur və istifadəçiyə giriş qadağan edilir.

İstifadəçi identifikasiyası üçün Java-da nümunə kod parçası(Şəkil 1.1):

Bu kod parçasında istifadəçi adına görə istifadəçinin bank hesabını verilənlər bazasından çəkir və parolunun düzgün olub olmadığını yoxlayır. Burada Spring freymvork-undan(Spring Security) istifadə edilərək autentifikasiya qurulmuşdur.

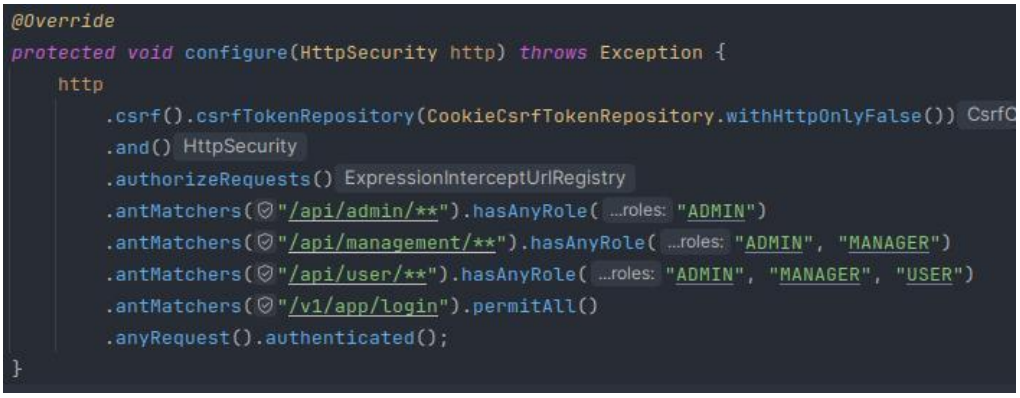
```
public boolean isValidUser(String username, String password) {  no usages
    Optional<BankAccount> bankAccount = bankAccountRepository.findByUserName(username);
    return bankAccount.map(account -> account.getUserPassword().equals(password)).orElse( t false);
}
```

Şəkil 1.1

1. Səlahiyyət:

Doğrulandıqdan sonra proqram istifadəçinin hansı hərəkətləri yerinə yetirə biləcəyini müəyyən etmək üçün onun imtiyazlarını yoxlayır.

Məsələn, adi istifadəçinin yalnız hesab balansına və əməliyyat tarixçəsinə baxmaq icazəsi ola bilər, menecer istifadəçi isə əməliyyatları təsdiqləmək və ya istifadəçi hesablarını idarə etmək kimi əlavə imtiyazlara malik ola bilər(Şəkil 1.2). Admin rolu isə adətən bütün imtiyazlara sahibdir və bu rol normal istifadəçilər üçün uyğun deyildir.



Şəkil 1.2

Veb tətbiqinin dizaynına güclü autentifikasiya və avtorizasiya mexanizmlərinin daxil edilməsi həssas məlumatların və funksiyaların icazəsiz girişdən qorunması, sistemin təhlükəsizliyini və bütövlüyünü təmin etmək üçün vacibdir.

Sessiyanın idarə edilməsi veb tətbiqi təhlükəsizlik dizaynının kritik aspektidir və istifadəçi seanslarının proqramla qarşılıqlı əlaqəsi boyunca təhlükəsiz şəkildə saxlanılmasına cavabdehdir (Whitman, Mattord, 2019). Sessiya idarəçiliyinin veb proqramda təhlükəsiz şəkildə necə həyata keçirilə biləcəyini göstərən bir nümunəni araşdıraq:

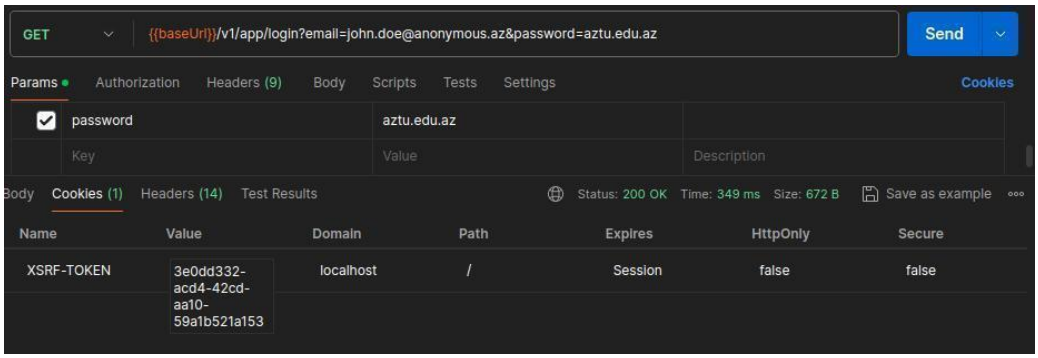
İstifadəçilərin məhsulları nəzərdən keçirə, səbətinə əşyalar əlavə edə və ödənişə davam edə biləcəyi bir e-ticarət platformasını nəzərdən keçirin. Seansın düzgün idarə edilməsi istifadəçilərin seanslarının təhlükəsiz olmasını, icazəsiz girişin qarşısının alınmasını və həssas məlumatların qorunmasını təmin edir.

1.Sessiyanın Başlanması:

İstifadəçi veb-sayta daxil olduqda, sessiya başlayır və unikal sessiya identifikatoru (sessiya ID) yaradılır.

Bu sessiya ID serverdə saxlanılır və istifadəçinin alış səbətinin məzmunu və autentifikasiya statusu kimi sessiya datası ilə əlaqələndirilir.

Bundan əlavə, sonrakı sorğular üçün istifadəçinin brauzerinə sessiya identifikatorunu ehtiva edən sessiya kukisi göndərilir. Nümunə olaraq burda Spring Security istifadə edərək CSRF tokenlərindən istifadəni konfiqurasiya etdiyimiz üçün bizə aktiv sessiya üçün kuki daxilində sessiya ID(CSRF token) göndərilir(Şəkil 1.3). Biz bu sessiya ID-sindən istifadə edərək etibarlı istifadəçi kimi vəziyyət dəyişən(POST, PUT, DELETE) əməliyyatları edə bilərik.



Şəkil 1.3

1.Sessiyanın Təsdiqlənməsi və Qorunması:

İstifadəçidən gələn hər bir sonrakı sorğuya sessiya identifikatorunu ehtiva edən sessiya kukisi daxildir.

Server sessiya identifikatorunun etibarlılığını yoxlayır və onun aktiv sessiyaya uyğun olmasını təmin edir.

Sessiya oğurlanması və fiksasiya hücumlarının qarşısını almaq üçün sessiya identifikatorları vaxtaşırı və ya autentifikasiya kimi əhəmiyyətli hadisələr zamanı bərpa edilməlidir.

1.Sessiyanın vaxtı:

İstifadəçinin sessiyanı nəzarətsiz tərk etməsi halında sessiyanın oğurlanması riskini azaltmaq üçün sessiyaların məhdud ömrü olmalıdır.

Sessiya fasiləsi həddinə çatdıqda, istifadəçidən yenidən autentifikasiya tələb olunmaqla sessiya etibararsız sayılmalıdır.

Seansın düzgün qurulması, doğrulama, qorunma və vaxt aşımı mexanizmləri kimi güclü sessiya idarəetmə üsullarını tətbiq etməklə veb proqramlar təhlükəsiz istifadəçi qarşılıqlı əlaqəni təmin edə, həssas məlumatları qoruya və sessiya ilə bağlı zəifliklərin riskini azalda bilər.

Veb tətbiqlərində müştərilər və serverlər arasında ötürülən həssas məlumatların qorunması üçün təhlükəsiz rabitə çox vacibdir. Şifrələmə və təhlükəsiz protokolların tətbiqi məlumatların məxfi qalmasını və ötürülmə zamanı bütövlüyün qorunmasını təmin edir (Bill, 2015). Nümunə olaraq HTTPS (HTTP Secure) istifadə edərək veb proqramında təhlükəsiz rabitəni təsvir edək:

1.HTTPS İcrası:

HTTPS Nəqliyyat Layer Təhlükəsizliyi (TLS) və ya onun sələfi Secure Sockets Layer (SSL) istifadə edərək müştərinin brauzeri ilə server arasında mübadilə edilən məlumatları şifrələyir.

Bu, etibarlı CA-dan SSL/TLS sertifikatı əldə etməyi və məlumatların şifrələnməsi üçün bu sertifikatdan istifadə etmək üçün veb serverin konfiqurasiyasını tələb edir.

İstifadəçi HTTPS vasitəsilə vebsayta daxil olduqda, server təhlükəsiz əlaqə yaradaraq müştəriyə SSL/TLS sertifikatını göndərir.

HTTPS-i aktivləşdirmək üçün Apache server konfiqurasiyasında nümunə kod parçası(Şəkil 1.4):

```
<VirtualHost *:443>
    ServerName example.com
    SSLEngine on
    SSLCertificateFile /path/to/certificate.crt
    SSLCertificateKeyFile /path/to/private.key
</VirtualHost>
```

Şəkil 1.4

HTTPS yönləndirməsini məcbur edilir:

Bütün trafikin şifrələnməsini təmin etmək üçün veb proqramlar HTTP sorğularını HTTPS-ə yönləndirməlidir.

Buna server tərəfində yönləndirmə qaydalarını konfiqurasiya etməklə və ya HTTP Strict Transport Security (HSTS) başlıqlarından istifadə etməklə nail olmaq olar.

HTTP-ni HTTPS-ə yönləndirmək üçün Apache konfiqurasiyasında nümunə kod parçası(Şəkil 1.5):

```
<VirtualHost *:80>
    ServerName example.com
    Redirect permanent / https://example.com/
</VirtualHost>
```

Şəkil 1.5

CSP:

CSP veb-səhifəyə hansı məzmun mənbələrinin yüklənməsinə icazə verildiyini müəyyən etməklə XSS hücumlarının qarşısını almağa kömək edən əlavə təhlükəsizlik tədbiridir.

O, veb səhifələrə yeridilmiş zərərli skriptlərlə bağlı riskləri azaltmağa kömək edir.

HTTPS tətbiq etməklə, təhlükəsiz yönləndirməni həyata keçirməklə və CSP kimi əlavə təhlükəsizlik tədbirlərindən istifadə etməklə, veb proqramlar həssas məlumatları zərərli aktyorlar tərəfindən tutulmaqdan və dəyişdirilməkdən qoruyaraq təhlükəsiz rabitə kanalları yarada bilər. Bu, məlumatların ötürülməsində məxfilik, bütövlük və orijinallığı təmin edərək, tətbiqin ümumi təhlükəsizlik vəziyyətini artırır.

II FƏSİL. TƏHLÜKƏSİZ DİZAYN NÜMUNƏLƏRİNİN ƏHƏMİYYƏTİ VƏ VEB TƏTBİQLƏRİ ÜÇÜN ÜMUMİ TƏHDİDLƏR

2.1. Təhlükəsiz dizayn nümunələrinin əhəmiyyəti

Təhlükəsiz dizayn nümunələri proqram təminatı və sistemlərin inkişafında mühüm əhəmiyyət kəsb edir, çünki onlar təhlükəsizlik təhdidlərinə və zəifliklərə qarşı davamlı olan proqramların yaradılması üçün çərçivə təmin edir.

Cədvəl 2.1

Təhlükəsiz dizayn nümunələrinin əhəmiyyəti

Zəifliklərin qarşısının alınması
Risklərin Azaldılması
Uyğunluq Tələbləri
Xərc-Səmərəlilik
Etimadın qorunub saxlanılması

Mənbə: Peter T, 2022, məlumatlarına əsasən müəllif tərəfindən hazırlanmışdır.

Təhlükəsiz dizayn nümunələri proqram sistemlərinin saysız-hesabsız təhlükəsizlik təhdidlərinə qarşı gücləndirilməsində mühüm rol oynayır. Təhlükəsizlik mülahizələrini dizayn mərhələsinə inteqrasiya etməklə tərtibatçılar potensial zəifliklər təxmin edə və riskləri effektiv şəkildə azaltmaq üçün qabaqlayıcı tədbirlər həyata keçirə bilərlər. Proqram təminatının hazırlanmasında ən çox yayılmış təhlükələrdən biri, davranışını manipulyasiya etmək və ya məlumatlara icazəsiz giriş əldə etmək üçün proqrama zərərli kodun yeridilməsi zamanı inyeksiya hücumlarıdır. Parametrləşdirilmiş sorğular və girişin yoxlanılması kimi təhlükəsiz dizayn nümunələri istifadəçi girişlərini təmizləmək və zərərli kodun icrasının qarşısını almaqla inyeksiya hücumlarının qarşısını almağa kömək edir (Peter, 2022). İnkişafın həyat dövrünün əvvəlində bu nümunələri qəbul etməklə tərtibatçılar inyeksiya zəifliklərinə qarşı güclü müdafiə mexanizmi qura bilərlər. Digər ümumi təhlükəsizlik tələsi autentifikasiya

mexanizmlərinin düzgün tətbiq olunmaması zamanı baş verən və həssas resurslara icazəsiz girişə səbəb olan pozulmuş autentifikasiyadır. Çox faktorlu autentifikasiya və sessiyanın idarə edilməsi ən yaxşı təcrübələr kimi təhlükəsiz dizayn nümunələri, autentifikasiya mexanizmlərini gücləndirmək və etimadnamənin və sessiyanın oğurlanmasına qarşı qorumaqdır. Sistem dizaynı zamanı bu nümunələrə riayət etməklə tərtibatçılar autentifikasiya proseslərini gücləndirə və istifadəçi şəxsiyyətlərinin məxfiliyini və bütövlüyünü qoruya bilərlər. Həssas məlumatlara məruz qalma proqram sistemləri üçün daha bir əhəmiyyətli təhlükə yaradır, burada məxfi məlumat təsadüfən icazəsiz şəxslərə açıqlanır. Məlumatların şifrələnməsi və təhlükəsiz ötürmə protokolları kimi təhlükəsiz dizayn nümunələri həssas məlumatları yad gözlərdən qoruyur və məlumatların pozulması riskini azaldır. Tərtibatçılar əvvəldən bu nümunələrdən istifadə etməklə həssas məlumatın saxlanmadan ötürülməsinə qədər bütün ömrü boyu adekvat şəkildə qorunmasını təmin edə bilərlər. Əslində, təhlükəsiz dizayn nümunələri təhlükəsizlik zəifliklərini qabaqlayıcı şəkildə aradan qaldırmaq və proqram sistemlərini potensial təhlükələrə qarşı gücləndirmək üçün qabaqlayıcı tədbirlər kimi xidmət edir. Proqram tərtibatçıları təhlükəsizlik mülahizələrini proqram təminatının dizaynına daxil etməklə, zəifliklərin daxil olma ehtimalını minimuma endirə və inyeksiya, sınmış autentifikasiya və həssas məlumatların ifşası kimi hücumlara qarşı tətbiqlərinin davamlılığını gücləndirə bilərlər. Bu proaktiv yanaşma yalnız proqram sistemlərinin təhlükəsizlik vəziyyətini gücləndirmir, həm də istifadəçilər və maraqlı tərəflər arasında proqram təminatının etibarlılığı və bütövlüyünə inam yaradır.

Təhlükəsizlik pozuntuları məlumatların pozulmasından tutmuş maliyyə itkilərinə və reputasiyaya zərər kimi bir sıra zərərli nəticələri əhatə edən təşkilatlar üçün kritik təhlükə yaradır. Təhlükəsiz dizayn nümunələri risklərin azaldılması strategiyalarının arsenalında əvəzolunmaz alətlər kimi dayanır, möhkəm təhlükəsizlik nəzarətinin həyata keçirilməsi və həssas məlumatların qorunması üçün yaxşı qurulmuş metodologiyalar təklif edir. Təhlükəsizlik pozuntuları ilə bağlı əsas risklərdən biri ciddi maliyyə nəticələri və hüquqi öhdəliklərlə nəticələnə bilən həssas məlumatların kompromisdir (David, 2019). Məlumatların şifrələnməsi və girişə nəzarət

mexanizmləri kimi təhlükəsiz dizayn nümunələri həssas məlumatlara icazəsiz girişə qarşı qoruyucu vasitə kimi xidmət edir. İstirahətdə və tranzitdə olan məlumatları şifrələməklə təşkilatlar zərərli aktyorların həssas məlumatları ələ keçirmək və ya manipulyasiya etmək cəhdlərinin qarşısını ala, bununla da məlumatların pozulması riskini və bundan sonra yaranan maliyyə nəticələrini azalda bilər. Təhlükəsizliyin pozulması nəticəsində yaranan maliyyə itkiləri təkcə tənzimləyici cərimələr və hüquqi hesablaşmalar kimi birbaşa xərcləri deyil, həm də bərpa səyləri və müştəri etibarının itirilməsi kimi dolayı xərcləri əhatə edən əhəmiyyətli ola bilər. Təhlükəsiz dizayn nümunələri uğurlu hücumlar ehtimalını azaltmaqla və potensial zərərin əhatə dairəsini məhdudlaşdırmaqla bu maliyyə risklərini minimuma endirməyə kömək edir. Ən az imtiyaz və dərindən müdafiə kimi müəyyən edilmiş təhlükəsizlik prinsiplərinə riayət etməklə təşkilatlar təhlükəsizlik təhdidlərinə qarşı dayanıqlığını gücləndirə və pozuntuların maliyyə təsirini azalda bilər. Üstəlik, təhlükəsizlik pozuntuları təşkilatın reputasiyasına uzunmüddətli ziyan vura bilər, müştərilər, tərəfdaşlar və maraqlı tərəflər arasında etibarı azalda bilər. Təhlükəsiz dizayn nümunələri proqram sistemlərinin təhlükəsizlik vəziyyətinə inam aşılamaqla təşkilati reputasiyanın qorunmasında mühüm rol oynayır. Təşkilatlar müdaxilənin aşkarlanması və insidentlərə cavab protokolları kimi təhlükəsizlik nəzarətlərini həyata keçirməklə təhlükəsizlik pozuntularını vaxtında aşkarlaya və azalda bilər, bununla da uzun müddət məlumatların ifşası və ya xidmətin kəsilməsi nəticəsində yaranan nüfuza zərəri minimuma endirirlər (Eloff, & Eloff, 2019). Xülasə, təhlükəsiz dizayn nümunələri təhlükəsizlik pozuntularının yaratdığı çoxşaxəli risklərin azaldılmasında alət rolunu oynayır. Təhlükəsizliyə nəzarəti həyata keçirmək və həssas məlumatı qorumaq üçün sübut edilmiş yanaşmalardan istifadə etməklə təşkilatlar təhlükəsizlik insidentlərinin ehtimalını və təsirini azalda, onların maliyyə sabitliyini və getdikcə daha düşmən rəqəmsal mühitdə nüfuzunu qoruya bilər.

