AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

YÜKSƏK TƏHSİL İNSTİTUTU

Əlyazması hüququnda

Qədir Məmmədli

Aysel Məmmədova

Samir İsrafilov

Gülər Əzimli

Ruslan Qələndərli

VEB SAYTLARIN TƏHLÜKƏSİZLİYİNİN TƏHLİLİ ÜÇÜN DEVOPS VASİTƏSİLƏ PROQRAM TƏMİNATININ İŞLƏNMƏSİ

mövzusunda

MAGİSTRLİK DİSSERTASİYASI

İxtisas: **060631- “ Kompüter mühəndisliyi”**

İxtisaslaşma**: “Biliklərin əldə edilməsi sistemləri”**

Elmi rəhbər: **t.e.n., dos. X. N. RZAYEV**

BAKI – 2024

Mündəricat

İXTİSARLARIN SİYAHISI

İT	Information Technology – *İnformasiya texnologiyaları*
FTP	File Transfer Protocol – *Fayl ötürmə protokolu*
HTTP	Hypertext Transfer Protocol – *Hipermətn ötürmə protokolu*
CSS	Cascading Style Sheet – *Kaskad üslub cədvəli*
HTML	HyperText Markup Language – *Hiper mətn işləmə dili*
URL	Uniform Resource Locator – *Vahid resurs axtarıcısı*
CMS	Content Management System – *Məzmun idarəetmə sistemi*
SSL	Secure Socket Layer – *Təhlükəsiz socket qatı*
DdoS	Distributed Denial of Service – *Paylanmış xidmətdən imtina*
SQL	Structured Query Language – *Strukturlaşdırılmış sorğu dili*
XSS	Cross-Site Scripting – *Skript kodları*
SFTP	Secure File Transfer Protocol – *Təhlükəsiz fayl transfer protokolu*
PCI	Peripheral component interconnect – *Periferiya komponentlərinin qarşılıqlı əlaqəsi*
MFA	Multi-Factor Authentication – *Çox faktorlu doğrulama*
WAF	Web Application Firewall – *Veb tətbiqi təhlükəsizlik divarı*
ASP	Active Server Pages – *Veb səhifələrin kodlanması üçün əlifba*
SOAP	Simple Object Access Protocol – *Sadə operator giriş protokolu*
CI	Continuous Integration **–** *Davamlı inteqrasiya*
CD	Continuous Delivery **–** *Davamlı çatdırılma*
DOKS	Digital Ocean Kubernetes – *İdarə olunan kubernetes xidməti*
CLI	Command Line İnterface – *Komanda xətti interfeysi*
DNS	Domain Name System – *Domen ad sistemi*
WWW	World Wide Web – *Ümumdünya hörümçək toru*
URI	Uniform Resource Identifier – *Resursun universal identifikatoru*

GİRİŞ

Veb saytınızı işə saldıqdan və onun uğur qazanacağına əmin olmaq üçün bütün lazımi addımları atdıqdan sonra veb saytın təhlükəsizliyini unutmuş ola bilərsiniz. Kiberhücumlar lazımi təhlükəsizlik tədbirləri görülməyən veb saytlar üçün kifayət qədər geniş yayıla bilər və onların təmizlənməsi, brendinizə zərər vurması və istifadəçiləri geri qayıtmaqdan çəkindirmək baha başa gələ bilər.

Xoşbəxtlikdən, güclü veb sayt təhlükəsizliyi ilə hamısını dayandıra bilərsiniz. Biz veb sayt təhlükəsizliyinin nə olduğunu və hackerin veb saytınızı hücum edə bilməyəcəyinə əmin olmaq üçün həyata keçirə biləcəyiniz təminatları nəzərdən keçirəcəyik.

Veb saytdakı məlumatların kibercinayətkarlar üçün əlçatan olmamasını təmin etmək və ya veb saytın hər hansı şəkildə istismarını dayandırmaq üçün həyata keçirilən hər hansı tədbir və ya proqram veb sayt təhlükəsizliyi adlanır. Bu addımları atmaqla vebsaytın həssas məlumatları, aparatı və proqram təminatı indi mövcud olan çoxsaylı hücumlardan qorunur. Bu Dissertasiya işində veb sayt təhlükəsizliyinin qorunmasında olan nüanslar, proqramlar, kodlar göstəriləcək. (Allan Joaquin, 2023)

I FƏSİL . Veb sayt təhlükəziliyinin növləri və İT sferasında rolu

1.1 Veb Sayt və onun növləri

Şək.1.1.Veb saytın təsəvvürü

(https://aexpan.com.tr/upload/icerik/21127ae.webp)

İngilis alimi Tim Berners-Li 1989-cu ildə CERN-də işləyərkən Ümumdünya Şəbəkəsini (WWW) yaratdı. İnternetin yaradılmasının əsas ideyası dünyanın hər yerində tədqiqat mərkəzlərində və universitetlərdə çalışan alimlər arasında avtomatlaşdırılmış məlumat mübadiləsi ehtiyacını ödəmək idi. (Cillium Hubble, 2024) Tim 1990-cı ilin oktyabrına qədər müasir internetin əsasını təşkil edən (və veb-brauzerinizdə müəyyən yerlərdə göründüyünü görmüsünüz) üç əsas texnologiyanı yazdı:

HyperText Markup Language və ya HTML internetin işarələmə (formatlaşdırma) dilidir.Vahid Resurs İdentifikatoru və ya URI hər bir onlayn saytı müəyyən etmək üçün istifadə edilən xüsusi bir "ünvan" növüdür. Bunun başqa bir tez-tez isitifadə olunan adı

URL-dir.Hypertext Transfer Protocol və ya HTTP əlaqəli olan onlayn materialların axtarışına imkan verir (Sirr Tim Berners-Lee, 2014).

Veb sayt HTML (HyperText Markup Language) ilə yazılmış rəqəmsal fayllar olan bir neçə veb səhifədən ibarətdir. İnsanların dünyanın istənilən yerindən ona daxil ola bilməsini istəyirsinizsə, veb saytınız həmişə onlayn olan kompüterdə yerləşdirilməlidir və ya saxlanılmalıdır. Bu cihazlara veb serverlər deyilir (Khan İmaran, 2022).

Veb saytın veb səhifələri ümumi interfeysə və dizayna malikdir. Burada hiperlinklər hipermətnlə əlaqələndirilir. Şəkillər, filmlər və digər rəqəmsal aktivlər kimi əlavə fayllar və sənədlər də veb saytda ola bilər.

İnternetin həyatımızın hər sahəsinə nüfuz etməsi nəticəsində biz müxtəlif səbəblər və məqsədlər üçün veb saytları görürük. Dizayn edildiyi təşkilatın məqsədlərinə çatmaq üçün biz veb saytı insanlar, yerlər və əşyalar arasında qarşılıqlı əlaqəni təşviq edərkən məlumat və həllər təqdim edə bilən rəqəmsal mühit kimi də müəyyən edə bilərik.

Veb-saytın komponentləri: Biz bilirik ki, veb sayt veb serverdə yerləşdirilən veb-səhifələrin toplusudur. Bunlar veb sayt yaratmaq üçün komponentlərdir.

Webhost: Hosting, veb saytın fiziki olaraq yerləşdiyi yerdir. Yalnız veb səhifə veb serverdə yerləşdirildikdə veb sayt adlandırılmaq üçün lisenziyalı veb səhifələr qrupu (əlaqəli veb səhifələr) olmalıdır. Veb server, veb saytın ünvanını təyin etdikdə istifadəçi kompüterlərinə ötürülən fayllar toplusudur.

Ünvan: Veb-saytın ünvanı, bəzən onun URL-i kimi istinad edilir. İstifadəçi veb saytı açmaq üçün veb saytın ünvanını və ya URL-ni veb brauzerinə daxil etməlidir; sonra veb-server tələb olunan veb səhifəni çatdırır.

Əsas səhifə: Veb səhifənin əsas səhifəsi tez-tez istifadə olunan və əhəmiyyətli komponentdir. Ziyarətçi veb saytına daxil olduqda, ilk gördükləri bu səhifədir. Veb saytın ana səhifəsi çox vacibdir, çünki o, saytın ümumi üslubunu yaradır və ziyarətçiləri digər səhifələrə yönəldir.

Dizayn: Veb saytın son görünüşü və hissi qrafika, tərtibat, naviqasiya menyuları və s. kimi bir çox aspektlərin effektiv istifadəsi və inteqrasiyası ilə müəyyən edilir.

***Məzmun**:* Veb saytın bütün veb səhifələri onun məzmununu təşkil edir. Yaxşı yazılmış məzmunu olan veb sayt vizual olaraq daha cəlbedici və təsirli olur.

***Naviqasiyanın təşkili**:* Veb saytın naviqasiya strukturu onun səhifələrinin düzülüşü və onlar arasında keçidlərin toplanması ilə müəyyən edilir. O, adətən bir və ya bir neçə naviqasiya menyusu ilə bir yerdə saxlanılır (Khan İmaran, 2022).

Veb saytlar necə işləyir?

Hər bir veb saytın özünə məxsus ünvanı var və brauzerlərə onun yeri haqqında məlumat verir. Sadə dillə desək, siz veb saytınıza daxil olub onlayn olduğunuz zaman kompüter brauzerdən səhifəni götürür. File Transfer Protocol (FTP) Məlumatların fayllarının internet sürəti və necə idarə edilməsi əsaslı şəkildə təsvir edilir və Hypertext Transfer Protocol (HTTP) kimi texnologiya bu məlumatı əldə etmək üçün veb xidmət tərəfindən istifadə olunur. Brauzer daha sonra əldə edilmiş veb səhifəni ekranınızda göstərmək üçün Cascading Style Sheet (CSS) və HyperText Markup Language (HTML) kimi texnologiyalardan istifadə edir. Brauzerə ekranda müəyyən elementlərin harada yerləşdirilməsi barədə dəqiq göstəriş verməklə, onlar veb səhifənin məzmununu təşkil edir və istifadəçiyə göstərirlər. (Marcus Fields, 2024)

Veb saytın əsas komponentlərini başa düşmək onun texnologiyası haqqında anlayışınızı yaxşılaşdıracaq. Bundan əlavə, öz veb saytınızı qurmaq, veb sayt komponentləri haqqında möhkəm bir anlayışınız varsa, vaxtınıza və pulunuza qənaət edəcəksiniz.

- ❖ Veb sahibi

Qısaca desək, veb hostinq veb sayt sahiblərinə veb saytlarını onlayn yaratmaq, idarə etmək və dərc etmək üçün tələb olunan alətləri verdiyi prosesdir. Əslində veb saytınızın onlayn baxılması üçün onun faylları və məzmunu fiziki olaraq veb server kimi tanınan güclü kompüterdə saxlanmalıdır.

Hostinger-in unikal və istifadəçi dostu idarəetmə paneli hPanel veb saytın idarə edilməsini sadələşdirmək üçün idarə olunan veb hostinq planlarına inteqrasiya olunub.

Tək bir tablosunda siz başqa resurslar arasında faylları yükləyə, resurslarınıza nəzarət edə və ehtiyat nüsxələri idarə edə biləcəksiniz (Akshay Kedari, 2024).

- ❖ Domen

Domen adı, istifadəçilərin internetdə veb sayta daxil olmaq üçün istifadə etdikləri xüsusi addır. Yadda qalan domen adı istifadəçilərin veb saytınızı yadda saxlamasını asanlaşdırır. Buna görə də, uyğun domen adının seçilməsi veb sayt yaratmaq prosesi üçün çox vacibdir.

- ❖ URL

İnternetdəki hər hansı unikal resursun URL (vahid resurs yeri) kimi tanınan virtual ünvanı var. Veb səhifə, şəkil, video və ya hətta kağız bu mənbə ola bilər. URL adətən yol, genişləndirmə, protokol və domen adı kimi bir neçə komponentdən ibarətdir.

- ❖ Dizayn

Vizual cəhətdən cəlbedici və yaxşı işləyən veb sayt dizaynı istifadəçiləri daha çox araşdırmağa və geri qayıtmağa sövq edə bilər. Əslində, istifadəçilərin 80%-dən çoxu şirkətin veb dizaynı ilə bağlı onun etibarlılığı haqqında fikirlərini əsas götürür.

Buna görə də, hər hansı bir onlayn biznes varlığının ən vacib komponenti hərtərəfli veb sayt dizaynıdır. Artıq, əvvəlcədən çox pul xərcləmədən veb sayt dizaynı yaratmağın bir neçə yolu var. Veb sayt qurucusunda dizaynerlərdən əvvəlcədən hazırlanmış mövzulardan istifadə bir alternativdir.

- ❖ Mətn

İstifadəçiləri istiqamətləndirmək, məlumatlandırmaq və ya inandırmaq üçün veb saytınızda təqdim etdiyiniz material məzmun kimi tanınır. Ziyarətçilərə bu məlumat mətn, şəkillər, audio fayllar və ya videolar vasitəsilə göstərilə bilər (Akshay Kedari, 2024).