Bugünkü tənzimləmə mənzərəsində məlumatların qorunması və məxfilik qaydalarına uyğunluq müxtəlif sənayelər üzrə təşkilatlar üçün çox vacibdir. GDPR (General Data Protection Regulation), HIPAA (Health Insurance Portability and Accountability Act) və PCI DSS (Payment Card Industry Data Security Standard) kimi

qaydalar həssas məlumatların qorunması və şəxslərin məxfilik hüquqlarının qorunması üçün ciddi tələblər qoyur. Təhlükəsiz dizayn nümunələri bu mürəkkəb tənzimləyici framevorkləri idarə etmək və uyğunluq məqsədlərinə nail olmaq istəyən təşkilatlar üçün əvəzedilməz alətlər kimi ortaya çıxır. Təhlükəsiz dizayn nümunələri inkişaf prosesinin əvvəlindən proqram arxitekturasının əsas strukturuna təhlükəsizlik tədbirlərini daxil etməklə normativ tələblərə uyğunluğu asanlaşdırır. Məsələn, GDPR təşkilatlara şəxsi məlumatların təhlükəsizliyini və məxfiliyini təmin etmək üçün müvafiq texniki və təşkilati tədbirlər həyata keçirməyi tapşırır. Məlumatların şifrələnməsi, psevdonimləşdirmə və giriş nəzarətləri kimi təhlükəsiz dizayn nümunələri şəxsi məlumatları icazəsiz giriş və sui-istifadədən qorumaq üçün güclü mexanizmlər təqdim etməklə təşkilatlara bu tələblərə cavab verməyə imkan verir. Eynilə, HIPAA səhiyyə təşkilatlarında qorunan sağlamlıq məlumatlarının (PHI) qorunması üçün ciddi təhlükəsizlik və məxfilik standartları tətbiq edir. RBAC, audit qeydi və təhlükəsiz rabitə protokolları kimi təhlükəsiz dizayn nümunələri həssas xəstə məlumatlarının məxfiliyini, bütövlüyünü və əlçatanlığını təmin etməklə səhiyyə təminatçılarına HIPAA tələblərinə əməl etməyə kömək edir. Bundan əlavə, ödəniş kartı məlumatlarını idarə edən təşkilatlar PCI DSS-nin ciddi tələblərinə tabedirlər ki, bu da kart sahibinin məlumatlarını qorumaq üçün hərtərəfli təhlükəsizlik nəzarətinin həyata keçirilməsini tələb edir. Tokenləşdirmə, təhlükəsiz autentifikasiya və təhlükəsiz kodlaşdırma təcrübələri kimi təhlükəsiz dizayn nümunələri ödəniş kartı məlumatlarının pozulması riskini azaltmaqla və kart sahibi məlumatlarının təhlükəsiz ötürülməsini və saxlanmasını təmin etməklə təşkilatlara PCI DSS uyğunluğuna nail olmaqda kömək edir (Menzel, Waidner, 2021). Təhlükəsiz dizayn nümunələrini proqram təminatının arxitekturasına inteqrasiya etməklə təşkilatlar uyğunluq səylərini sadələşdirə və cərimələr, hüquqi öhdəliklər və nüfuza zərər də daxil olmaqla, uyğunsuzluq cəzaları riskini azalda bilər. Bundan əlavə, təhlükəsiz dizayn nümunələrinin tətbiqi ilə təhlükəsizliyə proaktiv yanaşmanın mənimsənilməsi müştərilərin inamını və təşkilatın öz həssas məlumatlarını qorumaq öhdəliyinə inamını artırır və bununla da getdikcə daha çox tənzimlənən mühitdə uzunmüddətli əlaqələri və davamlı biznes artımını təşviq edir.

İnkişafdan sonra təhlükəsizlik zəifliklərinin aradan qaldırılması təşkilatlar üçün çətin və bahalı bir iş ola bilər. Yerləşdirmədən sonra təhlükəsizlik məsələlərinin həlli çox vaxt, resurslar və nüfuza ziyan baxımından əhəmiyyətli xərclərə səbəb olur. Təhlükəsiz dizayn nümunələri təhlükəsizlik mülahizələrini inkişaf prosesinə əvvəldən inteqrasiya etməklə bu xərcləri əhəmiyyətli dərəcədə azalda bilən təhlükəsizliyə proaktiv yanaşma təklif edir. Birincisi, inkişafın həyat dövrünün əvvəlində təhlükəsiz dizayn nümunələrinin tətbiqi ilk növbədə təhlükəsizlik insidentlərinin baş vermə ehtimalını azaldır. Tərtibatçılar dizayn və tətbiq mərhələlərində daxiletmənin yoxlanılması, autentifikasiya mexanizmləri və şifrələmə protokolları kimi ən yaxşı təhlükəsizlik təcrübələrini daxil etməklə, potensial zəifliklər təhlükəsizlik pozuntularına çevrilməzdən əvvəl qabaqlayıcı şəkildə həll edə bilərlər. Bu proaktiv yanaşma məlumatların pozulması, tənzimləyici cərimələr və hüquqi öhdəliklər daxil olmaqla təhlükəsizlik insidentləri riskini və onlarla əlaqəli xərcləri minimuma endirir. Bundan əlavə, təhlükəsiz dizayn nümunələri, inkişafdan sonrakı geniş təhlükəsizlik auditlərinə və remediasiya səylərinə ehtiyacı azaldaraq, təşkilatlara resursların bölüşdürülməsini optimallaşdırmağa kömək edir. Təhlükəsizliyə nəzarəti birbaşa proqram arxitekturasına daxil etməklə, təşkilatlar inkişaf prosesini sadələşdirə və resursları daha səmərəli şəkildə bölüşdürə bilər. Təhlükəsizliyə dair zəifliklər geriyə yönəltmək üçün əhəmiyyətli resurslar ayırmaq əvəzinə, təşkilatlar diqqətini məhsulun xüsusiyyətlərini təkmilləşdirməyə və müştərilərə dəyər təqdim etməyə, beləliklə, investisiyanın gəlirliliyini (ROI) maksimuma çatdıra və proqram təminatının həyat dövrü ərzində ümumi sahiblik xərclərini (TCO) minimuma endirə bilər. Bundan əlavə, təhlükəsiz dizayn nümunələrinin iqtisadi səmərəliliyi birbaşa maliyyə qənaətindən kənara çıxır (Harold, Micki, 2022). İnkişaf zamanı təhlükəsizlik risklərini fəal şəkildə həll etməklə, təşkilatlar öz brend reputasiyasını və müştəri etibarını qoruya, müştərinin itirilməsi və brend kapitalına ziyan kimi təhlükəsizlik insidentlərinin potensial uzunmüddətli nəticələrini azalda bilər. Bu qeyri-maddi fayda təşkilatın nəticələrinə və bazarda uzunmüddətli həyat qabiliyyətinə əhəmiyyətli təsir göstərə bilər. Yekun olaraq, inkişaf prosesinin əvvəlində təhlükəsiz dizayn nümunələrinin tətbiqi təşkilatlar üçün təhlükəsizlik zəifliklərini ilə bağlı riskləri və xərcləri azaltmaq üçün sərfəli

strategiya təklif edir. Təhlükəsizliyə dair problemləri qabaqcadan həll etməklə və proqram təminatının arxitekturasına güclü təhlükəsizlik nəzarətini inteqrasiya etməklə təşkilatlar təhlükəsizlik insidentlərinin baş vermə ehtimalını minimuma endirə və resursların bölüşdürülməsini optimallaşdıra, nəticədə səmərəliliyi və ROI-ni maksimuma çatdırmaqla öz proqram təminatının ümumi təhlükəsizlik vəziyyətini yaxşılaşdıra bilər.

Rəqəmsal dövrdə inam, bizneslər, müştərilər və maraqlı tərəflər arasında uğurlu əlaqələrin təməl daşını təşkil edir. Məlumatların məxfiliyi və təhlükəsizliyinin əsas narahatlıq doğurduğu bir-biri ilə əlaqəli dünyada inkişaf etmək istəyən bizneslər üçün etibarın yaradılması və saxlanması çox vacibdir. Təhlükəsiz dizayn nümunələri həssas məlumatları qorumaq və xidmətlərin bütövlüyünü və əlçatanlığını təmin etmək öhdəliyini nümayiş etdirməklə etimadın gücləndirilməsində mühüm rol oynayır. Müştərilər və maraqlı tərəflər bizneslərdən şəxsi və məxfi məlumatlarının qorunmasına üstünlük vermələrini gözləyirlər. Şifrələmə, giriş nəzarəti və təhlükəsiz autentifikasiya mexanizmləri kimi təhlükəsiz dizayn nümunələri müştərilərə onların məlumatlarına yüksək qayğı və hörmətlə yanaşıldığını bildirir. Bu nümunələri həyata keçirməklə təşkilatlar müştəriləri əmin edə bilərlər ki, onların məlumatı icazəsiz giriş, sui-istifadə və məlumatların pozulmasından qorunur və bununla da təşkilatın məxfilik və təhlükəsizliyə sadiqliyinə inam aşılaya bilər. Bundan əlavə, təhlükəsiz dizayn nümunələri xidmətlərin etibarlılığına və əlçatanlığına töhfə verir, ümumi istifadəçi təcrübəsini artırır, biznes və onların müştəriləri arasında etimadı gücləndirir. Təşkilatlar proqram təminatının arxitekturasına ehtiyat, əvəzetmə mexanizmləri və fəlakətin bərpası protokollarını daxil etməklə, müştərilərin xidmətlərə etibarlı və ardıcıl şəkildə daxil olmasını təmin edərək, xidmətin kəsilməsini və dayanma müddətini minimuma endirə bilər (Peter, 2020). Bu etibarlılıq, hətta gözlənilməz problemlər və ya pozulmalar qarşısında belə təşkilatın qüsursuz və fasiləsiz istifadəçi təcrübəsini təmin etməyə sadiqliyini nümayiş etdirməklə inamı artırır. Bundan əlavə, təhlükəsiz dizayn nümunələrinə riayət, markanın nüfuzuna və bazar rəqabət qabiliyyətinə daha geniş təsir göstərə bilər. Məlumatların pozulması və təhlükəsizlik insidentlərinin başlıqlarda üstünlük təşkil etdiyi bir dövrdə təhlükəsizlik və məxfiliyə üstünlük verən

bizneslər etibarlı tərəfdaşlar kimi seçilir. Təhlükəsiz dizayn nümunələrinə sərmayə qoymaq və təhlükəsizlik problemlərini fəal şəkildə həll etməklə, təşkilatlar özlərini rəqiblərdən fərqləndirə və məlumatların qorunması və kibertəhlükəsizlik sahəsində liderlər kimi yerləşdirə, bununla da öz brend reputasiyalarını artıra, yeni müştərilər və maraqlı tərəfləri cəlb edə bilərlər. Əslində, təhlükəsiz dizayn nümunələri bugünkü bir-biri ilə əlaqəli dünyada inam yaratmaq üçün əsas rol oynayır. Proqram təminatının dizaynında və tətbiqində təhlükəsizlik, məxfilik və etibarlılığa üstünlük verməklə təşkilatlar müştərilərə və maraqlı tərəflərə inam aşılaya, rəqiblərdən fərqlənə və etimad, dürüstlük və şəffaflıq üzərində qurulmuş uzunmüddətli əlaqələri gücləndirə bilər.

2.2. Veb tətbiqləri üçün ümumi təhdidlər

Veb proqramları ümumi zəifliklərdən tutmuş mürəkkəb hücumlara qədər çoxsaylı təhlükələrlə üzləşir və bu təhlükələr cədvəl 2.2-də verilmişdir.

Cədvəl 2.2

Veb tətbiqləri üçün ümumi təhdidlər

Saytlararası Skript (XSS)
SQL Injection (SQLi)
Saytlararası Sorğu Saxtakarlığı (CSRF)
Həssas Məlumatlara məruz qalma
Təhlükəsiz Birbaşa Obyekt İstinadları (IDOR)
Qırılmış Doğrulama
XML Xarici Müəssisə (XXE) Enjeksiyonu
Server Tərəfindən Sorğu Saxtakarlığı (SSRF)

Mənbə: Stuart McC., Joel S., George K, 2020, məlumatlarına əsasən müəllif tərəfindən hazırlanmışdır.

Saytlararası Skript (XSS) veb proqramlara təsir edən kritik təhlükəsizlik zəifliyidir. Bu, təcavüzkar digər istifadəçilər tərəfindən baxılan veb səhifələrə zərərli skriptlər yeritdikdə baş verir. Bu skriptlər qurbanların brauzerlərində icra edilir və təcavüzkarın həssas məlumatları oğurlamasına, səhifə məzmununu manipulyasiya etməyə və ya digər zərərli hərəkətlər etməyə imkan verir. XSS hücumlarının üç əsas növü var: əks olunan(Reflected), saxlanılan(Stored) və DOM əsaslı XSS.

1. Əks olunan XSS: Bu tip hücumda zərərli skript veb serverə göndərilən sorğuya daxil edilir və sonra həmin skripti cavabda istifadəçinin brauzerinə əks etdirir. Məsələn, təcavüzkar zərərli skripti ehtiva edən URL yarada bilər və istifadəçini ona kliklәməklə aldada bilər. İstifadəçi linki kliklədikdə, skript öz brauzerində icra olunur və bu, təcavüzkara onların sessiya kukilərini oğurlamağa və ya digər çirkin hərəkətlər etməyə imkan verir.

2. Saxlanan XSS: Davamlı XSS kimi də tanınır, bu tip hücum veb proqram verilənlər bazasına zərərli skriptlərin yeridilməsini nəzərdə tutur. Daha sonra yeridilmiş skript təhlükə altında olan səhifəyə daxil olan bütün istifadəçilərə təqdim olunur. Məsələn, təcavüzkar bir blog və ya sosial media platformasında şərhdə zərərli skript yerləşdirə bilər. Digər istifadəçilər şərhə baxdıqda, skript onların brauzerlərində icra olunur, potensial olaraq onların hesablarını təhlükə altına alır və ya məlumatlarını oğurlayır.

3. DOM əsaslı XSS: əks olunan və saxlanılan XSS-dən fərqli olaraq, DOM əsaslı XSS serveri əhatə etmir. Bunun əvəzinə hücum tamamilə müştəri tərəfi kodu (Sənəd Obyekt Modeli və ya DOM) daxilində baş verir. Təcavüzkar müştəri tərəfi skriptlərə zərərli skriptlər yeritməklə DOM mühitini manipulyasiya edir və sonradan qurbanın brauzeri tərəfindən icra olunur. Bu tip XSS-in aşkarlanması və qarşısının alınması xüsusilə çətindir, çünki o, müştəri tərəfində olan zəifliklərə əsaslanır.

XSS hücumlarını azaltmaq üçün veb tərtibatçıları bir neçə ən yaxşı təcrübəni, o cümlədən girişin doğrulanması və çıxışın kodlaşdırılmasını həyata keçirməlidirlər. Daxiletmənin yoxlanılması istifadəçi tərəfindən verilən məlumatların gözlənilən meyarlara cavab verməsini təmin edir, çıxış kodlaması isə skriptin icrasının qarşısını almaq üçün xüsusi simvollardan qaçır. Bundan əlavə, veb tətbiqi təhlükəsizlik divarları

(WAF) və CSP kimi təhlükəsizlik başlıqları zərərli yükləri aşkarlamaq və bloklamaqla XSS hücumlarına qarşı əlavə qorunma təmin edə bilər. XSS zəifliklərini təcavüzkarlar tərəfindən istismar edilməzdən əvvəl müəyyən etmək və həll etmək üçün müntəzəm təhlükəsizlik auditləri və nüfuz testi də vacibdir (Barnes, 2015).

SQL Injection (SQLi): SQLi verilənlər bazasına əsaslanan veb proqramları hədəf alan ciddi təhlükəsizlik zəifliyidir. Bu, təcavüzkarlara verilənlər bazası ilə qarşılıqlı əlaqə yaratmaq üçün tətbiq tərəfindən istifadə edilən SQL sorğularını manipulyasiya etməyə imkan verir. SQLi-dən istifadə etməklə təcavüzkarlar həssas məlumatlara icazəsiz giriş əldə edə, qeydləri dəyişdirə və ya silə, hətta bütün verilənlər bazası serverinə nəzarət edə bilərlər. SQL inyeksiya hücumlarının bir neçə növü var:

1.Classic SQL Injection: Klassik SQLi hücumlarında təcavüzkarlar tətbiqin verilənlər bazası sorğusuna zərərli SQL kodu daxil etmək üçün giriş formaları və ya axtarış çubuqları kimi daxiletmə sahələrini manipulyasiya edirlər. Məsələn, təcavüzkar sorğunun məntiqini dəyişdirən xüsusi hazırlanmış istifadəçi adı və ya parol daxil edə bilər ki, bu da onlara autentifikasiyadan keçməyə və ya həssas məlumatları əldə etməyə imkan verir.

2.Blind SQL Injection: Kor SQLi hücumlarında təcavüzkarlar öz zərərli sorğularının nəticələrini birbaşa almadan zəifliklərdən istifadə edirlər. Bunun əvəzinə, dolayı yolla məlumatları çıxarmaq üçün tətbiqin davranışına etibar edirlər. Məsələn, təcavüzkarlar faktiki nəticələri görmədən məlumatları çıxarmaq və ya verilənlər bazasının strukturunu müəyyən etmək üçün boolean əsaslı və ya vaxta əsaslanan üsullardan istifadə edə bilərlər.

3.Secondary SQL Injection: İkinci dərəcəli SQLi, yeridilmiş zərərli kod verilənlər bazasında saxlandıqda və daha sonra başqa istifadəçi tərəfindən baxıldıqda və ya icra edildikdə baş verir. Bu tip hücumu aşkar etmək və azaltmaq daha çətindir, çünki inyeksiya yükü işə salınana qədər hərəkətsiz qalır (Bill, 2015).

SQL inyeksiya təhlükələrini azaltmaq üçün tərtibatçılar ən yaxşı təcrübələrə əməl etməlidirlər, məsələn:

-Parametrləşdirilmiş Sorğular: Parametrləşdirilmiş sorğulardan və ya hazırlanmış ifadələrdən istifadə SQL kodunu istifadəçi daxiletməsindən ayırmağa kömək edir, təcavüzkarların sorğunun strukturunu dəyişməsinin qarşısını alır.

-Girişin Təsdiqlənməsi və Sanitizasiyası: İstifadəçi daxiletməsinin yoxlanılması və sanitarlaşdırılması SQLi zəifliklərinin ehtimalını azaldaraq, yalnız gözlənilən məlumat formatlarının qəbul edilməsini təmin etməyə kömək edə bilər.

-Ən Az İmtiyaz Prinsip: Hər bir proqram istifadəçisi üçün verilənlər bazası icazələrinin minimum tələb olunan səviyyəyə məhdudlaşdırılması uğurlu SQLi hücumlarının potensial təsirini məhdudlaşdıra bilər.

-Daimi Təhlükəsizlik Auditləri: Müntəzəm təhlükəsizlik auditlərinin və nüfuz testlərinin aparılması SQL inyeksiya zəifliklərini zərərli şəxslər tərəfindən istismar edilməzdən əvvəl müəyyən etməyə və aradan qaldırmağa kömək edə bilər.