Veb saytlara hucum üsulları

Veb saytı kiberhücumlardan qorumaq üçün görülən addımlar veb sayt təhlükəsizliyi adlanır. Bu, veb saytın viruslardan, fişinq fırıldaqlarından, hakerlərdən və səhvlərdən qorunmasını tələb edə bilər. Beləliklə, veb saytın təhlükəsizliyini qorumaq davamlı səy tələb edir və veb sayt idarəçiliyinin mühüm komponentidir.

Təhlükəsiz veb saytın təmin edilməsi istifadəçiləri və ziyarətçiləri zərərli aktorlardan, məlumat oğurluğundan və təhdidlərdən qorumaq üçün vacibdir. (Maricheva Alena, 2023).

Veb saytlarda təhlükəsizlik niyə vacibdir?

Xüsusilə böyük veb saytlar şəbəkəsini idarə edərkən veb sayt təhlükəsizliyini qorumaq çətin ola bilər. Birinin onlayn olması üçün təhlükəsiz veb sayta sahib olmaq veb sayt sahibi olmaq qədər vacibdir. Bir veb sayt, məsələn, sındırıldıqda və bloklandıqda, trafikinin 98%-ni itirə bilər. Məsələn, müştəri məlumatlarının pozulması bahalı cərimələrə, hüquqi tədbirlərə və reputasiyaya zərər verə bilər (Marcus Fields, 2024).

Şək.1.2. Sayt-da istifadəçinin daxil olması (https://psystems.co/wp-content/uploads/2023/09/aaa6219225.jpg)

Dərinlik Strategiyasının Müdafiəsi

Yığın boyunca istifadə olunan alətləri araşdırmaq üçün veb sayt təhlükəsizliyi üçün dərin müdafiə yanaşması hücum səthinin həm genişliyini, həm də dərinliyini

nəzərə alır. Bu üsul veb sayt təhlükəsizliyi üçün mövcud təhlükə mənzərəsinin daha real görünüşünü təklif edir.

1.2 Veb saytlara niyə hücumlar olur?

2019-cu ildə onlayn olaraq 1,94 milyarddan çox veb sayt var. Bu, pis aktyorlar üçün geniş oyun meydançası təqdim edir.

Tez-tez veb saytların niyə sındırıldığı ilə bağlı yanlış fikir var. Sahiblər və idarəçilər tez-tez saytları daha kiçik olduğu üçün sındırılmayacaqlarına inanırlar və buna görə də daha az cəlbedici hədəflər qoyurlar. Hakerlər məlumat oğurlamaq və ya təxribat etmək istəsələr, daha böyük saytlar seçə bilərlər. Onların digər məqsədləri üçün (daha çox yayılmışdır) istənilən kiçik sayt kifayət qədər dəyərlidir (Marcus Fields, 2024).

Veb saytlara avtomatlaşdırılmış hücumlar

WordPress, Magento, Joomla və ya Drupal kimi açıq mənbəli məzmun idarəetmə sistemləri (CMS) tipik veb sayt sahibinin sürətlə onlayn olmasını asanlaşdırdı.

Bu sistemlər tez-tez təhlükəsizlik yeniləmələrini buraxsa da, plaginlər və ya mövzular kimi genişləndirilə bilən üçüncü tərəf komponentlərinin istifadəsi fürsət hücumları tərəfindən asanlıqla istifadə edilən zəifliklər yaradır. Veb sayt sahiblərinə ətraflarını qorumaqda və riskləri azaltmaqda kömək etmək üçün hər bir tanınmış məzmun idarəetmə sistemi üçün hərtərəfli təhlükəsizlik qaydaları yaratdıq.

Dürüstlük son istifadəçilərin əldə etdiyi məlumatların doğru olduğuna və veb sayt sahibindən başqa heç kim tərəfindən dəyişdirilməməsinə zəmanət verir. Bunun üçün məlumatların tranzitdə şifrələnməsinə zəmanət verən Secure Socket Layer (SSL) sertifikatları kimi şifrələmə tez-tez istifadə olunur.

Üçlüyün son komponenti məlumatların lazım olduqda əlçatan olmasına zəmanət verən mövcudluqdur. Paylanmış xidmətdən imtina hücumları kimi də tanınan DDoS hücumları veb saytların mövcudluğuna ən çox rast gəlinən təhlükədir.

SQL inyeksiya hücumuna başlamaq üçün zərərli kod zəif SQL sorğusuna yeridilir. Onlar təcavüzkarın vebsaytın verilənlər bazasına göndərdiyi mesajın içərisinə xüsusi

hazırlanmış sorğu daxil etməsindən asılıdır. Hücum uğurlu olarsa, verilənlər bazası sorğusu veb saytın gözlədiyindən daha çox təcavüzkarın nəzərdə tutduğu məlumatı çatdırmaq üçün dəyişdiriləcək. Zərərli məlumatlar hətta SQL inyeksiyaları vasitəsilə verilənlər bazasına əlavə edilə və ya dəyişdirilə bilər. Zərərli müştəri tərəfi skriptləri veb saytları yayma vasitəsi kimi istifadə edən saytlararası skript hücumlarının bir hissəsi kimi veb saytlara yeridilir.

Saytlararası skript (XSS) ilə bağlı risk ondan ibarətdir ki, o, hakerə vebsayta məzmun daxil etməyə və onun necə göründüyünü dəyişdirməyə imkan verir, səhifə yüklənərkən qurbanın brauzerindən təcavüzkarın kodunu işlətməsini tələb edir. Skript, daxil olmuş sayt administratoru kodu yükləsə, ona təyin edilmiş imtiyaz səviyyəsi ilə işləyəcək və bu, saytın ələ keçirilməsi ilə nəticələnə bilər.

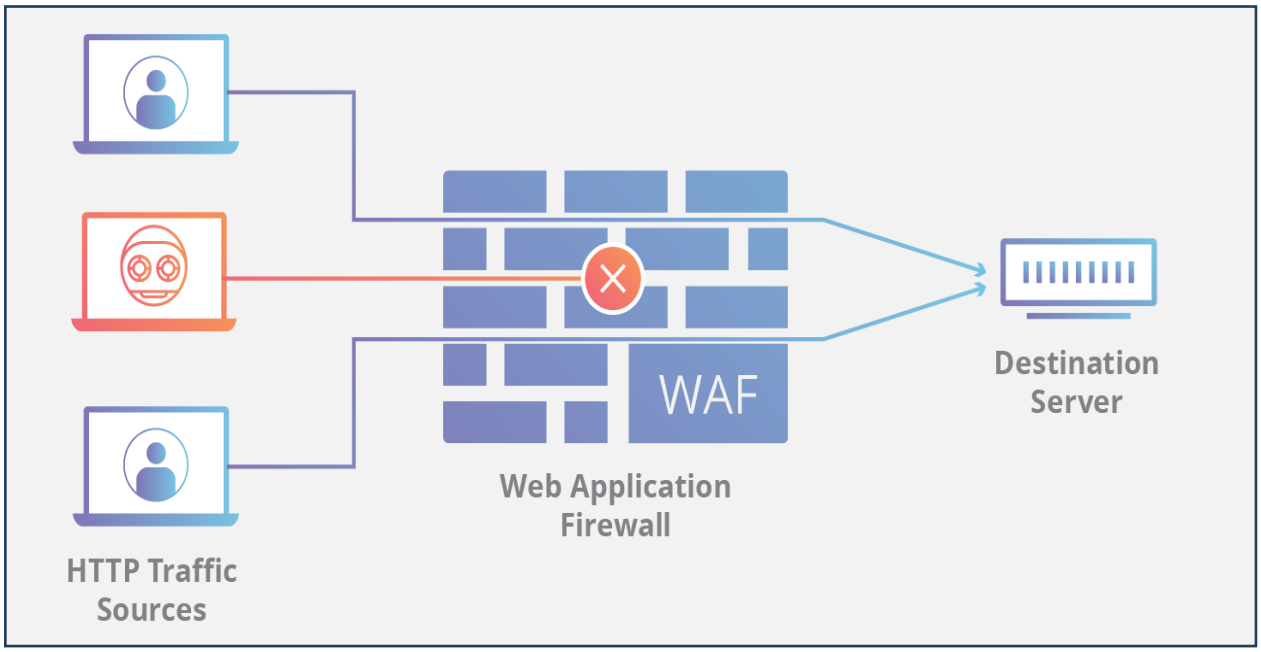
Veb saytın idarəetmə panelinə, admin sahəsinə və ya hətta SFTP serverinə daxil olmaq veb-saytların oğurlanmasının ən məşhur yollarından biridir. Təcavüzkarlar, uğur qazanana qədər müxtəlif istifadəçi adı və parol birləşmələrinə cəhd etmək üçün skript yazır, bu, kifayət qədər sadə bir prosesdir. Giriş əldə edildikdən sonra təcavüzkarlar kredit kartı oğurluğu, sikkə mədənçiliyi və spam kampaniyaları kimi geniş çeşidli çirkin əməliyyatlara başlamaq imkanı əldə edirlər (Marcus Fields, 2024).

Şək.1.3. Brute Force Hücum
(https://miro.medium.com/v2/resize:fit:1100/format:webp/1*yaTYhmeCgtx9H4sailcmmA.jpeg)

DDOS

Təhlükəli olmayan internet hücumu Paylanmış Xidmətdən imtina (DDoS) hücumu kimi tanınır. O, hədəflənmiş veb saytı aşağı salmaq və ya yavaşlatmaq cəhdi ilə şəbəkəni, serveri və ya proqramı uydurma trafiklə həddən artıq yükləmək üçün nəzərdə tutulmuşdur.

Təhlükəsizlik mühitindəki mühüm roluna görə, DDoS hücumları veb sayt sahiblərinin bilməli olduğu bir problemdir. DDoS hücumu həssas olan resurs-intensiv son nöqtəni hədəf aldıqda çox az trafiklə belə uğur qazana bilər (Shikhar Goel, 2023).



Şək.1.4. Veb Applikasiya Divarı
(https://miro.medium.com/v2/resize:fit:1400/format:webp/0*CA61sfjS9wmLjA-X.png)

Mühafizə

Profilaktik sayt təhlükəsizlik tədbirləri çoxsaylı səbəblərə görə vacibdir, lakin siz haradan başlayırsınız? Bunlara dərindən müdafiə və qoruyucu texnologiyalar deyilir.

Bəzən bu addımlar PCI uyğunluq tələblərini yerinə yetirir və ya hücuma meyilli mühitlərin virtual yamaqlarını və sərtləşdirilməsini asanlaşdırır. Girişə nəzarət və işçilərin təlimi ilə bağlı siyasətlər mühafizənin əlavə formalarıdır.

Veb tətbiqi təhlükəsizlik duvarını işə salmaq veb saytınızı qorumaq üçün ən yaxşı yollardan biridir. Təhlükəsizlik prosedurlarını, resursları və quraşdırmaları nəzərdən keçirmək üçün vaxt ayırmaq onlayn təhlükəsizlik duruşunuza təsir edəcək.

"Davamlı monitorinq" termini veb saytınızın aktivlərini izləmək və hər hansı problem barədə sizi xəbərdar etmək üçün alətlərdən istifadəni təsvir edir (Deeksha Karkera, 2024).

Parollara əsaslanan hücumlar

Onlar "sınmış autentifikasiya" hücumunda istifadə oluna bilsələr də, bunlar öz məkanlarına layiqdirlər. Parol əsaslı hücumların diapazonu və müxtəlifliyi aşağıdakılardır:

- Etibarnamənin dempinqi: sirrlərinizə daxil olmaq üçün kimsə RAM-ı oğurladıqda
- Brute force: düzgün parolu tapmaq üçün metodik yanaşma

Etibarnamənin doldurulması məlum etimadnamələri ilə bir neçə fərqli hesaba daxil olmaq təcrübəsidir.

Pass the Hash (PtH) üsullarından istifadə edərək, hashed etimadnaməsini oğurlayaraq yeni səlahiyyətli sessiya qura bilərsiniz.

Şifrə əsaslanan hücumların ehtimalı kod imzalanmasının həyata keçirilməsi, güclü parol tələblərinin tətbiqi, MFA-nın konfiqurasiyası və ən az imtiyaz prinsipinə riayət etməklə azaldıla bilər (Deeksha Karkera, 2024).