Bu tədbirləri həyata keçirməklə veb tərtibatçıları SQL inyeksiya hücumları riskini əhəmiyyətli dərəcədə azalda və öz proqramlarını məlumatların pozulması və digər təhlükəsizlik təhdidlərindən qoruya bilərlər.

Saytlararası Sorğu Saxtakarlığı (CSRF) veb proqramların təhlükəsizliyinə əhəmiyyətli təhlükə yaradır və təcavüzkarlara təsdiqlənmiş istifadəçilər adından icazəsiz hərəkətlər etməyə imkan verir. Bu hücum zərərli veb-sayt və ya e-poçt istifadəçini bilmədən onun kimliyi təsdiqləndiyi həssas veb proqrama sorğu göndərmək üçün aldatdığı zaman baş verir. CSRF hücumlarının əsasını təşkil edən əsas problem veb proqramlarda sorğunun mənbəyinin yoxlanılmasıdır. Veb sayta zərərli skriptlərin yeridilməsinə yönəlmiş XSS fərqli olaraq, CSRF veb tətbiqinin istifadəçinin brauzerinə verdiyi etibardan istifadə edir. Bu etibardan istifadə etməklə, təcavüzkarlar veb tətbiqi üçün qanuni görünən sorğuları saxtalaşdıra və bu, istifadəçinin razılığı olmadan əməliyyatların icrasına səbəb ola bilər. Ümumi ssenarilərdən biri təcavüzkarın zərərli URL-i hazırlaması və ya onu ayrı bir saytda yerləşdirilən şəkil və ya skript daxilində yerləşdirməsini əhatə edir (Bennett, 2011). Hədəf veb tətbiqinə autentifikasiya edilmiş qurban zərərli məzmuna daxil olduqda, onların brauzeri avtomatik olaraq həssas proqrama saxta sorğu göndərir. Sorğu etibarlı autentifikasiya etimadnaməsinə malik qurbanın brauzerindən gəldiyi üçün proqram onu qanuni olaraq

emal edir, nəticədə vəsait köçürmələri, profil dəyişiklikləri və ya məlumatların silinməsi kimi tədbirlər görülür. CSRF hücumlarını azaltmaq üçün veb tərtibatçıları bir neçə üsuldan istifadə edirlər:

1.CSRF Tokens: Hər bir istifadəçi sessiyası üçün unikal tokenlər yaratmaq və onları forma və ya sorğulara daxil etmək. Bu tokenlər server tərəfindən hər sorğu üçün təsdiqlənir və yalnız qanuni sorğuların işlənməsini təmin edir.

2.Same-Site Cookies: kukilərin eyni mənşəli istifadəsini məhdudlaşdırmaq üçün SameSite atributunun qurulması və bununla da onların mənşəli sorğularda göndərilməsinin qarşısını almaq.

3.Yönləndirici Siyasəti: İcazəsiz domenlərin həssas məlumatlara daxil olmasının qarşısını alaraq, CSRF hücumları riskini azaltmaqla HTTP başlıqlarında açıqlanan məlumatı məhdudlaşdırmaq üçün ciddi yönləndirici siyasətlərdən istifadə etmək.

4.Xüsusi Başlıqlar: Sorğulara xüsusi başlıqların daxil edilməsi və sorğuların etibarlı mənbələrdən olmasını təmin etmək üçün onların server tərəfində təsdiqlənməsi.

5.İkiqat kukilər göndərmək: CSRF hücumlarını azaltmaq üçün server onların uyğun olduğunu təsdiq etməklə həm kukidə, həm də sorğu parametri kimi təsadüfi nişanın göndərilməsi.

Bu yumşaldıcı tədbirlərə baxmayaraq, tərtibatçılar sayıq qalmalı və potensial zəifliklər üçün tətbiqlərini müntəzəm olaraq yoxlamalıdırlar. Bundan əlavə, istifadəçi təhsili həssas tətbiqlərə autentifikasiya edilərkən şübhəli keçidlərə klikləməyin və ya etibarsız vebsaytlara daxil olmağın təhlükələri haqqında məlumatlılığı artırmaqla CSRF hücumlarının qarşısının alınmasında mühüm rol oynayır. Texniki nəzarət və istifadəçi məlumatlılığının kombinasiyasını həyata keçirməklə təşkilatlar öz veb tətbiqləri üçün CSRF təhlükəsini effektiv şəkildə azalda bilərlər.

Veb proqramlarında həssas məlumatların ifşa edilməsi həm istifadəçilər, həm də təşkilatlar üçün əhəmiyyətli təhlükə yaradır və potensial olaraq şəxsiyyət oğurluğu, maliyyə itkisi, nüfuza zərər və hüquqi öhdəliklər kimi ağır nəticələrə gətirib çıxarır. Bu təhlükə şəxsi identifikasiya edilə bilən məlumat (PII), maliyyə məlumatları və ya həssas məlumatların icazəsiz şəxslərə təsadüfən və ya zərərli şəkildə açıqlandığı müxtəlif ssenariləri əhatə edir. Həssas məlumatları ifşa etmək üçün ümumi vektor zəif

autentifikasiya mexanizmləri, qeyri-kafi giriş nəzarəti və həssas kod daxil olmaqla, qeyri-adekvat təhlükəsizlik tədbirləridir (Mark, 2020). Məsələn, bir veb tətbiqi tranzit və ya istirahət zamanı həssas məlumatları şifrələyə bilmirsə, təcavüzkarlar rabitəni kəsə və ya verilənlər bazasına icazəsiz giriş əldə edə bilər və bu, məlumatların pozulmasına səbəb ola bilər. Bundan əlavə, SQL inyeksiya, XSS və təhlükəli birbaşa obyekt istinadları (IDOR) kimi boşluqlar təcavüzkarlara sorğuları manipulyasiya etməyə və ya zərərli skriptləri yeritməyə imkan verə bilər, bununla da proqramda və ya onun arxa sistemlərində saxlanılan həssas məlumatlara daxil olur. Bundan əlavə, yanlış konfiqurasiya edilmiş bulud yaddaşı və ya düzgün qurulmamış icazələr həssas məlumatların internetə təsadüfən məruz qalması ilə nəticələnə bilər. Bundan əlavə, veb proqramlar tərəfindən istifadə edilən üçüncü tərəf asılılıqları və API-lər əlavə risklər yaradır. Bu asılılıqlar təhlükəsizlik zəiflikləri üçün hərtərəfli yoxlanılmazsa və ya düzgün autentifikasiya və avtorizasiya mexanizmlərini tətbiq edə bilmirsə, təcavüzkarlar tətbiq tərəfindən emal edilən və ya saxlanılan həssas məlumatlara giriş əldə etmək üçün onlardan istifadə edə bilərlər. Həssas məlumatların ifşası təhlükəsini azaltmaq üçün veb tərtibatçıları və təşkilatları hərtərəfli təhlükəsizlik strategiyası həyata keçirməlidirlər:

1.Məlumatların Şifrələnməsi: Həssas məlumatları qorumaq üçün şifrələmə alqoritmlərindən istifadə edərək, ələ keçirilsə belə, məlumatların icazəsiz şəxslər üçün oxunmaz qalmasını təmin etmək.

2.Giriş Nəzarətləri: Həssas məlumatlara girişi yalnız səlahiyyətli istifadəçilər və ya rollarla məhdudlaşdırmaq üçün güclü giriş nəzarətlərini və autentifikasiya mexanizmlərini tətbiq etmək.

3.Həssaslığın İdarə Edilməsi: Dəyişikliklər və ya yeniləmələr vasitəsilə müəyyən edilmiş zəif cəhətləri operativ şəkildə aradan qaldıraraq, təhlükəsizlik zəiflikləri üçün veb proqramları müntəzəm olaraq skan etmək və yoxlamaq.

4.Təhlükəsiz Kodlaşdırma Təcrübələri: SQL inyeksiya, XSS və IDOR kimi ümumi zəifliklərin qarşısını almaq üçün təhlükəsiz kodlaşdırma qaydalarına riayət etmək, istifadə oluna bilən zəifliklər səbəbindən məlumatlarda tamlığın pozulması riskini azaltmaq.

5.Üçüncü Tərəf Risk İdarəetmə: Üçüncü tərəf asılılıqlarının və API-lərin təhlükəsizlik vəziyyətini hərtərəfli qiymətləndirərək, onların təhlükəsizlik üzrə ən yaxşı təcrübə və standartlara uyğunluğunu təmin etmək (Charles, Susan, 2021).

Təhlükəsizliyə proaktiv yanaşmanı mənimsəmək və müvafiq təminatları həyata keçirməklə təşkilatlar həm istifadəçilərini, həm də reputasiyalarını qoruyaraq, veb proqramlarında həssas məlumatların ifşası təhlükəsini effektiv şəkildə azalda bilər.

IDOR (Insecure Direct Object References) veb proqramların təhlükəsizliyinə əhəmiyyətli təhlükə yaradır. Bu boşluq, tətbiqin kodu istifadəçinin obyektlərə və ya resurslara girişini düzgün şəkildə təsdiq edə bilmədikdə yaranır və bu, bədniyyətlilərə icazəsiz məlumatlara daxil olmaq üçün girişi manipulyasiya etməyə imkan verir. İdentifikasiya mexanizmlərində və ya daxiletmənin təsdiqində qüsurlardan istifadəni əhatə edən təhlükəsizlik zəifliklərinin digər növlərindən fərqli olaraq, IDOR hücumları proqram daxilində obyekt istinadlarının birbaşa manipulyasiyasına diqqət yetirir. IDOR zəifliyinin mahiyyəti istifadəçi tərəfindən təmin edilən giriş və daxili obyekt istinadları arasında düzgün olmayan xəritələşmədədir. Tipik olaraq, veb proqramları verilənlər bazası açarları və ya fayl yolları kimi istifadəçinin əldə edə biləcəyi resurslar və daxili identifikatorlar arasında xəritəni saxlayır. Bu xəritələşdirmə adekvat şəkildə qorunmadıqda və ya təsdiqlənmədikdə, bədniyyətlilər sorğular və ya URL-lərdə obyekt istinadlarına müdaxilə edərək zəiflikdən istifadə edə bilərlər. IDOR zəifliklərinin baş verdiyi ümumi ssenarilərdən biri giriş nəzarətlərini tətbiq etmək üçün yalnız müştəri tərəfi yoxlamasına əsaslanan sistemlərdədir. Belə hallarda, bədniyyətlilər istənilən müştəri tərəfi məhdudiyyətlərini effektiv şəkildə keçərək və həssas məlumatlara icazəsiz giriş əldə edərək obyekt istinadlarını manipulyasiya etmək üçün sorğuları ələ keçirə və dəyişdirə bilərlər (Chapple, Seidl, 2020). Bədniyyətlilər proqram daxilində saxlanılan həssas məlumatlara potensial olaraq daxil ola, dəyişdirə və ya silə bildiyi üçün IDOR zəifliyinin təsiri ciddi ola bilər. Bura şəxsi istifadəçi məlumatları, maliyyə qeydləri, əqli mülkiyyət və proqram tərəfindən idarə olunan hər hansı digər həssas resurslar daxildir. Bundan əlavə, IDOR zəifliklər imtiyazların artırılması hücumları üçün də istifadə edilə bilər ki, bu da bədniyyətlilərə sistem daxilində yüksək icazələr əldə etməyə imkan verir. IDOR zəifliklərinin azaldılması

təhlükəsizliyə çox səviyyəli yanaşma tələb edir. İlk növbədə, tərtibatçılar proqram kodu daxilində müvafiq giriş nəzarəti və avtorizasiya mexanizmlərini tətbiq etməlidirlər. Buraya istifadəçi icazələrinin təsdiqlənməsi və həssas resurslara girişin təsdiqlənmiş istifadəçi identifikasiyası və rolları əsasında ciddi şəkildə tətbiq olunmasının təmin edilməsi daxildir. Əlavə olaraq, tərtibatçılar istifadəçi tərəfindən təmin edilən giriş və daxili obyekt istinadları arasında xəritələşdirməni abstraktlaşdırmaq üçün dolayı obyekt istinadları və ya girişə nəzarət siyahıları (ACL) kimi üsullardan istifadə etməlidirlər. İstifadəçi tərəfindən görünən identifikatorları daxili tətbiq detallarından ayırmaqla tərtibatçılar IDOR zəiflikləri riskini azalda bilər. Avtomatlaşdırılmış zəifliyin skan edilməsi və əl ilə kodun nəzərdən keçirilməsi daxil olmaqla, müntəzəm təhlükəsizlik testi veb tətbiqlərində IDOR zəifliklərinin müəyyən edilməsi və aradan qaldırılması üçün vacibdir. Təşkilatlar bu zəiflikləri fəal şəkildə müəyyən etməklə və aradan qaldırmaqla öz veb tətbiqlərində təhlükəsizlik vəziyyətini əhəmiyyətli dərəcədə artıra və həssas məlumatlara icazəsiz girişdən qoruya bilər.

Qırılmış autentifikasiya, autentifikasiya və sessiya idarəetmə mexanizmlərinin həyata keçirilməsində zəif cəhətlərdən irəli gələn veb tətbiqlərin təhlükəsizliyinə kritik təhlükə yaradır. Bu boşluq bədniyyətlilərə istifadəçi hesablarını ələ keçirməyə, həssas məlumatlara icazəsiz giriş əldə etməyə və proqram daxilində müxtəlif zərərli fəaliyyətlər həyata keçirməyə imkan verə bilər. Əsas etibarilə pozulmuş autentifikasiya istifadəçi girişi, sessiyanın idarə edilməsi, parolun idarə edilməsi və digər əlaqəli funksiyaları əhatə edən autentifikasiya prosesindəki qüsurlardan yaranır. Qırılmış autentifikasiyaya töhfə verən ümumi zəif cəhətlərə kifayət qədər təhlükəsiz etimadnamələrin saxlanması, qeyri-adekvat sessiyanın başa çatması və idarə olunması, proqnozlaşdırıla bilən sessiya nişanları və zəif parol siyasətləri daxildir. Qırılmış autentifikasiyaya səbəb olan geniş yayılmış problemlərdən biri parollar və ya autentifikasiya nişanları kimi istifadəçi etimadnamələrinin təhlükəsiz saxlanmamasıdır. Həssas məlumatlar düz mətndə saxlandıqda və ya zəif şifrələmə üsullarından istifadə edildikdə, bədniyyətlilər qanuni istifadəçiləri təqlid etmək və onların hesablarına icazəsiz giriş əldə etmək üçün bu etimadnamələri asanlıqla əldə edə və istifadə edə bilərlər (Qıllıcam, 2023). Bundan əlavə, düzgün olmayan sessiya

idarəetmə təcrübələri də veb proqramlarında zəifliklər yarada bilər. Məsələn, çıxış, fəaliyyətsizlik və ya hesab dəyişiklikləri zamanı istifadəçi seanslarını lazımi qaydada etibarsız hesab etməmək və ya müddətini başa çatdırmaq sessiyaları sessiyanın bağlanması, sessiyanın oğurlanması və ya sessiyanın təkrar oxunması hücumlarına qarşı həssas edə bilər. Bədniyyətlilər istifadəçi hesablarına davamlı giriş əldə etmək və ya onların xəbəri olmadan autentifikasiya edilmiş istifadəçiləri təqlid etmək üçün bu zəif cəhətlərdən istifadə edə bilərlər. Qırılmış autentifikasiyanın digər ümumi cəhəti zəif parol siyasətləridir, məsələn, istifadəçilərə asanlıqla təxmin edilə bilən parollar yaratmağa icazə vermək və ya parol mürəkkəbliyi tələblərini yerinə yetirməmək. Zəif parollar çox sayda hücumlara məruz qalır, burada bədniyyətlilər düzgün parolu tapana qədər sistematik olaraq müxtəlif kombinasiyaları sınayır və onlara istifadəçi hesablarına icazəsiz giriş imkanı verir. Qırılmış autentifikasiya təhlükəsini azaltmaq üçün autentifikasiya və sessiyanın idarə edilməsi üzrə ən yaxşı təcrübələrin möhkəm tətbiqi tələb olunur. Bura güclü kriptoqrafik həşinq(hashing) alqoritmlərindən istifadə edərək istifadəçi etimadnamələrinin təhlükəsiz saxlanması, əlavə təhlükəsizlik qatı əlavə etmək üçün çox faktorlu autentifikasiyanın (MFA) həyata keçirilməsi və sessiyanın bitməsi, sessiya işarələrinin fırlanması və təhlükəsiz kuki atributları kimi düzgün sessiyanın idarə edilməsi təcrübələrinin tətbiqi daxildir. Müntəzəm təhlükəsizlik qiymətləndirmələri, o cümlədən penetrasiya testi və zəifliyin skan edilməsi, veb tətbiqlərində pozulmuş autentifikasiya ilə bağlı zəifliklərin müəyyən edilməsi və aradan qaldırılması üçün vacibdir. Bundan əlavə, autentifikasiya ilə əlaqəli jurnalların və istifadəçi fəaliyyətinin davamlı monitorinqi şübhəli davranışı və icazəsiz giriş cəhdlərini aşkar etməyə kömək edə bilər, təşkilatlara sistemlərini və istifadəçi məlumatlarını istismardan qorumaq üçün vaxtında tədbirlər görməyə imkan verir.