Buzlanma

"Qeyri-səlis test" kimi tanınan bir növ onlayn hücum, əvvəlcə proqramın qəzaya uğramasına səbəb olmaq üçün çoxlu təsadüfi məlumat (fuzz) ilə doldurulmasını nəzərdə tutur. Zəif yerləri tapmaq üçün fuzzer proqram alətindən istifadə etmək növbəti addımdır. Təcavüzkar hədəfin təhlükəsizliyindəki zəif cəhətlərdən daha da istifadə edə bilər. Təhlükəsizliyiniz və digər proqramların ən son versiyalarını saxlamaq qeyri-səlis hücumlara qarşı ən yaxşı müdafiədir. Əgər yeniləməni hələ tətbiq etməmisinizsə, bu xüsusilə təcavüzkarların sizə qarşı istifadə edə biləcəyi yeniləmə ilə buraxılan hər hansı təhlükəsizlik yamaları üçün doğrudur (Thompson Ken, 2024).

1.3 Veb Saytın qorunma üsulları

Hakerlərin həssas məlumatlara daxil olmasını dayandırmaq və korporativ veb saytların təhlükəsizliyini qorumaq üçün veb sayt və veb tətbiqi təhlükəsizliyinin zəruriliyi bütün dünyada böyük təşkilatlara, banklara, federal dövlət qurumlarına və minlərlə digər müəssisələrə təsir edən veb sayt təhlükəsizliyinin son və davam edən pozuntuları ilə bir daha təsdiqləndi. Veb tətbiqi təhlükəsizliyi bir çox insanın mübarizə apardığı bir mövzu olsa da, öyrənilməsi çətin bir sahədir.

Bəzi təhlükəsizlik mütəxəssislərinin pis niyyətli istifadəçilərin veb proqramınızdan sui-istifadə etməsinin qarşısını almaq üçün tələb olunan hər cür təhlükəsizlik tədbiri və tədbiri təmin etməsi qeyri-mümkün olardı. Təcavüzkarın veb saytınızı pozmaq üçün istifadə edə biləcəyi minlərlə potensial qüsuru olan veb tətbiqi təhlükəsizliyinin mürəkkəb bir anlayış olması məntiqlidir. Bununla belə, bir çox veb tərtibatçıları veb saytlarını qorumağa çalışarkən bir şəkildə mübarizə aparacaqlar.

Artıq qeyd etdiyimiz kimi, veb saytınızı qorumaq çox sadədir. Kömək etmək üçün altı ideya və ya proseduru qeyd etdik. Bu güllə nöqtələri başlanğıc nöqtəsi və veb tətbiqi təhlükəsizliyinin müəyyən aspektlərinin digərlərinə nisbətən daha təcili təmin edilməli olduğuna dair incə xatırlatma rolunu oynayacaq (Shikhar Goel, 2023).

- **Hostinq Seçimləri**

Əksər saytlar veb hostinq xidmətləri olmadan mövcud olmazdı. Veb tətbiqetmələrini yerləşdirməyin ən çox yayılmış iki yolu paylaşılan hostinqdir, bunda veb serveri öz veb proqramlarını idarə etmək üçün istifadə edəcək digər istifadəçilərlə paylaşırsınız və veb proqramınızın xüsusi serverdə yerləşdiyi müntəzəm hostinq nəzərdə tutulur.
Paylaşılan hostinqin bir sıra üstünlükləri var. Hostinq sahəsini bölüşməyi üstün tutan kiçik müəssisələr adətən bu seçimə cəlb olunurlar, çünki bu, öz xüsusi serverinizə sahib olmaqdan daha ucuzdur. Funksional nöqteyi-nəzərdən, paylaşılan və xüsusi hostinq arasındakı fərq əhəmiyyətli görünməyəcək, çünki veb sayt işləməyə davam edəcək; buna baxmayaraq, təhlükəsizlik məsələsinə gəldikdə, biz məsələyə çox fərqli yanaşmalıyıq.

Paylaşılan hostinqin mənfi tərəfi onun təklif edə biləcəyi üstünlüklərdən üstündür. Veb server çoxsaylı veb proqramlar arasında paylaşıldığı üçün istənilən hücumlar da onlar arasında paylaşılacaq. Məsələn, veb serverinizi veb saytında xidmətdən imtina hücumları həyata keçirmiş təcavüzkarlar tərəfindən hədəfə alınmış bir təşkilatla paylaşsanız, veb proqramınız da həmin serverdə yerləşdirildiyi üçün təsirlənəcək.

Bu arada, provayder onlayn proqramınızı hücuma məruz qoya biləcək qərarlar qəbul edə biləcək, çünki onlar veb server üzərində tam nəzarətə malik olmayacaqlar. Paylaşılan serverdə yerləşən veb saytlardan biri təhlükə altına düşərsə, bütün digər veb saytlar, eləcə də veb serverin özü həssas ola bilər (Shikhar Goel, 2023).

- **Kod baxışlarının yerinə yetirilməsi**

Veb tətbiqlərinə edilən təsirli hücumların böyük əksəriyyəti əsas platformanın özü tərəfindən deyil, təhlükəli kod tərəfindən törədilir. Nümunə olaraq, SQL Injection zəifliyinin 20 ildən çox olmasına baxmayaraq, ondan istifadə edilən hücumlar hələ də ən çox yayılmış hücum növüdür. Bu zəiflik bütünlüklə daxilolma sanitarizasiyasının tərtibatçı tərəfindən edilməməsi ilə bağlıdır və bu, etibarsız daxiletmənin heç bir filtrləmə olmadan işlənməsi ilə nəticələnir. Bu, verilənlər bazası sisteminin özü tərəfindən səhv daxil edilməsindən yaranmır.

Bu üsul yalnız inyeksiya hücumları ilə işləyir. Kodu yoxlamaq adətən bu qədər sadə deyil. Əvvəlcədən qurulmuş proqramın ən son versiyasına yenilənməsi onun heç bir həssas kodun olmadığına zəmanət verəcək. Lakin, əgər siz xüsusi hazırlanmış proqramlardan istifadə edirsinizsə, inkişaf komandanız hərtərəfli kodu nəzərdən keçirməlidir. Hansı proqramdan istifadə etməyinizdən asılı olmayaraq, kodun təhlükəsizliyi vacibdir. Əks halda, onlayn tətbiqin təməli qeyri-sabit və hücuma həssas olacaq (Shikhar Goel, 2023).

- **Proqram təminatının güncəl saxlanması**

Ən son yeniləmələrdən istifadə üçüncü tərəf tərəfindən təmin edilən proqram təminatından istifadə edərkən kodun təhlükəsiz olmasına zəmanət verməyin ən yaxşı yoludur. Əsas veb tətbiqi tərəfindən istifadə edilən bir çox komponent, əgər düzəldilməzsə, effektiv hücumlarla nəticələnə bilər. Məsələn, adi Linux veb server quraşdırması proqram təminatının həssas versiyalarının sui-istifadə edilməsinin qarşısını almaq üçün müntəzəm olaraq yenilənməli olan bir sıra xidmətlərə malik olacaq. Məsələn, MySQL və PHP hər ikisi bir zamanlar zəifliklərə qarşı həssas idi, lakin sonradan düzəldildi.