XXE inyeksiyası XML girişini təhlil edən veb proqramların təhlükəsizliyinə əhəmiyyətli təhlükə yaradır. Bu boşluq, proqram düzgün təsdiqlənmədən etibarsız mənbələrdən XML daxiletməsini emal etdikdə, bədniyyətlilərə həssas məlumatlara daxil olmaq, ixtiyari kod icra etmək və müxtəlif zərərli fəaliyyətlər həyata keçirmək üçün XML təhlili funksiyasından istifadə etməyə imkan verən zaman yaranır. XXE inyeksiyasının mahiyyəti bədniyyətlilərin tətbiq tərəfindən işlənmiş XML sənədlərinə

xarici obyektlər və ya istinadlar daxil etmək imkanıdır (Whitman, Mattord, 2019). Bu qurumlar yerli faylları oxumaq, xarici serverlərə şəbəkə sorğularını başlamaq və hətta əsas serverdə sistem əmrlərini yerinə yetirmək də daxil olmaqla müxtəlif hücumları həyata keçirmək üçün istifadə edilə bilər. XXE inyeksiya zəifliklərinin baş verdiyi ümumi ssenarilərdən biri SOAP və ya REST API-ləri, XML əsaslı fayl yükləmələri və ya müştəri ilə server arasında mübadilə edilən XML verilənləri kimi istifadəçilərdən XML əsaslı daxiletmələri qəbul edən veb proqramlardadır. Tətbiq XML daxiletməsini düzgün şəkildə təsdiq edə və təmizləyə bilmirsə, bədniyyətlilər XML təhlilçisinin funksionallığından istifadə edən xarici obyekt istinadları olan zərərli XML yükləri yarada bilərlər. XXE inyeksiya zəifliyinin təsiri şiddətli ola bilər və bədniyyətlilərə serverdə saxlanılan konfiqurasiya faylları, etimadnamələr və ya digər sistem resursları kimi həssas məlumatlara daxil olmağa imkan verir. Əlavə olaraq, XXE inyeksiyası serverdən əldə edilə bilən digər sistemlərə və ya xidmətlərə qarşı hücumların həyata keçirilməsi üçün istifadə edilə bilər ki, bu da potensial olaraq tətbiqin və ya əsas infrastrukturun daha da pozulmasına gətirib çıxarır. XXE inyeksiya təhlükəsinin azaldılması proqram yığınının müxtəlif səviyyələrində bir neçə təhlükəsizlik tədbirinin həyata keçirilməsini tələb edir. İlk növbədə, tərtibatçılar XML daxiletməsini idarə edərkən təhlükəsiz kodlaşdırma təcrübələrini qəbul etməlidirlər, o cümlədən xarici obyektlərin daxil edilməsinin qarşısını almaq üçün istifadəçi tərəfindən təmin edilmiş XML məlumatlarını yoxlamaq və təmizləyərək uyğunlaşdırmaq lazımdır. Bundan əlavə, XML təhlilçilərini xarici obyektlər üçün dəstəyi söndürmək və ya onların istifadəsini məhdudlaşdırmaq üçün konfiqurasiya etmək XXE inyeksiya riskini azaltmağa kömək edə bilər (Darril, 2020). Buna DTD(Document Type Definition) emalını söndürmək və ya XXE zəifliklərini təbii olaraq azaldan təhlükəsiz XML emal kitabxanalarından istifadə etmək kimi müvafiq təhlilçi konfiqurasiyaları qurmaqla nail olmaq olar. Zəifliyin skan edilməsi və penetrasiya testi daxil olmaqla, müntəzəm təhlükəsizlik testi veb tətbiqlərində XXE inyeksiya zəifliklərinin müəyyən edilməsi və aradan qaldırılması üçün vacibdir. Avtomatlaşdırılmış alətlər ümumi XXE nümunələrini və potensial inyeksiya nöqtələrini aşkar etmək üçün istifadə oluna bilər, əllə sınaq isə daha mürəkkəb hücum vektorlarını və kənar halları aşkar edə bilər. Bu

təhlükəsizlik tədbirlərini həyata keçirməklə və zəifliyin idarə edilməsinə proaktiv yanaşma tətbiq etməklə, təşkilatlar XXE inyeksiya təhlükəsini effektiv şəkildə azalda və öz veb proqramlarını zərərli aktorların istismarından qoruya bilər.

SSRF veb proqramların təhlükəsizliyinə əhəmiyyətli təhlükə yaradır və bədniyyətlilərə proqram adından digər daxili və ya xarici sistemlərə sorğular etmək üçün serveri manipulyasiya etməyə imkan verir. Bu boşluq, proqram istifadəçi tərəfindən təmin edilən daxiletməni qəbul etdikdə və ondan lazımi yoxlama və ya təmizlənmədən digər serverlərə və ya xidmətlərə sorğu yaratmaq üçün istifadə etdikdə yaranır. Bədniyyətlilər SSRF boşluqlarından müxtəlif zərərli fəaliyyətlər həyata keçirmək üçün istifadə edə bilər, o cümlədən həssas məlumatlara daxil olmaq, təhlükəsizlik duvarı məhdudiyyətlərini keçmək, icazəsiz hərəkətlərə başlamaq və daxili şəbəkə infrastrukturunda kəşfiyyat aparmaq. SSRF boşluqlarının baş verdiyi ümumi ssenarilərdən biri, uzaq serverlərdən məlumatların alınması, API sorğularının olunması və ya istifadəçi tərəfindən təmin edilmiş URL-lərin işlənməsi kimi xarici resurslarla qarşılıqlı əlaqədə olan veb proqramlardadır. Tətbiq istifadəçi tərəfindən təmin edilən girişi adekvat şəkildə təsdiq edə və ya məhdudlaşdıra bilmədikdə, bədniyyətlilər daxili resurslara daxil olmaq və ya xarici sistemlərdə hərəkətlər etmək üçün serverin etibarından sui-istifadə edən zərərli sorğular hazırlaya bilərlər. SSRF zəifliyinin təsiri təhlükəyə məruz qalmış serverin imkanlarından və daxili şəbəkənin təhlükəsizlik vəziyyətindən asılı olaraq ciddi ola bilər. Hücumçular həssas məlumatlara daxil olmaq, imtiyazları artırmaq, şəbəkə daxilində digər sistemlərə keçmək və ya hətta təhlükəyə məruz qalmış server adından xarici xidmətlərə qarşı hücumlar etmək üçün SSRF-dən istifadə edə bilərlər. SSRF təhlükəsinin azaldılması proqram yığınının müxtəlif təbəqələrində bir neçə təhlükəsizlik tədbirinin həyata keçirilməsini tələb edir. İlk növbədə, tərtibatçılar, xüsusən də xarici resurslara sorğular hazırlayarkən istifadəçi tərəfindən təmin edilən bütün daxiletmələri təsdiq etməli və təmizləməlidir. Buraya məqbul URL və ya IP ünvanlarının ağ siyahılarının və ya icazəli siyahılarının tətbiqi və potensial həssas və ya daxili resurslara sorğuların bloklanması daxildir (Ambrosıanı, 2015). Əlavə olaraq, şəbəkə seqmentasiyası və fayrvol qaydaları serverin daxili sistemlər və xarici xidmətlərlə əlaqə saxlamaq qabiliyyətini məhdudlaşdırmağa kömək

edə bilər, SSRF zəiflikləri üçün hücum səthini azalda bilər. Zərərli SSRF sorğularını real vaxt rejimində aşkar etmək və bloklamaq üçün veb tətbiqi təhlükəsizlik divarları (WAFs) və müdaxilənin aşkarlanması/qarşısının alınması sistemləri (IDS/IPS) də istifadə edilə bilər. Zəifliyin skan edilməsi və penetrasiya testi daxil olmaqla, müntəzəm təhlükəsizlik testi veb tətbiqlərində SSRF zəifliklərinin müəyyən edilməsi və aradan qaldırılması üçün vacibdir. Təhlükəsizliyə proaktiv yanaşmanı mənimsəməklə və SSRF hücumlarına qarşı möhkəm müdafiəni həyata keçirməklə təşkilatlar istismar riskini azalda, tətbiqlərini və əsas infrastrukturunu icazəsiz giriş və sui-istifadədən qoruya bilər.

2.3. Veb tətbiqlərin təhlükəsizliyin aşkarlanmasında müasir texnologiyalar

Veb tətbiqi təhlükəsizliyinin aşkarlanması sahəsi daim inkişaf edir, yeni texnologiyalar və getdikcə artan təhdidləri aradan qaldırmaq üçün ortaya çıxan yanaşmalar buna misaldır. Hazırda mövcud olan ən qabaqcıl texnologiyalar cədvəl 2.3-də verilmişdir.

Cədvəl 2.3

Veb tətbiqlərin təhlükəsizliyin aşkarlanmasında müasir texnologiyalar

Maşın Öyrənməsi (ML) və Süni İntellekt (AI)
Davranış Analizi
Statik Tətbiq Təhlükəsizliyi Testi (SAST)
Dinamik Tətbiq Təhlükəsizlik Testi (DAST)
İşləmə Zamanı Tətbiqin Özünü Mühaafizəsi (RASP)
Konteyner Təhlükəsizliyi
Blokçeyn Texnologiyası

Mənbə: Thomas P.C, 2020, məlumatlarına əsasən müəllif tərəfindən hazırlanmışdır.

Maşın Öyrənməsi (ML) və Süni İntellekt (AI) anomaliyaların aşkarlanması, nümunənin tanınması və davranış analizi üçün qabaqcıl imkanlar təklif edərək veb proqram təhlükəsizliyinin aşkarlanmasında inqilab edir. Bu texnologiyalar real vaxt rejimində təhlükəsizlik təhdidlərini müəyyən etmək və onlara cavab vermək üçün böyük həcmdə məlumatlardan istifadə edərək veb proqramların ümumi təhlükəsizlik vəziyyətini əhəmiyyətli dərəcədə artırır. Veb tətbiqi təhlükəsizliyində ML və AI-nin əsas tətbiqlərindən biri anomaliyaların aşkarlanmasıdır (Atalay, 2016). ML alqoritmləri giriş nümunələri, istifadəçi qarşılıqlı əlaqələri və sistem fəaliyyətləri kimi tarixi məlumatları təhlil edərək veb proqramların və onların istifadəçilərinin normal davranışını öyrənə bilər. Normal davranışdan sapmalar baş verdikdə, bu alqoritmlər onları potensial təhlükəsizlik təhdidləri kimi qeyd edə bilər. Məsələn, trafikdə qeyri-adi sıçrayışlar, gözlənilməz istifadəçi davranışları və ya anormal sistem fəaliyyətləri təhlükəsizlik pozuntusunun göstəriciləri ola bilər. Nümunənin tanınması veb tətbiqi təhlükəsizliyində ML və AI-nin digər mühüm aspektidir. Bu texnologiyalar SQL inyeksiya, XSS və ya paylanmış xidmətdən imtina (DDoS) hücumları kimi məlum təhlükəsizlik təhdidləri ilə əlaqəli təkrarlanan nümunələri müəyyən etmək üçün geniş məlumat dəstlərini təhlil edə bilər. Bu nümunələri tanımaqla, ML və AI sistemləri oxşar təhdidləri tətbiqə və ya onun istifadəçilərinə əhəmiyyətli zərər vurmazdan əvvəl aktiv şəkildə aşkarlaya və azalda bilər. Bundan əlavə, ML və AI veb tətbiqi təhlükəsizliyində davranış analizinə imkan verir. İstifadəçi qarşılıqlı əlaqəsini və sistem fəaliyyətlərini davamlı olaraq izləməklə, bu texnologiyalar qanuni istifadəçilər üçün ilkin davranışlar qura və zərərli niyyəti göstərə biləcək kənarlaşmaları aşkar edə bilər. Məsələn, ML alqoritmləri real vaxt rejimində istifadəçi davranışını təhlil edərək şübhəli giriş cəhdlərini, qeyri-adi giriş nümunələrini və ya icazəsiz məlumatların çıxarılmasını aşkar edə bilər. Veb tətbiqi təhlükəsizliyində ML və AI-nin əsas üstünlüklərindən biri onların zamanla uyğunlaşma və təkamül etmək qabiliyyətidir (Charles, Susan, 2022). Bəd niyyətlilər yeni texnika və taktikalar inkişaf etdirdikcə, ML və AI sistemləri keçmiş hadisələrdən öyrənə və yaranan təhlükələri effektiv şəkildə aşkar etmək üçün alqoritmlərini yeniləyə bilər. Bu uyğunlaşma qabiliyyəti veb proqramların inkişaf edən təhlükəsizlik sorğularına qarşı davamlı olmasını təmin edir.

Bundan əlavə, ML və AI təhlükəsizlik insidentlərinə cavab tədbirlərinin səmərəliliyini və dəqiqliyini artıra bilər. Təhlükəsizlik xəbərdarlıqlarının təhlilini avtomatlaşdıraraq və risk səviyyələrinə əsasən prioritetləşdirməklə bu texnologiyalar təhlükəsizlik qruplarına səylərini ən kritik təhlükələrə yönəltməyə imkan verir. Əlavə olaraq, ML-yə əsaslanan proqnozlaşdırıcı analitika tarixi məlumatlar əsasında potensial təhlükəsizlik təhdidlərini proqnozlaşdıra bilər ki, bu da təşkilatlara qabaqlayıcı tədbirləri fəal şəkildə həyata keçirməyə imkan verir. Nəticə olaraq, maşın öyrənmə və süni intellekt veb proqramların təhlükəsizliyinin aşkarlanması imkanlarının artırılmasında mühüm rol oynayır. Anomaliyaların aşkarlanması, nümunənin tanınması və davranış analizi üçün qabaqcıl alqoritmlərdən istifadə etməklə, bu texnologiyalar təşkilatlara real vaxt rejimində təhlükəsizlik təhdidlərini effektiv şəkildə müəyyən etmək və onlara cavab vermək imkanı verir və bununla da öz veb proqramlarını və istifadəçilərini zərərdən qoruyur.

Davranış analizi, tətbiq daxilində istifadəçilər və sistemlər tərəfindən nümayiş etdirilən tipik davranışların başa düşülməsi və monitorinqinin vacibliyini vurğulayan müasir veb tətbiqi təhlükəsizliyi aşkarlama strategiyalarının təməl daşıdır. İlkin davranışları müəyyən etməklə, təhlükəsizlik qrupları icazəsiz giriş cəhdləri və ya məlumatların pozulması kimi potensial təhlükəsizlik təhdidlərini göstərə bilən sapmaları effektiv şəkildə müəyyən edə bilər. Veb tətbiqi təhlükəsizliyində davranış təhlilinin əsas cəhətlərindən biri istifadəçi davranışının monitorinqidir. Bu, daxil olmaq cəhdləri, sessiya müddətləri, naviqasiya nümunələri və məlumat əldə etmək sorğuları daxil olmaqla, proqram daxilində müxtəlif istifadəçi qarşılıqlı əlaqələrini izləməyi əhatə edir. Bu məlumatları təhlil edərək, təhlükəsizlik sistemləri normal istifadəçi davranışı nümunələri yarada və zərərli fəaliyyət göstərə biləcək hər hansı anomaliyaları müəyyən edə bilər. Məsələn, uğursuz giriş cəhdlərinin qəfil artması və ya müəyyən bir istifadəçi tərəfindən qeyri-adi məlumatlara giriş nümunəsi həddindən artıq giriş hücumunu və ya icazəsiz giriş cəhdini göstərə bilər. Eynilə, davranış təhlili veb tətbiqinin özünün davranışının monitorinqini əhatə edir. Buraya fayl girişi, verilənlər bazası sorğuları, API sorğuları və şəbəkə trafiki kimi izləmə sistemi fəaliyyətləri daxildir. Tətbiqin normal işləməsini başa düşməklə, təhlükəsizlik qrupları qeyri-adi

resurs istifadəsi, gözlənilməz sistem sorğuları və ya təhlükəsizlik pozuntusunu göstərə biləcək şübhəli şəbəkə əlaqələri kimi anomaliyaları aşkar edə bilər. Davranış analizi üsulları davranış nümunələrinin müəyyən edilməsi və təhlili prosesini avtomatlaşdırmaq üçün tez-tez maşın öyrənməsi və süni intellekt alqoritmlərindən istifadə edir. Bu texnologiyalar ilkin davranışları qurmaq və zamanla inkişaf edən təhdidlərə dinamik şəkildə uyğunlaşmaq üçün tarixi məlumatlardan istifadə edə bilər. İstifadəçi və sistem davranışlarını davamlı olaraq izləməklə, ML ilə işləyən təhlükəsizlik sistemləri real vaxt rejimində təhlükəsizlik insidentlərini proaktiv şəkildə aşkarlaya və onlara cavab verə bilər, məlumatların pozulması və icazəsiz giriş riskini azaldır. Bundan əlavə, davranış təhlili təşkilatlara istifadəçi rollarına və davranış profillərinə əsaslanan ətraflı giriş nəzarəti və təhlükəsizlik siyasətlərini həyata keçirməyə imkan verir. İstifadəçilərin profilini yaratmaq və davranışlarına görə risklərini qiymətləndirməklə təşkilatlar yüksək riskli istifadəçilər və ya cihazlar üçün daha sərt təhlükəsizlik tədbirləri tətbiq edə, eyni zamanda normal fəaliyyətlərin fasiləsiz davam etməsinə icazə verə bilərlər. Nəticə olaraq, davranış təhlili istifadəçilərin və sistemlərin tipik davranışlarını anlamaq və izləmək üzərində fokuslanaraq veb tətbiq təhlükəsizliyinin aşkarlanmasında mühüm rol oynayır. Qabaqcıl analitika və maşın öyrənmə üsullarından istifadə etməklə təşkilatlar real vaxt rejimində təhlükəsizlik təhdidlərini effektiv şəkildə aşkarlaya və onlara cavab verə bilər və bununla da veb tətbiqlərinin ümumi təhlükəsizlik vəziyyətini gücləndirə bilər.

SAST tətbiqin məhsuldar mühitdə istifadəsindən əvvəl potensial təhlükəsizlik zəifliklərini müəyyən etmək üçün tətbiqlərin mənbə kodunu təhlil etməyə yönəlmiş veb proqram təhlükəsizliyinin aşkarlanmasının mühüm komponentidir. SAST alətləri mənbə kodunu sistematik şəkildə skan etmək üçün müxtəlif üsullardan istifadə edir, SQL inyeksiyası, XSS və proqramın təhlükəsizliyinə xələl gətirə biləcək konfiqurasiya qüsurları kimi problemləri müəyyənləşdirir. SAST alətləri tərəfindən istifadə edilən əsas üsullardan biri nümunə uyğunluğudur, burada ümumi zəifliklər göstərən əvvəlcədən təyin edilmiş nümunələr mənbə kodu daxilində axtarılır. Məsələn, bu yanaşmadan istifadə etməklə təhlükəli funksiya çağırışları, etibarlı olmayan məlumatlarla işləmə təcrübələri və ya daxiletmə validasiyasının olmadığı nümunələr

müəyyən edilə bilər. Bütün kod bazasını skan etməklə, SAST alətləri məlum zəiflikləri səmərəli şəkildə aşkarlaya və aradan qaldırılması üçün təsirli anlayışlar təqdim edə bilər (Peltier, 2017). Qabaqcıl SAST alətləri sadə nümunə uyğunlaşdırılmasından kənara çıxır və məlumat axınının təhlili və simvolik icra kimi mürəkkəb üsullardan istifadə edir. Məlumat axınının təhlili tətbiq vasitəsilə məlumat axınını izləyir, ləkələnmiş girişin təhlükəsizlik zəifliyinə səbəb ola biləcəyi potensial nöqtələri müəyyən edir. İstifadəçi tərəfindən idarə olunan daxiletmənin kod vasitəsilə necə yayıldığını izləməklə, SAST alətləri potensial inyeksiya zəifliklərini və məlumatla bağlı digər təhlükəsizlik qüsurlarını müəyyən edə bilər. Simvolik icra, tətbiq daxilində müxtəlif icra yollarını araşdırmaq üçün qabaqcıl SAST alətləri tərəfindən istifadə edilən başqa bir güclü texnikadır. Kodu simvolik girişlərlə simvolik olaraq icra etməklə, bu alətlər müəyyən şərtlər altında təhlükəsizlik zəifliyinə səbəb ola biləcək yolları müəyyən edə bilər. Bu yanaşma SAST alətlərinə ənənəvi statik analiz üsulları vasitəsilə görünməyən mürəkkəb zəiflikləri aşkar etməyə imkan verir. Bundan əlavə, SAST alətləri tez-tez inkişaf mühitləri və davamlı inteqrasiya/davamlı yerləşdirmə (CI/CD pipeline) ilə inteqrasiya edir və bu inkişaf iş axınının bir hissəsi kimi kodun avtomatlaşdırılmış şəkildə skan edilməsinə imkan verir. Təhlükəsizlik testini inkişaf prosesinə inteqrasiya etməklə, təşkilatlar inkişafın həyat dövrünün əvvəlində zəiflikləri müəyyən edə və aradan qaldıra bilər ki, bu da istehsala etibarsız kodun yerləşdirilməsi riskini azaldır. Ümumiyyətlə, SAST veb proqram təhlükəsizliyinin aşkarlanması üçün kritik texnologiyadır, tərtibatçılara və təhlükəsizlik qruplarına mənbə kodundakı təhlükəsizlik boşluqlarını istismar edilməzdən əvvəl müəyyən etmək və aradan qaldırmaq üçün vasitələr təqdim edir. Nümunə uyğunluğu, məlumat axını təhlili və simvolik icra kimi üsullardan istifadə etməklə, SAST alətləri veb proqramların təhlükəsizliyini və bütövlüyünü təmin etməkdə mühüm rol oynayır.

DAST işləyən veb proqramlarına hücumları simulyasiya edərək xarici perspektivdən zəifliklərin müəyyən edilməsinə diqqət yetirən veb proqram təhlükəsizliyinin aşkarlanmasında əsas texnologiyadır. Mənbə kodunu təhlil edən SAST-dan fərqli olaraq, DAST alətləri onların təhlükəsizlik vəziyyətini qiymətləndirmək üçün canlı proqramlarla qarşılıqlı əlaqədə olur. DAST alətləri

potensial bədniyyətlilərin davranışını təqlid edərək hədəf veb tətbiqinə müxtəlif növ sorğular göndərməklə işləyir. Bu sorğulara SQL inyeksiya, XSS və ya etibarsız server konfiqurasiyaları kimi ümumi zəifliklərdən istifadə etmək üçün nəzərdə tutulmuş girişlər daxil ola bilər (Dhillon, Backhouse, 2020). Tətbiqdən gələn cavabları təhlil edərək, DAST alətləri səhv mesajları, gözlənilməz davranışlar və ya icazəsiz məlumatların açıqlanması kimi zəiflik əlamətlərini müəyyən edə bilər. DAST-ın əsas üstünlüklərindən biri onun təkcə mənbə kodunun statik analizi ilə görünməyən zəiflikləri aşkar etmək qabiliyyətidir. Tətbiqlə real vaxt rejimində qarşılıqlı əlaqədə olmaqla, DAST alətləri müxtəlif komponentlərin və ya iş vaxtı mühitinin qarşılıqlı əlaqəsi nəticəsində yaranan zəiflikləri aşkar edə bilər. Məsələn, DAST təhlükəsizlik səhv konfiqurasiyalarını, autentifikasiya qüsurlarını və ya yalnız iş zamanı müşahidə oluna bilən girişə nəzarət problemlərini müəyyən edə bilər. Bundan əlavə, DAST real şəraitdə onun funksionallığını sınaqdan keçirməklə proqramın təhlükəsizlik vəziyyətinin hərtərəfli qiymətləndirilməsini təmin edir. Natamam kod əhatə dairəsi və ya məhdud kontekstual anlayışa görə zəiflikləri nəzərdən qaçıra bilən statik təhlildən fərqli olaraq, DAST istifadəçi daxiletməsi, sessiyanın idarə edilməsi və server tərəfində emal kimi amilləri nəzərə alaraq tətbiqi bütünlüklə qiymətləndirir. Bundan əlavə, DAST alətləri zamanla tətbiqdəki dəyişikliklərə uyğunlaşan dinamik tarama imkanları təklif edir. Hədəf tətbiqini davamlı olaraq izləmək və hücum strategiyalarını buna uyğun olaraq tənzimləməklə, DAST alətləri ortaya çıxan yeni boşluqları aşkar edə və təhlükəsizlik qruplarına vaxtında xəbərdarlıq edə bilər. İnkişaf iş axınları və avtomatlaşdırılmış sınaq pipeline-lar ilə inteqrasiya DAST alətlərinin başqa bir əsas xüsusiyyətidir. DAST skanlarını davamlı inteqrasiya/davamlı yerləşdirmə (CI/CD) pipeline-a daxil etməklə, təşkilatlar təhlükəsizlik sınaqlarının bütün inkişaf dövrü ərzində ardıcıl olaraq həyata keçirilməsini təmin edə bilər və həssas kodun istehsal mühitlərinə yerləşdirilməsi riskini azalda bilər. Nəticə olaraq, DAST veb proqram təhlükəsizliyinin aşkarlanması üçün vacib texnologiyadır və təşkilatlara işləyən proqramlara simulyasiya edilmiş hücumlar vasitəsilə xarici perspektivdən zəiflikləri müəyyən etmək üçün vasitələr təqdim edir. Statik təhlili dinamik sınaq üsulları ilə

tamamlayaraq, DAST alətləri inkişaf edən təhdidlərə qarşı veb tətbiqlərinin təhlükəsizliyini və dayanıqlığını təmin etməkdə mühüm rol oynayır (David, 2021).

RASP real vaxt rejimində təhlükəsizlik təhdidlərinin müəyyən edilməsi və azaldılması üçün proaktiv yanaşma təklif edən veb proqram təhlükəsizliyinin aşkarlanmasında qabaqcıl texnologiyanı təmsil edir. Firewall və müdaxilənin aşkarlanması sistemləri kimi xarici müdafiə vasitələrinə diqqət yetirən ənənəvi təhlükəsizlik tədbirlərindən fərqli olaraq, RASP alətləri birbaşa veb proqramlara inteqrasiya olunur ki, bu da onlara icra zamanı tətbiq davranışını izləməyə və idarə etməyə imkan verir. RASP texnologiyasının əsasını onun təhlükəsizlik təhdidlərinin baş verdiyi zaman aşkar etmək və qarşısını almaq üçün tətbiqin icrasının müxtəlif aspektlərini, o cümlədən daxiletmələr, çıxışlar və icra yollarını təhlil etmək qabiliyyəti dayanır. Daxil olan sorğuları izləməklə RASP alətləri parametrləri və faydalı yükləri SQL inyeksiyası, XSS və ya əmr inyeksiya hücumları kimi zərərli niyyət əlamətləri üçün yoxlaya bilər. Çıxış məlumatlarını təhlil edərək, RASP alətləri həssas məlumatı çıxarmaq və ya server cavablarını manipulyasiya etmək cəhdlərini müəyyən edə bilər. RASP-nin əsas üstünlüklərindən biri onun tətbiq səviyyəsində ətraflı qoruma təmin etmək qabiliyyətidir. Perimetrdə işləyən şəbəkə əsaslı təhlükəsizlik həllərindən fərqli olaraq, RASP alətləri tətbiqin daxili işlərinə dərindən baxır və onlara təhdidləri dəqiqliklə aşkar etməyə və onlara cavab verməyə imkan verir. Bu, RASP həllərinə ənənəvi perimetr müdafiəsini aşa bilən inyeksiya hücumları kimi zəifliklərə qarşı effektiv müdafiəni təmin etməyə imkan verir. Bundan əlavə, RASP həlləri iş vaxtı şərtləri və tətbiq davranışı əsasında təhlükəsizlik nəzarətlərini dinamik şəkildə uyğunlaşdıra bilər. Tətbiqin icrasına davamlı olaraq nəzarət etməklə, RASP alətləri yaranan təhdidləri və inkişaf edən hücum texnikalarını həll etmək üçün təhlükəsizlik siyasətlərini və icra mexanizmlərini tənzimləyə bilər (Pfleeger, Pfleeger, Margulies, 2017). Bu uyğunlaşma qabiliyyəti veb proqramların mürəkkəb hücumlara və sıfır gün zəifliklərinə qarşı davamlı olmasını təmin edir. Bundan əlavə, RASP texnologiyası real vaxt təhlükə kəşfiyyatı və hərəkətə keçə bilən anlayışlar üçün potensial təklif edir. RASP alətləri təhlükəsizlik hadisələri, təhdid lentləri və davranış analitikası kimi bir çox mənbədən məlumatları birləşdirərək veb proqramların təhlükəsizlik vəziyyətinə

hərtərəfli görünürlük təmin edə bilər. Bu, təhlükəsizlik qruplarına məlumatlı qərarlar qəbul etməyə və baş verən təhlükəsizlik insidentlərinə dərhal reaksiya verməyə imkan verir. Nəticə olaraq, RASP veb proqram təhlükəsizliyinin aşkarlanması üçün güclü texnologiyanı təmsil edir, real vaxt rejimində təhlükənin aşkarlanması və qarşısının alınması imkanlarını birbaşa tətbiqin icra müddətində təklif edir. Girişləri, çıxışları və icra yollarını təhlil edərək, RASP alətləri inyeksiya və saytlararası skript hücumları kimi təhlükəsizlik təhdidlərini effektiv şəkildə aşkarlaya və azalda bilər, təşkilatları inkişaf edən kibertəhlükələrə qarşı proaktiv müdafiə ilə təmin edə bilər.

Konteyner Təhlükəsizliyi, xüsusən Docker və Kubernetes kimi konteynerləşdirmə texnologiyalarının geniş tətbiqi ilə veb tətbiqi təhlükəsizliyinin aşkarlanmasının kritik aspekti kimi ortaya çıxdı. Təşkilatlar tətbiqlərini konteynerləşdirilmiş mühitlərə köçürdükcə, bu konteynerlərin təhlükəsizliyinin təmin edilməsi əsas məsələyə çevrilir. Konteyner təhlükəsizliyi həlləri konteynerləşdirilmiş mühitlərin yaratdığı unikal problemləri həll etmək və onların daxilində işləyən tətbiqlər üçün hərtərəfli qorunma təmin etmək üçün nəzərdə tutulmuşdur. Konteyner təhlükəsizliyi həllərinin əsas xüsusiyyətlərindən biri zəifliyin skan edilməsidir. Bu alətlər konteyner şəkillərini skan edir və proqram komponentlərində və konteynerlərdə paketlənmiş asılılıqlarda məlum zəiflikləri müəyyən edir. Zəiflik verilənlər bazası və təhlükəsizlik lentlərindən istifadə etməklə, konteyner təhlükəsizliyi həlləri köhnəlmiş proqram versiyaları, yanlış konfiqurasiyalar və məlum təhlükəsizlik qüsurları kimi boşluqları aşkarlaya bilər və təşkilatlara konteynerləri istehsal mühitlərinə yerləşdirməzdən əvvəl bu problemləri aradan qaldırmağa imkan verir. İş vaxtının qorunması konteyner təhlükəsizliyinin digər mühüm aspektidir. Konteyner təhlükəsizlik həlləri, icazəsiz giriş, proses manipulyasiyası və şəbəkə müdaxilələri kimi zərərli fəaliyyətləri aşkar edərək və qarşısının alınması zamanı konteynerləşdirilmiş tətbiqlərə nəzarət edir. Girişə nəzarət, şəbəkə seqmentasiyası və davranış monitorinqi siyasətlərini həyata keçirməklə, bu həllər real vaxt rejimində təhlükəsizlik təhdidlərini azalda bilər, konteynerləşdirilmiş mühitlərin ümumi təhlükəsizlik vəziyyətini gücləndirə bilər. Bundan əlavə, konteyner təhlükəsizlik həlləri girişə nəzarət və imtiyazların idarə edilməsi üçün funksiyalar təklif edir. Bu alətlər konteynerləşdirilmiş

mühitlərdə icazə və imtiyazları məhdudlaşdırmaq, icazəsiz giriş və imtiyazların artması riskini azaltmaq üçün ətraflı giriş nəzarətlərini tətbiq edir. RBAC və PoLP tətbiq etməklə təşkilatlar hücum səthini məhdudlaşdıra və potensial təhlükəsizlik pozuntularının təsirini minimuma endirə bilər. Konteyner təhlükəsizliyi həlləri həmçinin uyğunluğun idarə edilməsi və audit imkanlarını asanlaşdırır. Konteynerləşdirilmiş mühitlərdə görünürlük təmin etməklə və audit jurnalları və uyğunluq hesabatları yaratmaqla bu alətlər təşkilatlara tənzimləyici tələblərə və sənaye standartlarına riayət olunmasını nümayiş etdirməyə kömək edir. Bu, maliyyə, səhiyyə və hökumət kimi ciddi uyğunluq mandatları olan sektorlarda xüsusilə vacibdir. Nəticə olaraq, Konteyner Təhlükəsizliyi müasir İT mühitlərində veb proqram təhlükəsizliyinin aşkarlanması üçün kritik texnologiyadır. Zəifliyin skan edilməsi, iş vaxtının qorunması, girişə nəzarət və uyğunluğun idarə edilməsi kimi funksiyaları təklif etməklə, konteyner təhlükəsizliyi həlləri təşkilatlara konteynerləşdirilmiş tətbiqlərini inkişaf edən kibertəhlükələrdən qorumağa və normativ tələblərə uyğunluğu təmin etməyə imkan verir. Konteynerləşmə sürət qazanmağa davam etdikcə, etibarlı konteyner təhlükəsizliyi həllərinə investisiya etmək veb proqramların təhlükəsizliyini və bütövlüyünü təmin etmək üçün getdikcə vacib olur.

Blokçeyn texnologiyası, tətbiq fəaliyyətinin mərkəzləşdirilməmiş və müdaxiləyə davamlı qeydini təmin etməklə veb proqram təhlükəsizliyinin aşkarlanmasını artırmaq potensialına görə əhəmiyyətli diqqət qazanmışdır. Əvvəlcə Bitcoin kimi kriptovalyutalar üçün əsas texnologiya kimi inkişaf etdirilən blokçeyn, təhlükəsiz məlumatların saxlanması və tranzaksiyaların yoxlanılması da daxil olmaqla, rəqəmsal valyutalardan kənarda geniş tətbiqetmələr təklif etmək üçün inkişaf etmişdir. Blokçeyn texnologiyasının əsas xüsusiyyətlərindən biri onun dəyişməz və şəffaf əməliyyatlar kitabçasını yaratmaq qabiliyyətidir. Blokçeyn-dəki hər bir blok əvvəlki blokun kriptoqrafik həşini(hash) ehtiva edir və təhlükəsiz və müdaxiləyə davamlı şəkildə bir-birinə bağlanmış bloklar zəncirini yaradır. Bir əməliyyat blokçeynində qeydə alındıqdan sonra, şəbəkə iştirakçılarının razılığı olmadan dəyişdirilə və ya silinə bilməz, bu da onu saxtalaşdırmaya və icazəsiz dəyişikliklərə qarşı yüksək dərəcədə davamlı edir (Acar, 2020). Veb tətbiqi təhlükəsizliyinin aşkarlanması kontekstində

blokçeyn tətbiq fəaliyyətinin, o cümlədən istifadəçi qarşılıqlı əlaqəsi, verilənlərə giriş hadisələri və sistem fəaliyyətlərinin saxtakarlığa qarşı qeydlərini yaratmaq üçün istifadə edilə bilər. Bu fəaliyyətləri blokçeyn əsaslı kitabda qeyd etməklə, təşkilatlar tətbiq daxilində görülən bütün hərəkətlərin yoxlanıla bilən audit izini yarada bilər ki, bu da bədniyyətlilərin məlumat və ya sistemləri iz qoymadan təhlükəyə atmasını çətinləşdirir. Bundan əlavə, blokçeyn texnologiyası veb proqramlarda saxlanılan məlumatların bütövlüyünü və şəffaflığını artıra bilər. Məlumatları blokçeyn şəbəkəsində saxlamaqla təşkilatlar onların zamanla dəyişməz və etibarlı qalmasını təmin edə bilərlər. Bu, istifadəçi etimadnamələri, maliyyə əməliyyatları və məlumatların bütövlüyünün vacib olduğu tənzimləmə uyğunluğu məlumatları kimi həssas məlumatlar üçün xüsusilə faydalıdır. Bundan əlavə, blokçeyn əsaslı ağıllı müqavilələr veb tətbiqləri daxilində təhlükəsizlik siyasətlərini və icra mexanizmlərini avtomatlaşdırmaq üçün istifadə edilə bilər. Ağıllı müqavilələr blokçeynində kodlanmış əvvəlcədən müəyyən edilmiş qaydalar və şərtlərlə öz-özünə icra olunan müqavilələrdir. Girişə nəzarət, məlumatların yoxlanılması və təhlükəsizlik protokollarını idarə etmək üçün ağıllı müqavilələr tətbiq etməklə təşkilatlar insan səhvi və icazəsiz giriş riskini azaldaraq təhlükəsizlik tədbirlərini avtomatik tətbiq edə bilər. Nəticə olaraq, blokçeyn texnologiyası tətbiq fəaliyyətinin mərkəzləşdirilməmiş və saxtakarlığa qarşı qeydini təmin etməklə veb proqram təhlükəsizliyinin aşkarlanmasının artırılması üçün perspektivli imkanlar təklif edir. Blokçeyn əsaslı kitablardan istifadə etməklə, təşkilatlar yoxlanıla bilən audit yolları yarada, məlumatların bütövlüyünü təmin edə və veb proqramlarında təhlükəsizlik siyasətlərini avtomatlaşdıra, bununla da kiber təhdidlərə qarşı ümumi təhlükəsizlik vəziyyətini və dayanıqlığını gücləndirə bilər. Blokçeyn inkişaf etməyə davam etdikcə, onun veb proqramların təhlükəsizliyinin aşkarlanmasında inqilab etmək potensialının artacağı gözlənilir ki, bu da yaranan təhdidlərdən və zəifliklərdən qorunmaq üçün yeni yollar təklif edir.

2.4. Veb tətbiqlərin təhlükəsizlik dizaynında qiymətləndirilmə üsulları, beynəlxalq standartlar və davamlı təkmilləşdirilmə yolları.

Veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi onların potensial zəiflikləri və təhdidləri nə dərəcədə effektiv şəkildə həll etdiyini təhlil etməyi əhatə edir. Bu məqsədlə ən çox istifadə olunan üsullar bunlardır:

1. Təhdidin modelləşdirilməsi: Potensial təhlükələrin müəyyən edilməsi, ehtimalının və təsirinin qiymətləndirilməsi, təhlükəsizlik nəzarətlərinin layihələndirilməsi.

2. Təhlükəsizlik tələblərinin təhlili: Veb tətbiqinin autentifikasiya, avtorizasiya, məlumatların yoxlanılması və şifrələmə kimi təhlükəsizlik tələblərinə uyğunluğunun qiymətləndirilməsi.

3. Kod baxışı: SQL inyeksiya, XSS və təhlükəsiz olmayan birbaşa obyekt istinadları kimi ümumi zəifliklərin avtomatlaşdırılmış alətlər və əl ilə yoxlanması.

4. Penetrasiya testi: Real dünya hücumlarını simulyasiya edərək digər üsullarla görünməyən zəiflikləri müəyyən etmək və dizayn nümunələrinin effektivliyini yoxlamaq.