Məzmunlarına xidmət etmək üçün HTTPS-dən istifadə edən veb proqramların əksəriyyətində mövcud olan OpenSSL-də heartbleed (boşluq) zəifliyi yeniləmənin əhəmiyyətini vurğulayır. Bununla belə, bu zəifliklər aradan qaldırmaq üçün müvafiq yamaq istifadəyə verildikdən sonra proqram təminatınızı yeniləmək kifayətdir (Shikhar Goel, 2023).

- **İcazəsiz müdaxilələrdən müdafiə**

Proqram yeniləmələri sisteminizin məlum zəifliklərdən azad olmasına zəmanət versə də, hələ də əvvəlki məsləhətlərimizin diqqətdən kənarda qaldığı və təcavüzkarın sisteminizə daxil olmasına imkan verən giriş nöqtələri ola bilər. “Firewall”-ların faydalı olduğu yer budur. Quraşdırmalarınız əsasında trafiki məhdudlaşdırdığı və əməliyyat sistemlərinin əksəriyyətində əvvəlcədən quraşdırıldığı üçün firewall tələb olunur.

Bununla yanaşı, bir firewall yalnız şəbəkə trafikini təhlil edə bilər, buna görə də bir veb tətbiqini yerləşdirsəniz, mütləq Veb Tətbiq Firewall qurmalısınız. Veb serverə göndərilən zərərli sorğuların aşkarlanmasına gəldikdə, WAF-lar ən uyğundur. Sorğu veb serverə çatmazdan əvvəl WAF, onda SQL Injection faydalı yükü aşkar edərsə, onu rədd edəcək. Dayandırılmalı olan sorğulardan asılı olaraq, əgər WAF müəyyən sorğulara müdaxilə edə bilmirsə, aralıqda xüsusi qaydalar da qoya bilərsiniz (Glen Willims, 2024).

- **Veb Zəiflik Skanlarının həyata keçirilməsi**

Son məhsul istənilən sayda kod nəzərdən keçirilməsi və təkmilləşdirmələr nəticəsində istismara məruz qala bilməz və ya həssas edilə bilməz. Veb zəifliyinin aşkarlanması çox vacibdir, çünki çalışan kod yoxlanılmır, kod nəzərdən keçirilməsi məhduddur. Ağ qutunun skan edilməsi və ya kodun nəzərdən keçirilməsindən fərqli olaraq, veb skanerlər veb tətbiqini "qara qutu" kimi görəcək və burada son məhsulu təhlil edəcəklər. Boz qutunun skan edilməsi müəyyən skanerlərin təklif etdiyi başqa bir xüsusiyyətdir. Bu, veb sayt skanlarını kodu təhlil edə bilən bir backend agenti ilə birləşdirməyi əhatə edir.

Müasir veb proqramların mürəkkəbliyini və ölçüsünü nəzərə alsaq, əllə daxil olma testi apararkən bəzi zəiflikləri nəzərdən qaçırmaq asan ola bilər. Veb zəifliyi skanerləri məlum qüsurların əksəriyyətini aşkarlaya və bu prosesi sizin üçün avtomatlaşdıra bilər, bu da sizə daha böyük vebsaytı daha tez əhatə etməyə imkan verir. Veb skanerləri hələ də DOM əsaslı XSS kimi zəiflikləri aşkar edə bilir, bu, tanınmış, lakin tapmaq çətin olan zəiflikdir. Siz bu boşluqları həll etməyə çalışarkən, veb zəiflik skanerləri sizə Veb Tətbiq Təhlükəsizlik Divarının (WAF) bloklaması lazım olan sorğular göndərəcək (Glen Willims, 2024).

- **Monitorinqin əhəmiyyəti**

Müasir veb proqramların mürəkkəbliyini və ölçüsünü nəzərə alsaq, əllə daxil olma testi apararkən bəzi zəiflikləri nəzərdən qaçırmaq asan ola bilər. Veb zəifliyi skanerləri məlum qüsurların əksəriyyətini aşkarlaya və bu prosesi sizin üçün avtomatlaşdıra bilər, bu da sizə daha böyük vebsaytı daha tez əhatə etməyə imkan verir. Veb skanerləri hələ də DOM əsaslı XSS kimi zəiflikləri aşkar edə bilir, bu, tanınmış, lakin tapmaq çətin olan zəiflikdir. Siz bu boşluqları həll etməyə çalışarkən, veb zəiflik skanerləri sizə Veb Tətbiq Təhlükəsizlik Duvarının (WAF) bloklaması lazım olan sorğular göndərəcək.

Siz server qeydlərinə diqqət yetirməklə və məsələn, fayl əlavə edildikdə və ya silindikdə göndəriləcək xəbərdarlıqları qurmaqla bunun qarşısını ala bilərsiniz. Bu şəkildə, əgər siz həmin xüsusi faylı dəyişməmisinizsə, başqasının serverinizə icazəsiz daxil olmasından xəbərdar olacaqsınız. Hücumun faylları dəyişdirmək qədər gizli

olmadığı vəziyyətlərdə, məsələn, veb serverinizə qarşı Xidmətdən imtina hücumu baş verdikdə, iş vaxtına da diqqət yetirə bilərsiniz. Bu alətlər veb saytınız bağlanan kimi sizi xəbərdar edəcək və sizi ziyarətçilərin problem barədə məlumat verməsini gözləmək məcburiyyətindən xilas edəcək (Glen Willims, 2024).

Monitorinq xidmətlərini izlənilməli olan ilə eyni veb serverə yerləşdirmək, edə biləcəyiniz ən pis şeydir. Monitorinq xidmətinin bu serverin ləğv edilib-edilmədiyini sizə bildirməsinin heç bir yolu yoxdur.

- **Həmişə öyrənməyə davam edin**

Nəhayət, veb təhlükəsizliyi haqqında bilmək üçün hər şeyi bilmək heç vaxt kifayət etmir. Demək olar ki, hər gün veb saytınız üçün yeni bir istismar var, buna görə də onlayn tətbiqinizi necə daha təhlükəsiz etmək barədə öyrənməyi heç vaxt dayandırmamalısınız. Sıfır günlük hücumlar istənilən vaxt baş verə bilər, ona görə də tətbiq edə biləcəyiniz hər hansı yeni təhlükəsizlik tədbirindən xəbərdar olmaq çox vacibdir. Bir neçə veb təhlükəsizlik bloqları veb-sayt administratorunun öz veb-saytını necə dəstəkləməli olduğunu göstərən belə məsləhətlər verir (Shikhar Goel, 2023).

Web Shell hücumu

PHP, JSP və ya ASP-də yazılmış zərərli skriptlər - ən çox istifadə olunan üç veb proqram dilləri - veb qabıqları kimi tanınır. Onları veb server üçün əməliyyat sistemində quraşdırmaq uzaqdan idarəetməni asanlaşdırır. Təhdid edənlər veb qabığı silahlanmış zaman faylları dəyişdirə və hətta hədəflənmiş veb serverin kök kataloquna daxil ola bilərlər. Veb qabığı hücumları həm interneti hədəfləyə bilər.

Web Shell hücumları necə işləyir?

Shodan.io kimi skan etmə texnologiyalarından istifadə edərək kiber hücumçular əvvəlcə veb qabığı hücumlarına həssas olan serverləri müəyyənləşdirirlər. Veb serverlər və son nöqtələr kimi gizli veb serverlərə daxil olmaq üçün hücum vektoru kimi istifadə edilə bilən internetə qoşulmuş istənilən cihaz Shodan tərəfindən ifşa

olunur. Ekspozisiya üçün düzəliş quraşdırılmazdan əvvəl, kiberhücumçular zəiflik aşkar edildikdən sonra tez bir veb qabığı hücumu həyata keçirirlər.

CVE-2020-5902 zəifliyindən istifadə kibercinayətkarların veb qabığı inyeksiyalarını mümkün edən zəifliklərdən necə tez istifadə etdiklərinin nümunəsidir. F5 Networks 30 iyun 2020-ci ildə Trafik İdarəetmə İstifadəçi İnterfeysi (TMUI) üçün düzəliş buraxdı. Zəiflik zərərli kodların uzaqdan hədəflənmiş sistemə, Uzaqdan Kodun İcrası (RCE) kimi tanınan kiberhücum daxil edilməsini asanlaşdırdı.

Web Shell hücumunun nümunəsi:

Çinin kibercinayətkar dəstəsi Hafnium 2021-ci ilin martında başlıqlara çevrilən son əhəmiyyətli onlayn mərmi hücumunu həyata keçirdi. Hücumun agenti Microsoft Exchange Serverlərində ciddi bir boşluq vasitəsilə veb qabığına daxil edilmiş China Chopper adlı zərərli proqram idi.

Zədələnmiş sistemdə yaradılan China Chopper veb qabığının arxa qapısı server zəifliyi aradan qaldırıldıqdan sonra uzun müddət davam etdi və bu, onu daha təhlükəli etdi. (Kost Edward, 2023).

Web Shell inyeksiyalarını necə bloklamaq olar?

Veb qabığının inyeksiyasına imkan verən qüsurları həll etmək veb qabığı hücumlarını dayandırmaqdan daha asandır. İT ekosisteminizdə hər hansı potensial veb qabığı inyeksiya saytlarını tapmaq və düzəltmək üçün aşağıda tövsiyə olunan nəzarət və təhlükəsizlik alətlərindən istifadə edin.

- Ən Son Təhlükəsizlik Güncəlləmələrini izləyin

Veb qabığı hücumları üçün ən çox görülən marşrutlar təhlükəsizlik qüsurlarıdır. Bu giriş nöqtələrinə girişin qarşısını almaq üçün bütün veb serverlərə, veb proqramlara, məzmun idarəetmə sistemlərinə və üçüncü tərəf proqramlarına ən son təhlükəsizlik yamaqlarının tətbiq olunduğundan əmin olun.

- Həddindən artıq veb server xüsusiyyətlərini söndürün

Əgər php.ini-də veb server skriptləri ilə qarşılıqlı əlaqədə olan funksiyalar deaktiv edilirsə, yeridilmiş veb qabığı icra edə bilməyəcək.

Bu veb server xüsusiyyətləri arasında:

exec ()
eval()
shell _exec()
assert()

- Həssas Kataloqların Adlarını dəyişin

İdeal olaraq, bunun baş verməsinin qarşısını almaq üçün düzgün olmayan şəkil fayllarının yüklənməsinə icazə verən qovluqlar tamamilə söndürülməlidir.
Bu həssas qovluqların standart adları dəyişdirilməlidir ki, bu cür yükləmə metodu tələb olunarsa, onları tapmaq çətinləşsin. İnsayder təhdid hücumlarını azaltmaq üçün bu dəyişikliklərə yalnız imtiyazlı şəxslər daxil ola bilməlidir.
Bundan əlavə, veb serverinizə yüklənməsinə icazə verilən fayl növlərini məhdudlaşdıran bir filtr təyin edin.

- Hər Ehtiyacsız WordPress Plugini söndürün

WordPress plaginlərinin hər kəs, hətta kibercinayətkarlar tərəfindən hazırlana bilməsi onları məşhur hücum vektorlarına çevirir. Hər hansı kənar plaginləri sildiyinizə və bu vektorların təhlükəsizliyini təmin etmək üçün yalnız nüfuzlu tərtibatçıların plaginlərini quraşdırdığınızdan əmin olun.

- Firewall yerləşdirin

Bütün şəbəkə trafikini filtrləyərək, Veb Tətbiq Firewall (WAF) təhlükəli faydalı yüklərin və veb qabıqlarının mühitə yeridilməsinin qarşısını alır. Antivirus proqramında olduğu kimi, firewallınızı ən son kibertəhlükəsizlik yamaları ilə yeniləmək çox vacibdir.

- Fayl bütövlüyünün monitorinqini praktikada tətbiq edin

Təmiz kataloq skriptlərinin və qovluq dəyişikliklərinin vaxt ştampları fayl bütövlüyünün monitorinqi sistemi ilə müqayisə edilir. Məqsədli veb serverin kod kataloqunda nəzərdə tutulan quraşdırma ya rədd ediləcək, ya da uyğunsuzluq aşkar edilərsə, qarşısı alınacaq (Kost Edward, 2023).

II FƏSİL. Veb Sayt haqqında istifadə olunan Proqram Təminatları və DevOps ilə əlaqə

2.1 C# veb servicelər və soap ilə əlaqəsi

Microsoft öz.NET texnologiyası üçün təkmilləşdirilmiş dil olan yeni nəsil kompüter dilləri olan C# yaratdı.