5. Təhlükəsizlik arxitekturasına baxış: Veb tətbiqinin ümumi təhlükəsizlik arxitekturasını və ən yaxşı təcrübələrə uyğunluğunu qiymətləndirmək.

6. Təhlükəsizliyə nəzarətlərin qiymətləndirilməsi: Girişin yoxlanılması, çıxışın kodlaşdırılması və girişə nəzarət mexanizmlərinin effektivliyinin təhlili.

7. Uyğunluğun qiymətləndirilməsi: Veb tətbiqinin OWASP Top 10, PCI DSS və GDPR kimi təhlükəsizlik standartlarına uyğunluğunu təmin etmək (Radack, DeBar, 2019).

8. Təhlükəsizlik test framevorkləri: Təhlükəsizlik testi proseslərinin avtomatlaşdırılması üçün alətlərdən istifadə edərək zəiflikləri müəyyən etmək.

9. Sənədləşməyə baxış: Təhlükəsizlik dizayn nümunələri ilə bağlı ən müasir və hərtərəfli sənədlərin yoxlanılması.

10. Qarşılaşdırmalar və metriklər: Təhlükəsizlik insidentlərinə cavab müddəti, pozuntuların tezliyi və təhlükəsizlik siyasətlərinə uyğunluq kimi əsas göstəricilərin ölçülməsi.

Bu metodların kombinasiyasından istifadə edərək təşkilatlar təhlükəsizlik dizayn nümunələrinin effektivliyini qiymətləndirə və ümumi təhlükəsizlik vəziyyətini yaxşılaşdırmaq üçün təkmilləşdirilməli sahələri müəyyən edə bilərlər.

ISO/IEC 27001 informasiya təhlükəsizliyi idarəetmə sistemləri (ISMS) üçün qlobal səviyyədə tanınmış framevork təklif edir və təşkilatlara informasiya təhlükəsizliyinin idarə edilməsini yaratmaq, həyata keçirmək, saxlamaq və daim təkmilləşdirmək imkanı verir. Veb tətbiqi təhlükəsizlik dizayn nümunələrini öyrənməsə də, onun ümumi prinsipləri və təlimatları veb proqramların təhlükəsizlik vəziyyətinin qiymətləndirilməsi üçün güclü zəmin yaradır. ISO/IEC 27001-in əsas aspektlərindən biri onun risklərin idarə edilməsinə vurğulanmasıdır. Standart informasiya təhlükəsizliyi risklərinin müəyyən edilməsi, qiymətləndirilməsi və aradan qaldırılması üçün sistematik yanaşmanın tərəfdarıdır. Bu yanaşma veb proqram təhlükəsizliyində yüksək dərəcədə tətbiq olunur. Hərtərəfli risk qiymətləndirmələri apararaq, təşkilatlar veb tətbiqlərində potensial zəiflikləri müəyyən edə və müvafiq təhlükəsizlik tədbirləri hazırlaya bilərlər.

ISO/IEC 27001 davamlı təkmilləşdirmə konsepsiyasını təşviq edir və inkişaf edən təhdidlərə uyğunlaşmaq üçün təhlükəsizlik tədbirlərinin müntəzəm monitorinqini müdafiə edir. Bu iterativ proses yeni zəifliklərin ortaya çıxdığı və hücum üsullarının inkişaf etdiyi veb proqram təhlükəsizliyinə uyğun gəlir. Bununla təşkilatlar veb tətbiqi təhlükəsizlik dizayn nümunələrinin effektivliyini təmin edə bilərlər. Standart, həmçinin informasiya təhlükəsizliyinə vahid yanaşmanın vacibliyini vurğulayır. Yalnız texniki tədbirlərə diqqət yetirmək əvəzinə, təşkilati kontekst, qanuni tələblər və maraqlı tərəflərin gözləntilərini nəzərə alaraq, təhlükəsizlik vəziyyətini gücləndirmək mümkündür. ISO/IEC 27001-ə uyğun sertifikatlaşdırmanın əldə edilməsi təşkilatın informasiya təhlükəsizliyinə sadiqliyini göstərir və maraqlı tərəflər üçün təminat yaradır.

NIST (National Institute of Standards and Technology) kibertəhlükəsizlik sahəsində aparıcı orqan kimi, təşkilatların təhlükəsizlik vəziyyətini artırmağa yönəlmiş çoxlu təlimatlar təklif edir. NIST SP 800-53 veb proqramların təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi üçün əhəmiyyətlidir. Bu nəşr veb proqramların

təhlükəsizlik ehtiyaclarını qarşılamaq üçün təhlükəsizlik nəzarətlərinin əhatəli kataloqunu təqdim edir. NIST SP 800-53, təhlükəsizlik nəzarətlərinin təsnifatına və dəqiqləşdirilməsinə sistemli yanaşma təklif edir (Eloff, Eloff, 2019). Nəşr təhlükəsizlik nəzarətlərini giriş nəzarəti, insidentlərə reaksiya və təhlükəsizliyin qiymətləndirilməsi kimi sahələr üzrə təşkil edir, bu da veb tətbiqi təhlükəsizlik dizayn nümunələrinə uyğun nəzarət vasitələrinin müəyyən edilməsini asanlaşdırır.

NIST SP 800-53 risklərin idarə edilməsi və davamlı monitorinqin vacibliyini vurğulayır. Təhlükəsizlik nəzarətlərinin kataloqundan əlavə, həmin nəzarətlərin həyata keçirilməsi və qiymətləndirilməsi üzrə əlavə təlimatlar təklif edir. NIST-in risklərin idarə edilməsi prinsiplərinə əməl etməklə təşkilatlar veb tətbiqlərinin təhlükəsizlik vəziyyətini gücləndirə və kibertəhlükələri azalda bilərlər.

BSIMM Synopsys Software Integrity Group tərəfindən hazırlanmış və proqram təminatının təhlükəsizliyi framevorkü kimi tanınır. Rəsmi beynəlxalq standart olmasa da, BSIMM proqram təminatının təhlükəsizliyi praktikalarının, xüsusən də veb proqram təhlükəsizliyinin dizayn nümunələri ilə bağlı olanların qiymətləndirilməsi və təkmilləşdirilməsi üçün strukturlaşdırılmış metodologiya təklif edir. BSIMM, müxtəlif təşkilatlarda real proqram təminatının təhlükəsizliyi təşəbbüslərinin müşahidəsi və təhlilinə əsaslanan təsviri modeldir və ən yaxşı təcrübələr toplusundan ibarətdir. Bu framevork idarəetmə, kəşfiyyat, təhlükəsiz proqram təminatının inkişafı və yerləşdirmə kimi proqram təminatının təhlükəsizliyinin müxtəlif aspektlərini əhatə edən on iki təcrübə dəsti ətrafında qurulub. Hər bir təcrübə üçün müxtəlif yetkinlik mərhələlərini təmsil edən səviyyələr və xüsusi fəaliyyətlər müəyyən edilir. Bu detallı yanaşma təşkilatlara təhlükəsizlik yetkinliklərinin cari vəziyyətini qiymətləndirməyə, təkmilləşdirilməli sahələri müəyyən etməyə və zamanla irəliləyişi izləməyə imkan verir.

BSIMM-in empirik əsası onun güclü tərəflərindən biridir. Sənaye liderləri və praktikantlardan toplanmış empirik sübutlara əsaslanaraq, model praktiki uyğunluq və etibarlılıq qazanır. BSIMM framevorkü, proqram təminatının təhlükəsizliyi sahəsində yüksək dərəcədə uyğunlaşdırıla bilir və veb proqram təhlükəsizliyi dizayn nümunələrinin qiymətləndirilməsi və təkmilləşdirilməsi üçün effektiv şəkildə tətbiq

oluna bilər. Framevork davamlı araşdırma və icmadan rəy əsasında müntəzəm yeniləmələr və təkmilləşdirmələrlə dinamik inkişaf edir ki, bu da modelin aktual qalmasını və proqram təminatının təhlükəsizliyi sahəsində ən son inkişaflara uyğun olmasını təmin edir.

PCI DSS kibertəhlükəsizlik sahəsində əsas beynəlxalq standart kimi xidmət edir və ödəniş kartı məlumatlarının qorunmasına diqqət yetirir. PCI DSS-nin əsas məqsədi həssas kart sahibi məlumatlarını qorumaq olsa da, bu standart ödəniş kartı məlumatlarını idarə edən veb proqramların təhlükəsizliyini təmin etmək üçün də dəyərli tələblər və ən yaxşı texnikalar təqdim edir. PCI DSS uyğunluğu məlumatların pozulması və icazəsiz girişlə bağlı riskləri azaltmaq üçün xüsusi veb proqram təhlükəsizlik dizayn nümunələrinin həyata keçirilməsini tələb edir. Standart on iki əsas tələbdən ibarətdir və bunlar müxtəlif təhlükəsizlik nəzarətlərini və ödəniş kartı məlumatlarının təhlükəsiz idarə olunmasını təmin etməyə yönəlib. PCI DSS-nin tələb 6-sı təhlükəsiz sistemlərin və proqramların işlənib hazırlanmasına və saxlanmasına yönəlib və təşkilatlara ümumi veb proqram zəifliklərindən qorunmaq üçün təhlükəsiz kodlaşdırma təcrübələri və tədbirləri həyata keçirməyi tapşırır.

Bu tələblərdən əlavə, PCI DSS təhlükəsizlik nəzarətinin effektivliyini təmin etmək üçün davamlı təhlükəsizlik sınaqlarının və zəifliyin qiymətləndirilməsinin vacibliyini vurğulayır. Tələb 11, potensial zəiflikləri müəyyən etmək üçün veb proqramların müntəzəm penetrasiya testini tələb edir. Təşkilatlar bu testlər vasitəsilə təhlükəsizlik problemlərini vaxtında müəyyənləşdirə və həll edə bilərlər. Nəticə olaraq, PCI DSS ödəniş kartı məlumatlarının təhlükəsizliyinə diqqət yetirir və bu məlumatları idarə edən veb proqramların təhlükəsizliyini təmin etmək üçün dəyərli təlimatlar təqdim edir. Tələblərə riayət etməklə təşkilatlar veb proqramlarının təhlükəsizliyini gücləndirə və həssas ödəniş kartı məlumatlarını icazəsiz giriş və istismardan qoruya bilərlər.

MITRE Corporation tərəfindən idarə olunan CWE, proqram və aparat zəifliklərinin standartlaşdırılmış dilini təmin edərək ümumi təhlükəsizlik zəifliklərini müəyyənləşdirmək, kateqoriyalara bölüşmək və təsvir etmək üçün xidmət edir. CWE, veb tətbiqi təhlükəsizlik dizayn nümunələrinin standart olmasa da, bu dizayn

nümunələrinin azaldılması lazım olan zəiflikləri müəyyənləşdirmək və aradan qaldırmaqda mühüm rol oynayır. CWE, hər bir zəiflik üçün özünəməxsus identifikator və təsviri ilə fərqli siniflər təklif edir. Bu siniflər, proqram və aparat sistemlərində geniş yayılmış təhlükəsizlik zəifliklərini də əhatə edir. CWE-yə istinad edərək, təşkilatlar veb tətbiqləri üçün risk yaradan xüsusi zəifliklər haqqında məlumat əldə edə və bu zəifliklərin səbəblərini və potensial təsirlərini başa düşə bilərlər. CWE-nin veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsinə töhfə verməsinin səbəblərindən biri təhlükəsizlik zəifliklərinin müzakirəsi üçün standartlaşdırılmış lüğətin təqdim edilməsidir. Dizayn nümunələri, ümumi təhlükəsizlik problemlərini həll etmək üçün təkrar istifadə edilə bilən həllər və ya ən yaxşı texnikalar təklif etməklə nəzərdə tutulub. Xüsusi zəifliklərlə əlaqəli CWE identifikatorlarına istinad edərək, təşkilatlar dizayn nümunələrinin bu zəif cəhətləri effektiv şəkildə azaltmasını və məlum təhlükələrə qarşı adekvat müdafiəni təmin edə bilərlər. CWE siyahısına zəifliklərin şiddəti və təsirinin kəmiyyət ölçüsünü təmin edən CVSS balları kimi göstəricilər daxildir (Menzel, Waidner, 2021). CWE təsnifatlarına və əlaqəli CVSS xallarına əsaslanaraq zəiflikləri prioritetləşdirməklə təşkilatlar öz resurslarını veb tətbiqlərinə təsir edən ən kritik təhlükəsizlik risklərinin həllinə yönəldə bilərlər. CWE, veb tətbiqi təhlükəsizliyi ilə bağlı yeni zəiflikləri və yaranan təhlükələri özündə birləşdirərək təkamül və genişlənməyə davam etməklə kibertəhlükəsizlik icması daxilində əməkdaşlığı və bilik mübadiləsini dəstəkləyir. CWE, bütün dünyada təhlükəsizlik tədqiqatçılarının, praktikantların və təşkilatların töhfələrini təşviq edir. İcmanın kollektiv təcrübəsindən istifadə etməklə, CWE veb tətbiqi təhlükəsizliyi ilə bağlı anlayışlarını təkmilləşdirə bilər və veb tətbiqlərinin təhlükəsizliyini artırmaq üçün daha geniş kibertəhlükəsizlik icması ilə əməkdaşlıq edə bilər.

PoLP icazəsiz giriş və potensial istismara qarşı əsas müdafiə qatını təşkil edir və veb tətbiqlərin təhlükəsiz dizayn nümunələrinin təməl daşı kimi qəbul edilir. Bu prinsipin təkmilləşdirilməsi texnoloji həlləri və təşkilati təcrübələri əhatə edən çoxşaxəli yanaşma tələb edir. PoLP-ni təkmilləşdirməyin əsas aspektlərindən biri istifadəçi giriş hüquqlarının və sistem icazələrinin müntəzəm auditlərinin aparılmasıdır. Bu, hər bir istifadəçi rolu və ya hesabına verilən icazələrin vaxtaşırı

yenidən qiymətləndirilməsini və ən az imtiyaz prinsipinə uyğunluğunun təmin edilməsini əhatə edir. Avtomatlaşdırılmış alətlər anomaliyaları və müəyyən edilmiş siyasətlərdən kənarlaşmaları aşkar etmək üçün istifadə edilə bilər, əlavə araşdırma üçün potensial təhlükəsizlik risklərini qeyd edir.

Güclü autentifikasiya və avtorizasiya mexanizmlərinin tətbiqi də PoLP-nin effektiv həyata keçirilməsi üçün vacibdir. Güclü parol siyasətləri, çoxfaktorlu autentifikasiya və sessiya idarəetmə üsulları istifadəçilərin şəxsiyyətini yoxlamaq və yalnız səlahiyyətli mənbələrə giriş vermək üçün tətbiq edilməlidir. Rol əsaslı giriş nəzarəti (RBAC) və atribut əsaslı giriş nəzarəti (ABAC) kimi metodlar istifadəçi rollarına, atributlarına və kontekst faktorlarına əsasən giriş hüquqlarını dəqiq müəyyən etməyə və tətbiq etməyə kömək edə bilər.

PoLP-nin təkmilləşdirilməsinin digər vacib aspekti bu prinsipin veb tətbiqlərin dizayn və inkişaf mərhələlərinə daxil edilməsidir. Tərtibatçılar təhlükəsiz kodlaşdırma təcrübələrinə riayət etməli və ən az imtiyaz prinsiplərini təşviq edən freymvork və kitabxanalardan istifadə etməlidir. Girişin doğrulanması, çıxışın kodlaşdırılması və parametrləşdirilmiş sorğular kimi təhlükəsiz kodlaşdırma standartları ümumi təhlükəsizlik zəifliklərini azaltmaq üçün inkişaf iş prosesinə inteqrasiya edilməlidir. Davam edən təhsil və təlim təşəbbüsləri də təşkilat daxilində təhlükəsizlik şüurunu artırmaq üçün vacibdir. Təhlükəsizlik məlumatlandırma proqramları, fişinq simulyasiyaları və insidentlərə cavab təlimləri işçilərə ən az imtiyaz prinsipinin əhəmiyyətini dərk etməyə və bu imtiyazı qorumaqda öz rollarını başa düşməyə kömək edə bilər.

Xülasə, PoLP-yə əsaslanan veb tətbiqlərin təhlükəsizlik dizayn nümunələrinin davamlı olaraq təkmilləşdirilməsi texniki tədbirləri, təşkilati siyasətləri və istifadəçi məlumatlandırma təşəbbüslərini əhatə edən vahid yanaşmanı tələb edir. Ən az imtiyaz prinsiplərini proqram təminatının inkişaf dövrünün hər bir aspektinə inteqrasiya etməklə və təhlükəsizlik şüurunun mədəniyyətini inkişaf etdirməklə təşkilatlar potensial istismarlara qarşı müdafiələrini gücləndirə və həssas məlumatları icazəsiz girişdən qoruya bilərlər.

OWASP Top 10 ən aktual təhlükəsizlik risklərinin müəyyən edilməsi və həlli üçün mühüm resurs rolunu oynayır. OWASP Top 10-un proqram təminatının inkişaf dövrünə (SDLC) inteqrasiyası təhlükəsizlik risklərini azaltmağa kömək edə bilər (Stamp, 2019). Bu, OWASP Top 10-da vurğulanan zəifliklərin aradan qaldırılmasına diqqət yetirməklə, tələblərin toplanması və dizayn mərhələləri zamanı hərtərəfli risk qiymətləndirmələrinin aparılmasını əhatə edir. OWASP tərəfindən müəyyən edilmiş riskləri azaltmaq üçün təhlükəsizlik nəzarətləri dizayn nümunələrinə daxil edilməlidir. OWASP-ın geniş resurslarından istifadə, təhlükəsizlik tədbirlərinin effektiv həyata keçirilməsinə kömək edə bilər. OWASP-ın tövsiyələrinə riayət etməklə təşkilatlar təhlükəsizlik mövqelərini gücləndirə bilərlər. Təhlükəsizlik qiymətləndirmələrinin və penetrasiya testlərinin aparılması OWASP Top 10 zəifliklərinin azaldılmasında dizayn nümunələrinin effektivliyini təsdiq etməyə kömək edə bilər.

Veb tətbiqi təhlükəsizlik dizayn nümunələrinin təkmilləşdirilməsi icazəsiz girişdən qorunmaq və potensial təhlükəsizlik risklərini azaltmaq üçün autentifikasiya və avtorizasiyaya kompleks yanaşma tələb edir. Əsas strategiyalardan biri autentifikasiya texnologiyaları və ən yaxşı təcrübələrdən xəbərdar olmaqdır. Bu, veb tətbiqlərinin istifadəçi hesablarını və həssas məlumatları qorumaq üçün ən son təhlükəsizlik tədbirlərindən istifadə etməsini təmin edir. Çox faktorlu autentifikasiya (MFA) kimi güclü autentifikasiya mexanizmlərinin tətbiqi istifadəçilərdən parollar, biometrik məlumatlar və ya birdəfəlik kodlar kimi çoxsaylı faktorlardan istifadə etməklə öz şəxsiyyətlərini yoxlamağı tələb edir və veb proqramların təhlükəsizliyini əhəmiyyətli dərəcədə artırır. Yeni autentifikasiya üsullarını davamlı olaraq qiymətləndirmək və qəbul etmək icazəsiz girişə qarşı güclü müdafiəni saxlamaq üçün vacibdir.

Müvafiq icazə nəzarətlərinin həyata keçirilməsi də çox vacibdir. RBAC və ABAC kimi icazə modelləri təşkilatlara istifadəçi rollarına və ya digər kontekst faktorlarına əsaslanan giriş siyasətlərini müəyyən etməyə imkan verir. İstifadəçi rollarında və təşkilati strukturlarda dəyişikliklər əsasında bu icazə nəzarətlərinin təkmilləşdirilməsi və tənzimlənməsi icazəsiz giriş və məlumat pozuntusu riskini azaltmağa kömək edir (Whitman, Mattord, 2019). Girişə nəzarət siyasətlərini və

icazələrini müntəzəm olaraq nəzərdən keçirmək və yeniləmək autentifikasiya və avtorizasiya ilə bağlı veb tətbiqi təhlükəsizlik dizayn nümunələrinin təkmilləşdirilməsinin digər mühüm aspektidir. Dövri giriş yoxlamaları və auditlər lazımsız imtiyazları müəyyən etməyə və aradan qaldırmağa, hücum səthini azaltmağa kömək edir. Avtomatlaşdırma vasitələri istifadəçilərin ən az imtiyaz prinsipinə uyğun olaraq, yalnız öz vəzifələrini yerinə yetirmək üçün lazım olan resurslara çıxışını təmin edərək, giriş icazələrinin idarəedilməsini sadələşdirə bilər.

Sessiyanın oğurlanmasının və istifadəçi hesablarına icazəsiz girişin qarşısını almaq üçün güclü sessiya idarəetmə təcrübələrinin həyata keçirilməsi də vacibdir. Bu, sessiya nişanlarının etibarlı idarə edilməsini, sessiyanın fasilələrinin tətbiqini və həssas hərəkətlər üçün sessiyanın yenidən autentifikasiyasını əhatə edir. Davamlı olaraq monitorinq və giriş sessiyası fəaliyyəti icazəsiz giriş cəhdlərini aşkar etməyə və azaltmağa kömək edir.

Ümumilikdə, identifikasiya və avtorizasiya ilə bağlı veb tətbiqi təhlükəsizliyi dizayn nümunələrinin təkmilləşdirilməsi güclü autentifikasiya mexanizmlərinin, müvafiq avtorizasiya nəzarətlərinin və sessiya idarəetmə təcrübələrinin həyata keçirilməsini tələb edir. Giriş nəzarətlərini müntəzəm olaraq nəzərdən keçirmək və yeniləmək, avtomatlaşdırma alətlərindən istifadə etmək təşkilatların veb tətbiqlərinin təhlükəsizlik vəziyyətini yaxşılaşdırmasına və təhlükəsizlik risklərindən qorunmasına kömək edir.

Təhlükəsizlik başlıqları müxtəlif hücumlara qarşı əlavə müdafiə qatını təmin edir. CSP, X-Content-Type-Options, X-Frame-Options və X-XSS-Protection kimi təhlükəsizlik HTTP başlıqları veb proqramların təhlükəsizlik dizaynını əhəmiyyətli dərəcədə təkmilləşdirir. CSP, XSS hücumlarının qarşısını alır, X-Content-Type-Options MIME-ikləmə hücumlarını azaldır, X-Frame-Options klik hücumlarının qarşısını alır və X-XSS-Protection XSS hücumlarını aşkar edir və azaldır. Təhlükəsizlik başlıqlarını tətbiq etməklə yanaşı, müntəzəm təhlükəsizlik qiymətləndirmələri və penetrasiya testləri tətbiqin arxitekturasında və kod bazasında zəiflikləri müəyyən etməyə və aradan qaldırmağa kömək edir. Təhlükəsiz kodlaşdırma təcrübələrindən istifadə və ən az imtiyaz prinsipinə əməl etmək təhlükəsiz veb tətbiqi

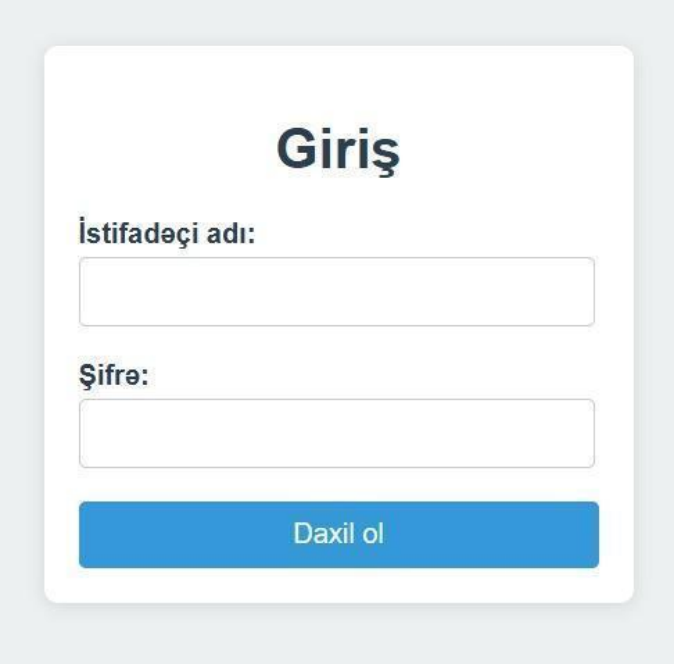
mühitinin saxlanması üçün vacibdir. Təhlükəsizliyə proaktiv yanaşma və təhlükəsizlik mülahizələrini inkişaf dövrü boyunca inteqrasiya etməklə, təşkilatlar riskləri effektiv şəkildə azalda və inkişaf edən təhdidlərdən veb tətbiqlərini qoruya bilərlər.

III FƏSİL. VEB TƏTBİQİNİN TƏHLÜKƏSİZLİYİ DİZAYN NÜMUNƏLƏRİNİN TƏHLİL VƏ TƏKMİLLƏŞDİRMƏ PROSESLƏRİ

3.1. Tətbiqin ümumi baxışı: Layihənin məqsədi və funksionallığı

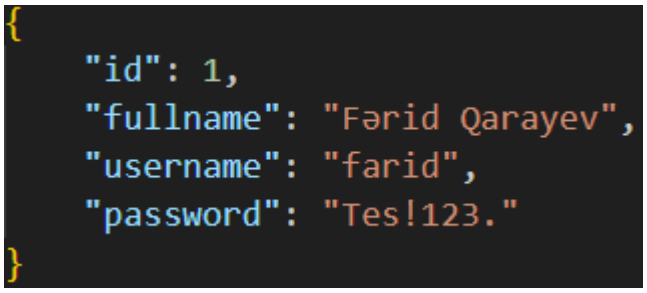
Bu hissədə tətbiqin qurulması, əsas xüsusiyyətləri, məqsədləri və funksionallıqları barədə ümumi məlumat veriləcək. Burada layihənin məqsədi, tətbiqin istifadə sahələri və hədəf auditoriyası izah ediləcək. Eyni zamanda, tətbiqin qurulma mərhələləri və ilkin dizayn aspektləri təqdim ediləcək, istifadəçilərə tətbiqin necə işlədiyini anlamaqda yardımçı olmaq üçün vacib məqamlar vurğulanacaq.

Praktiki olaraq görmək üçün hazırlanan kiçik tətbiq, təhsil prosesləri ilə bağlı məlumatları əldə etmək və idarə etmək üçün tətbiq edilmişdir. Bu tətbiq, tələbələrin öz şəxsi məlumatlarına, tələbə məlumatlarına, dərs mərkəzi məlumatlarına daxil olmağa imkan verir. Əsas məqsəd, tələbələrin təhsil prosesləri ilə bağlı məlumatlarına asan və effektiv şəkildə çatmağını təmin etməkdir.



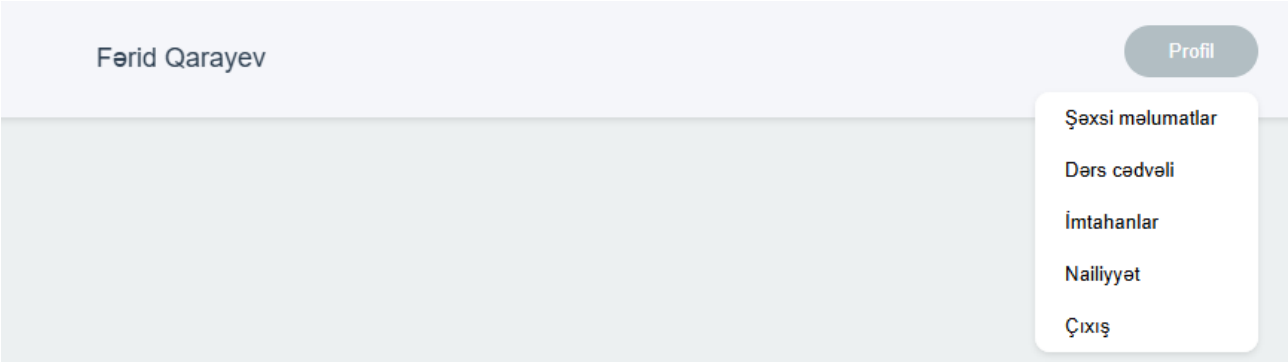
Şəkil 3.1

Tətbiqimizdəki giriş səhifəsində (*Şəkil 3.1*), tələbələr öz istifadəçi adları və şifrələri ilə sistemə daxil ola bilərlər. Daxil olduqda, tələbələr şəxsi məlumatlarına, dərs vaxtlarına, imtahanlar və tapşırıqlar kimi təhsil prosesləri ilə bağlı məlumatlara çatmaq imkanına malik olurlar.



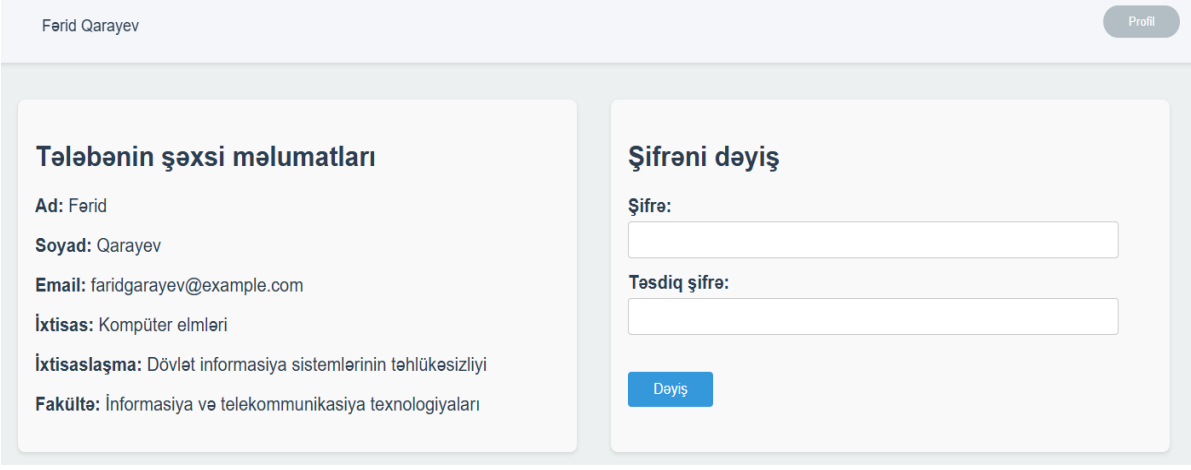
Şəkil 3.2

Burada istifadəçi adı və şifrənin daxil edilməsi ilə, daxil ol düyməsinə tıklanması nəticəsində, bir servis çağrılır və bu servis tərəfindən bir cavab (*Şəkil 3.2*) qaytarılır. Qayıdan cavabdan istifadə edərək növbəti səhifəyə keçid edilir.



Şəkil 3.3

Şəkil 3.3-dəki pəncərədən şəxsi məlumatlar bölməsinə daxil olaraq tələbə məlumatlarını əldə edə və şifrəsini dəyişə bilər.



Şəkil 3.4

Şəkil 3.4-də şifrə və təsdiq şifrəni daxil edərək dəyiş düyməsinə klik etdikdə, servisə sorğu göndərilir və serverdən istifadəçiyə “ok” qayıdarsa şifrə uğurla dəyişdirilmiş sayılır.

3.2. Tətbiqin texniki təhlükələrinin qiymətləndirilməsi.

Bu bölmədə, tətbiqin texniki təhlilinə yönələcək və zəiflikləri araşdırılmışdır. Veb tətbiqlərinə qarşı müxtəlif hücum metodları mövcuddur ki, bunlar tətbiqin təhlükəsizliyini ciddi şəkildə təhdid edə bilər. Burada, tətbiqə CSRF ilə baş verə biləcək hücumu göstərib, bu hücumun yaranma səbəbləri, metodları və necə həyata keçirildiyi barədə məlumat verəcək, həmçinin tətbiqin zəifliyini və hansı fəsadlara yol açа biləcəyi göstəriləcəkdir. Tətbiqin texniki təhlili, təhlükəsizlik boşluqlarının müəyyən edilməsi və bu boşluqlardan sui-istifadə etməklə həyata keçirilə biləcək hücumların qarşısını almaq üçün vacibdir. Bu analiz nəticəsində əldə edilən məlumatlar əsasında tətbiqin təhlükəsizlik səviyyəsinin artırılması üçün effektiv müdafiə strategiyaları hazırlanacaqdır.

Hər hansı bir istifadəçi aktiv sessiyada olduğu zaman onun e-poçt qutusuna bir mesaj gəlir və həmin mesaja tıklaması xahiş edilir. Əgər istifadəçi bu e-poçta tıklayarsa, onun şifrəsi dəyişdiriləcək və o, bir daha öz hesabına daxil ola bilməyəcəkdir. Bu, CSRF (Cross-Site Request Forgery) hücumunun bir nümunəsidir. Belə hücumlar, istifadəçinin xəbəri olmadan onun adına zərərli əməliyyatların aparılmasına səbəb ola bilər.

Şəkil 3.5

Linkə tıkladıqda, Şəkil 3.5-də göstərilən səhifə açılacaq və istifadəçi bunu normal bir səhifə kimi qəbul edəcəkdir. Əslində, arxaplanda görünməyən bir script işləyir və istifadəçinin şifrəsini dəyişdirir. Bu, istifadəçini yanıltmaq və onun icazəsiz

hərəkətlərə səbəb olmaq üçün yayılmış bir CSRF (Cross-Site Request Forgery) hücumu metodudur.

```
<h1>Səhifəmizə xoş gəlmisiniz!</h1>

<form hidden target="formFrame" action="./edit.html" method="post" id="passwordForm">
    <div class="inputContainer">
        <label class="label" for="password">Şifrə:</label>
        <input type="password" id="password" name="password" value="csrfattack" class="input">
    </div>
    <div class="inputContainer">
        <label class="label" for="confirm">Təsdiq şifrə:</label>
        <input type="password" id="confirm" name="confirm" value="csrfattack" class="input">
    </div>
    <button type="submit" class="button" onclick="handleSave()">Dəyiş</button>
</form>

<body onload="document.forms[0].submit">

<iframe hidden name="formFrame" id="formFrame"></iframe>
```

Şəkil 3.6

Göstərilən kod parçasında (Şəkil 3.6), skript işləyərək istifadəçinin şifrəsini "csrfattack" olaraq dəyişdirir. Şəkil 3.5-də qeyd edilən "Səhifəmizə xoş gəlmisiniz!" mesajı görünsə də, form gizlədildiyinə görə istifadəçiyə görünmür. Bu, istifadəçini aldatmaq üçün istifadə olunan tipik bir CSRF hücumudur.

```
function handleSave() {
    const password = document.getElementById('password').value;
    const confirm = document.getElementById('confirm').value;
    if (password !== confirm) { alert("Şifrələr uyğun gəlmir!"); return; }
    const reqBody = { id: 1, password: password };
    fetch('http://localhost:8080/student', {
        method: 'PUT', headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(reqBody)
    }).then(response => response.json()).then(data => {
        console.log('Success:', data);
    }).catch((error) => {
        console.error('Error:', error);
    });
}
```

Şəkil 3.7

Şəkil 3.7-dəki kod parçasında isə hücumçu, səhifədə istifadə olunan servisi işə salaraq, istifadəçinin məlumatlarını dəyişdirir. Bununla da, istifadəçi xəbərsiz olaraq səhifəsinə yenidən daxil olmağa cəhd etdikdə, şifrəsinin yanlış olduğunu öyrənir. Bu

tip hücumlar istifadəçilərin hesablarının təhlükəsizliyini ciddi şəkildə təhdid edir və diqqətli yanaşma tələb edir.

3.3. Tətbiqin müdafiə strategiyaları: Hücumlara qarşı cavab və zəifliklərin aradan qaldırılması.

Bu bölmədə, CSRF hücumlarına qarşı veb tətbiqin müdafiə strategiyaları haqqında ətraflı məlumat verəcəyik.

Eyni-Mənşə Siyasəti(Same-Origin Policy) strategiyası fərqli mənşəli sorğuları effektiv şəkildə bloklayır. Belə olan halda hücumçu fərqli mənşəli sorğudan istifadə etdiyinə görə öz istəyinə nail ola bilməyəcək. Lakin biz veb tətbiqimizin sadəcə lokal şəbəkə daxilində çalışmasını, başqaları tərəfindən əlçatan olmamasını istəmədiyimizə görə bu strategiya bizim üçün istənilən nəticəni vermir.

Çarpaz-Mənşəli Resurs Paylaşımı(CORS)-nı deaktiv edərək təhlükəsizliyi təmin edə bilərik, amma bu da yalnız lokal şəbəkələr üçün həll yoludur. Xarici mənşəli sorğuları bloklayaraq Eyni-Mənşə siyasətindən istifadə etmiş oluruq.

Eyni Sayt (Same Site) kukilərindən istifadə hücuma qarşı məlumatların müdafiəsində vacib rola malikdir. Parametrlərindən düzgün istifadə hücumun qarşısını ala bilər. SameSite=Strict parametri həssas məlumatları yalnız eyni sayt daxilində paylaşmağa icazə verir. Yəni əgər istifadəçi example.com saytındadırsa və example.com linkine klikləsə, kuki göndəriləcək, əks halda istifadəçi malicious-site.com saytındadırsa, kuki göndərilməyəcək və sorğu xəta ilə yekunlaşacaq. SameSite=Lax parametri ilə daha az həssas məlumatları göndərmək olar. Əgər istifadəçi another-website.com saytındadırsa və example.com linkinə klikləyirsə, kuki göndəriləcək. Bununla belə, other-website.com saytından example.com-a şəkil və ya AJAX sorğusu edilərsə, kuki göndərilməyəcək. SameSite=None parametrində heç bir yoxlama olmuyacaq və istənilən mənşəli sorğular üçün kuki göndəriləcək.

Fərdi http başlıqlarından istifadə hücumun qarşısını ala bilər. Məsələn, istifadəçinin keçərli olub olmamasını “User-ID” fərdi http başlığı ilə yoxlaya bilərik. Şəkil 3.8-dəki nümunəyə diqqət yetirsək əvvəlcə istifadəçi məxfi məlumatları ilə

uyğunluq yoxlanılır, istifadəçi tapılarsa "User-ID" http başlığı ilə istifadəçinin İd-sinin eyni olub olmaması yoxlanılır. Amma hücumçunun bu kimi http başlıqlarından xəbərinin olması bütün işi poza bilər, yəni yalnız fərdi http başlıqlarından istifadə etmək kifayət olmuya bilər.

```
public APIResponse loginStudent(HttpServletRequest request, String username, String password) {
    Student student = repository.findStudentByUsernameAndPassword(username, password);
    String userIdHeaderValue = request.getHeader( s: "User-ID");

    if (userIdHeaderValue != null && Integer.parseInt(userIdHeaderValue) == student.getId())
        return APIResponse.builder()
                .data(student)
                .build();
    else
        return APIResponse.builder()
                .error(APIError.builder()
                        .code(401)
                        .message("Unauthorized Attempt")
                        .build())
                .build();
}
```

Şəkil 3.8

Hər bir sessiya və istifadəçi üçün unikal və təsadüfi yaradılmış CSRF tokenlərin istifadəsi effektiv müdafiə strategiyası ola bilər. CSRF tokenləri sistemə daxil olduqdan sonra aktiv sessiya üçün kuki vasitəsilə back-end server tərəfindən front-end serverə göndərilir. Bu tokenlər, istifadəçi hər hansı bir əməliyyat həyata keçirərkən sorğu məlumatlarına əlavə olunur və server tərəfindən yoxlanılır. Əgər front-end serverdən gələn tokenlə back-end serverdən göndərilən kukidəki token eyni deyilsə əməliyat rədd edilir, beləliklə də, icazəsiz girişlərin qarşısı alınır.

Tokenlərin Java(Spring freymvork) ilə necə yaradılmasını göstərən konfiqurasiya faylı şəkil 3.9-da göstərilmişdir(Burada Spring Security freymvork-unun standart CSRF tokenindən istifadə olunmuşdur):

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    http
            .csrf() CsrfConfigurer<HttpSecurity>
            .csrfTokenRepository(CookieCsrfTokenRepository.withHttpOnlyFalse())
            .and() HttpSecurity
            .authorizeRequests() ExpressionInterceptUrlRegistry
            .antMatchers( ...antPatterns: "/student").permitAll()
            .anyRequest().authenticated()
            .and() HttpSecurity
            .formLogin() FormLoginConfigurer<HttpSecurity>
            .and() HttpSecurity
            .httpBasic();
}
```

Şəkil 3.9

Daxil olma servisini aşağıdakı kimi dəyişərək tokeni istifadəçi üçün əlçatan edirik. Burada yeni unikal token yaradılaraq istifadəçinin sessiyası üçün ona kuki daxilində göndərilir.

```
public APIResponse loginStudent(HttpServletRequest request, HttpServletResponse response,
                                String username, String password) {
    Student student = repository.findStudentByUsernameAndPassword(username, password);
    if (student != null) {
        CsrfToken csrfToken = csrfTokenRepository.generateToken(request);
        csrfTokenRepository.saveToken(csrfToken, request, response);

        Cookie cookie = new Cookie( name: "XSRF-TOKEN", csrfToken.getToken());
        cookie.setPath("/");
        response.addCookie(cookie);

        return APIResponse.builder()
                .data(student)
                .build();
    } else {
        return APIResponse.builder()
                .error(APIError.builder()
                        .code(401)
                        .message("Unauthorized Attempt")
                        .build())
                .build();
    }
}
```

Şəkil 3.10

Burada istifadəçi, istifadəçi adı və parolu ilə daxil olduqdan sonra əlavə olaraq csrf tokeni də kuki daxilində cavabda qayıdacaq.

İstifadəçi bundan sonra aktiv sessiya üçün bütün əməliyyatlarında bu tokeni http başlıq(varsayılan olaraq X-XSRF-TOKEN) vasitəsi ilə back-end serverə göndərəcək.(Şəkil 3.11). Əgər uyğunluq varsa əməliyyat icra olunacaq və ya məlumat cavabda qaytarılacaq, əks halda front-end server xəta ilə qarşılaşacaq. Bu halda biz CSRF hücumunun qarşısını almış oluruq.

```
const token = getCookie("csrfToken");
function handleSave() {
    const password = document.getElementById('password').value;
    const confirm = document.getElementById('confirm').value;
    if (password !== confirm) { alert("Şifrələr uyğun gəlmir!"); return; }
    const reqBody = { id: 1, password: password };
    fetch('http://localhost:8080/student', {
        method: 'PUT', headers: {
            'X-XSRF-TOKEN': token,
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(reqBody)
    }).then(response => response.json()).then(data => {
        console.log('Success:', data);
    }).catch((error) => {
        console.error('Error:', error);
    });
}
```

Şəkil 3.11

NƏTİCƏ VƏ TƏKLİFLƏR

Nəticə etibarilə, veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi rəqəmsal ekosistemlərin inkişaf edən kibertəhlükələrə qarşı möhkəmliyini və dayanıqlığını təmin etmək üçün mühüm əhəmiyyət kəsb edir. Diqqətli yoxlama və təhlil vasitəsilə təşkilatlar sistemlərini gücləndirə, həssas məlumatları qoruya və istifadəçilər və maraqlı tərəflər arasında etimadı inkişaf etdirə bilər. Veb proqram təhlükəsizliyinin çoxşaxəli xarakteri qiymətləndirməyə hərtərəfli yanaşma tələb edir. Bu, müxtəlif dizayn nümunələrinin yoxlanılmasını, onların effektivliyinin qiymətləndirilməsini və potensial zəifliklərin müəyyən edilməsini əhatə edir. OWASP Top 10 kimi framevorklərdən istifadə etməklə qiymətləndiricilər təhlükəsizlik tədbirlərini prioritetləşdirə və inyeksiya hücumlarından tutmuş sınmış autentifikasiyaya qədər ümumi tələləri həll edə bilərlər. Bundan əlavə, qiymətləndirmə prosesi nəzəri qiymətləndirmələrdən kənara çıxaraq praktiki həyata keçirmə və sınaqdan keçirməlidir. Penetrasiya testi, zəifliyin skan edilməsi və kodun nəzərdən keçirilməsi zəif tərəflərin aşkar edilməsində və seçilmiş təhlükəsizlik nümunələrinin effektivliyinin təsdiqində ayrılmaz rol oynayır. Real dünya ssenariləri müxtəlif hücum vektorları altında sistem davranışı ilə bağlı əvəzolunmaz fikirlər təklif edir və təşkilatlara öz müdafiələrini müvafiq olaraq dəqiqləşdirməyə imkan verir. Bundan əlavə, kibertəhlükələrin dinamik təbiəti davamlı qiymətləndirmə və uyğunlaşmanın vacibliyini vurğulayır. Bu gün effektiv olduğunu sübut edən təhlükəsizlik dizayn nümunələri sabah ortaya çıxan təhlükələr və ya texnoloji irəliləyişlər qarşısında köhnələ bilər. Buna görə də, təşkilatlar inkişaf edən riskləri azaltmaq üçün təhlükəsizlik protokollarını mütəmadi olaraq yeniləyərək sayıqlıq və çeviklik mədəniyyətini inkişaf etdirməlidirlər. Əməkdaşlıq və bilik mübadiləsi də qiymətləndirmə prosesini gücləndirir. Kibertəhlükəsizlik icması ilə məşğul olmaq təşkilatlara kollektiv təcrübədən istifadə etməyə, ən yaxşı təcrübələri bölüşməyə və sənaye inkişaflarından xəbərdar olmağa imkan verir. Forumlarda, konfranslarda və məlumat mübadiləsi təşəbbüslərində iştirak etməklə, maraqlı tərəflər yaranan təhlükələr və fəal müdafiə strategiyaları haqqında anlayışlarını gücləndirə bilərlər.

Nəhayət, veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsinin məqsədi rəqəmsal mənzərədə dayanıqlığı artırmaqdır. Təşəbbüskar və sistematik bir yanaşma tətbiq etməklə, təşkilatlar zəiflikləri qabaqcadan müəyyən edə və azalda bilər, pozuntular və məlumatların tamlıq problemi ehtimalını minimuma endirir. Bundan əlavə, güclü təhlükəsizlik tədbirləri istifadəçi inamını artırır, uzunmüddətli əlaqələri gücləndirir və biznesin inkişafına təkan verir. Əslində, veb tətbiqi təhlükəsizlik dizayn nümunələrinin qiymətləndirilməsi sadəcə texniki bir cəhd deyil, həm də strateji bir məcburiyyətdir. Təhlükəsizliyə üstünlük verməklə, təşkilatlar rəqəmsal infrastrukturlarını gücləndirə, normativlərə uyğunluğu dəstəkləyə və getdikcə bir-birinə bağlı olan dünyada öz nüfuzlarını qoruya bilərlər. Davamlı diqqətlilik və əməkdaşlıq sayəsində maraqlı tərəflər inkişaf edən təhlükə mənzərəsini inam və möhkəmliklə idarə edə bilərlər.

Təkliflər:

1.Hərtərəfli Riskin Qiymətləndirilməsi: Veb tətbiqə və onun mühitinə xas olan potensial təhdidləri və zəiflikləri müəyyən etmək üçün hərtərəfli risk qiymətləndirməsini aparmaq. Bu qiymətləndirmə tətbiqin tələblərinə və potensial risklərə uyğunlaşdırılmış müvafiq təhlükəsizlik dizayn nümunələrinin seçilməsi üçün əsas kimi xidmət edir.

2.Standartlaşdırılmış Nümunələrin Qəbul Edilməsi: İdentifikasiya və avtorizasiya mexanizmləri, girişin yoxlanılması, çıxışın kodlaşdırılması və sessiyanın idarə edilməsi kimi standartlaşdırılmış təhlükəsizlik dizayn nümunələrinin qəbulunu vurğulamaq. Bu nümunələr kibertəhlükəsizlik icması tərəfindən geniş şəkildə yoxlanılıb və ümumi təhlükəsizlik problemlərinə sübut edilmiş həllər təklif edir.

3.Daimi Təhlükəsizlik Auditləri və Testləri: Seçilmiş təhlükəsizlik dizayn nümunələrinin effektivliyini təsdiqləmək üçün müntəzəm təhlükəsizlik auditləri və sınaq prosedurlarını həyata keçirmək. Bura penetrasiya testinin aparılması, zəifliyin skan edilməsi və təhlükəsizlik zəifliklərini proaktiv şəkildə müəyyən etmək və aradan qaldırmaq üçün kod təhlili daxildir.

4.Təhsil və Təlim: Təhlükəsizliyin dizayn nümunələrinin əhəmiyyəti və onların düzgün həyata keçirilməsi haqqında tərtibatçılara təhsil və təlim vermək. Təhlükəsiz

kodlaşdırma təcrübələrinə riayət etməyi təşviq etmək və ən son təhlükəsizlik tendensiyaları, texnikaları və nümunələri haqqında öyrənmək üçün resurslar təklif etmək ən yaxşı təlim təcrübələrindəndir.

5.İcma İştirakı və Əməkdaşlıq: Yaranan təhlükələr, zəifliklər və ən yaxşı təcrübələr haqqında məlumatlı olmaq üçün kibertəhlükəsizlik icması ilə əlaqə saxlamaq. Fikir mübadiləsi aparmaq və veb tətbiqi təhlükəsizliyi dizayn nümunələrinin təkmilləşdirilməsində həmyaşıdları ilə əməkdaşlıq etmək üçün forumlarda, konfranslarda və bilik mübadiləsi platformalarında iştirak etmək.

İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT SİYAHISI

Acar Y. “Web Uygulama Güvenliğinde Uzmanlaşmak: Kapsamlı Bir Kılavuz”, Bursa, “Vipaş Yayınları”, 2020, 485 s.

Ambrosıanı B. “Web Uygulaması Güvenliği Esasları”, İstanbul, “Alfa”, 2015, 392 s..

Atalay İ. “Web Uygulaması Güvenliği El Kitabı: En İyi Uygulamalar ve Teknikler”, İstanbul, “İnkılap Kitabevi”, 2016, 378 s.

Barnes M.P. “Pratik Web Uygulama Güvenliği”, İstanbul, “Alfa”, 2015, 352 s.

Bauduin P. “Güvenli Web Uygulamaları Oluşturma”, Ankara, “Dost Kitabevi”, 2019, 400 s.

Benevolo L. “Web'in Güvenliğini Sağlama”, İstanbul, “Afa Yayıncılık”, 2016, 413 s.

Bennett M. “Web Uygulaması Güvenliği Esasları: Dijital Varlıklarınızı Korumak”, İstanbul, “Timaş Yayınları”, 2011. 456 s.

Bill Jan. “Bulut Çağında Web Uygulama Güvenliği”, İstanbul, “Alfa”, 2015, 226 s.

Brınk S. “WASP İlk 10'un Ötesinde: Web Uygulama Güvenliğinde Ortaya Çıkan Tehditler”, Alfa, “İstanbul”, 2017, 300 s.

Chapple M., Seidl D. “CISSP Official (ISC)2 Practice Tests”. Indianapolis, “Wiley Publishing”, 2020, 630 p.

Charles C.Mann., Susan L.H. “Information Security and Ethics”, “Iowa State University” 2022, 504 p.

Clark D., Wilson D. “A Hands-On Introduction to Forensic Science: Cracking the Case”, Boca Raton, “CRC Press”, 2018, 400 p.

Darril G. “Web Application Security Handbook”, Berlin, “Springer”, 2020, 473 p.

David A.C. “Computer Security: A Hands-On Approach”, London, “Routledge”, 2022, 425 p.

David J. P. “Information Security: A Manager's Guide”, New York, “Wiley-Interscience”, 2021, 419 p.

David L.E. “Information Security: The Complete Reference”, Cambridge, “MIT Press”, 2019, 531 p.

David M.Kroll., Timothy W.L. “Web Application Security 101”, Boston, “Cengage Learning”, 2017, 576 p.

Dhillon G., Backhouse J. “Web Application Security Governance”, New York, “Wiley Publishing”, 2020, 500 p.

Eloff, M., & Eloff, J. “Advanced Persistent Threats: Defending Web Applications”, Boca Raton, “CRC Press”, 2019, 400 p.

Gollmann D. “Computer Security”, Chichester, “Microsoft Press”, 2016, 300 p.

Hacking I. “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy”, San Francisco, “No Starch Press”, 2017, 440 p.

Harold F.T., Micki K. “Web Application Security”, New York “IEEE Signal Processing Magazine”, 2018, 588 p

Qıllıcam S. “Information Security: Concepts and Practice”, New York, “Wiley Publishing”, 2023, 521 p

Mark J.C. “Mastering Web Application Security”, Sydney, “Waterwise Publishing”, 2020, 480 p.

Menzel C., Waidner M. “Secure Internet Programming: Security Issues for Mobile and Distributed Objects”, Berlin, “Springer”, 2021, 566 p.

Peltier T. “Web Application Security Essentials”, Boca Raton, “CRC Press”, 2017, 631 p.

Peter K. “The Hacker Playbook: A Practical Guide to Penetration Testing”, New York, “Garden Press”, 2020, 453 p.

Peter T. “Information Security: A Business Perspective”, New Jersey, “Wiley-Blackwell”, 2022, 414 p.

Pfleeger C., Pfleeger S., Margulies J. “Security in Computing”, Boston, “Pearson”, 2017, 588 p.

Radack S., DeBar H. “Computer Security Basics”, Sebastopol, “O'Reilly Media”, 2019, 389 p.

Schneier B. “Secrets and Lies: Digital Security in a Networked World”, New York, “Wiley Publishing”, 2020, 450 p.

Simson G.S. “Web Security, Privacy & Commerce”, Sebastopol, “O'Reilly Media”, 2021, 570 p.

Stallings W. “Threat Modeling for Web Applications”, Boston, “Pearson”, 517 p.

Stamp M. “The Art of Secure Coding: Principles for Web Applications”, Hoboken, “Que”, 2019, 533 p.

Stuart McC., Joel S., George K. “Threat Modeling for Web Applications”, Boston, “CRC Press”, 2020, 502 p.

Thomas P.C. “Cybersecurity for Cloud Computing: Securing the Cloud from End to End”, Chicago, “CropTech Publications”, 2021, 462 p.

Thomas P.C. “Cybersecurity for IoT: Securing the Internet of Things”, New Jersey, “Time Publishing”, 2020, 547 P.

Whitman M., Mattord H. “Principles of Information Security”, Boston, “Cengage Learning”, 2019, 428 p.

https://brightsec.com/blog/8-types-of-web-application-attacks-and-protecting-your-organization/

https://brightsec.com/blog/web-application-security/

https://dev.to/sardarmudassaralikhan/design-patterns-and-their-benefits-2ngk

https://hal.science/hal-01334292/document

https://inria.hal.science/hal-00794514/document

https://insights.sei.cmu.edu/documents/813/2009_005_001_15110.pdf

https://learn.microsoft.com/en-us/azure/well-architected/security/design-patterns

https://owasp.org/www-community/vulnerabilities/

https://owasp.org/www-project-top-ten/

https://safedesignaustralia.com.au/what-is-safe-design/

https://securelist.com/top-10-web-app-vulnerabilities/112144/

https://tectrain.ch/en/top-10-software-architecture-patterns

https://www.azion.com/en/learning/websec/common-web-application-security-threat/

https://www.checkpoint.com/cyber-hub/cloud-security/what-is-application-security-appsec/owasp-top-10-vulnerabilities/

https://www.defense.com/blog/secure-by-design

https://www.dmp.wa.gov.au/Safety/What-is-safe-design-4794.aspx
https://www.fortinet.com/resources/cyberglossary/web-security-threats
https://www.freecodecamp.org/news/the-basic-design-patterns-all-developers-need-to-know/
https://www.geeksforgeeks.org/top-10-security-risks-in-web-applications/
https://www.gmihub.com/blog/benefits-of-secure-software-architecture/
https://www.loginradius.com/blog/identity/7-web-app-sec-threats/
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8296130/
https://www.redscan.com/news/a-guide-to-the-owasp-top-10-web-application-security-risks/
https://www.safetyartisan.com/2023/06/07/safe-design-in-australia/
https://www.safeworkaustralia.gov.au/safety-topic/managing-health-and-safety/safe-design/overview
https://www.sitesafe.org.nz/globalassets/guides-and-resources/health-and-safety-guides/safetyindesigninconstructionguide.pdf
https://www.stackhawk.com/blog/10-web-application-security-threats-and-how-to-mitigate-them/
https://www.toptal.com/cybersecurity/10-most-common-web-security-vulnerabilities
https://www.trendminer.com/security-by-design-in-operational-software/
https://www.veracode.com/security/owasp-top-10