Qısaca olaraq.NET Framework (həmçinin.NET texnologiyası və ya.NET platforması kimi tanınır) müəyyən edək. 2000-ci ildən əvvəl, gələcək proqramçılar istifadə etməli olduqları proqramlaşdırma dili ilə bağlı ciddi qərar çatışmazlığı ilə mübarizə aparırdılar. Çünki seçilmiş texnologiyanın ətrafdan nə dərəcədə asılı olması, onların hansı işi yerinə yetirəcəyi və seçilmiş proqramlaşdırma dilinin nə qədər öhdəsindən gələ biləcəyi kimi elementlərə diqqət yetirilməli idi. Bundan əlavə, veb proqramları yazarkən dillər ətraf mühitdən asılı olduğundan, proqramçılar müxtəlif arxitekturalı çoxsaylı fərqli sistemlərdən və maşınlardan ibarət olan İnternet olan virtual aləmdə ilişib qalırlar.

Proqramlaşdırma dilində hazırlanmış proqram mühitdən asılılıq səbəbindən yalnız müəyyən sistemlərdə və müəyyən prosessorlarda işləyə bilər. Tutaq ki, siz sistemin prosessor sayının ancaq idarə edə biləcəyi bir proqram qurursunuz. Proqramımızı sistemə və prosessora uyğunlaşdırmaq üçün yenidən tərtib etmək, onu başqa bir sistem üçün yazmaq lazım olurdu. İşi qüsursuz yerinə yetirə biləcək bir proqramlaşdırma dili yox idi. Yəni, yüksək səviyyəli bir layihə üçün zəmin qurulduqdan sonra iş o yerə çatdı ki, layihənin müəyyən bir hissəsi uyğun olan bir dildə, digər hissələr isə digər dildə yazılması uyğun görüldü. Bundan əlavə, Microsoft əməliyyat sistemi olan.NET Framework-ü istehsal etdi və onunla uyğun gələn çoxlu sayda cari proqramlaşdırma dillərini təqdim etdi. Bu, onun bir çox proqramlaşdırma dillərində eyni tapşırığı yerinə yetirməyə qadir olduğunu göstərir.

Xüsusi standartlara cavab verən və .NET Framework-dən kitabxanalardan istifadə edən dilin Framework ilə uyğun olduğu deyilir. Buna görə də .NET Framework mühitində proqramçı layihəni daha kiçik komponentlərə ayıra, hər bir komponent üçün

ən yaxşı dili seçib onunla işləyə, sonra komponentləri bir yerə qoyub hazır məhsulu göstərə bilər. Bunun üçün .NET Framework mühiti hazırlanmışdır.

NET Framework üçün ən çox istifadə edilən proqramlaşdırma dillərinin çoxu da optimallaşdırılıb. Buna baxmayaraq, heç bir proqramlaşdırma dili.NET Framework ilə tam uyğunlaşdırıla bilməz. çünki dilin strukturunda əsaslı dəyişiklik mümkün deyildi. Nəticədə Microsoft, .NET Framework-ü tam dəstəkləyən proqramlaşdırma dili olan C#-ı yaratdı.

Başqa sözlə, C# dili innovativ və irəli düşünən .NET Framework mühitində işləmək üçün xüsusi olaraq hazırlanmışdır. Obyekt yönümlü proqramlaşdırmanın əsasları tamamilə C# ilə dəstəklənir. C# dili dünyanın ən nüfuzlu kompüter dili mütəxəssislərindən biri olan Anders Hejlsberg tərəfindən yaradılmışdır. 1960-cı illərdə geniş istifadə olunan Turbo Pascal dilinin orijinal tərtibatçısı: (Rufulllazada Rəhman, 2018).

Şək.2.1. Anders Hejlsberg
(https://habrastorage.org/r/w1560/getpro/habr/post_images/742/1b2/88a/7421b288a12bc47a8f614190cd5c69b8.jpg)

RESTful API nədir?

İki kompüter sistemi arasında İnternet üzərindən təhlükəsiz məlumat mübadiləsi üçün interfeys RESTful API adlanır. Müxtəlif vəzifələri yerinə yetirmək üçün iş proqramlarının əksəriyyəti digər daxili və ya xarici proqramlarla interfeys tələb edir. Məsələn, aylıq əmək haqqı cədvəli yaradarkən hesab-faktura prosedurlarını avtomatlaşdırmaq üçün daxili vaxt cədvəli tətbiqi ilə əlaqə qurmağınız lazım ola bilər. Alternativ olaraq, daxili hesabat sisteminizdəki məlumatları müştərinin bank sistemindəki məlumatlarla mübadilə etməli ola bilərsiniz. RESTful API-ləri müxtəlif sistemlər arasında məlumat axını asanlaşdıran təhlükəsiz, təhlükəsiz və effektiv proqram rabitəsi üçün standart təklif edir.

RESTful API-lər hansı üstünlükləri təklif edə bilər?

- Migyaslanan

REST API-lərdən istifadə edən sistemlər yaxşı inkişaf edə bilər, çünki REST müştəri-server əlaqələrini optimallaşdırır. Davamlılıq serverdən əvvəlki müştəri sorğularından verilənləri saxlamağı tələb etmədiyindən, server yükünü aradan qaldırır. Müəyyən müştəri-server qarşılıqlı əlaqələri yaxşı idarə olunan keşləmə strategiyası ilə minimuma endirilə və ya tamamilə aradan qaldırıla bilər. Bu aspektlərin hər biri əlaqə sıxlığından performansın azalması ilə nəticələnmədən miqyas almağa imkan verir.

- Uyğunlaşma

Ümumi müştəri-server ayrılması RESTful veb xidmətləri tərəfindən dəstəklənir. Hər bir komponentin müstəqil inkişafına imkan vermək üçün bu xidmətlər çoxsaylı server komponentlərini ayırır və sadələşdirir. Server tətbiqinin platformasına və ya texnologiyasına edilən dəyişikliklər müştəri tətbiqinə heç bir təsir göstərmir. Tətbiq funksiyalarını yığma qabiliyyəti daha çox yönlülük əlavə edir. Tərtibatçılar, məsələn, verilənlər bazasını dəyişdirmədən verilənlər bazasını üzərinə yaza bilərlər.

- Bağımsızlıq

İstifadə olunan texnologiyadan asılı olmayaraq, REST API-ləri işləyir. API dizaynını dəyişdirmədən müştəri və server proqramlarını inkişaf etdirmək üçün

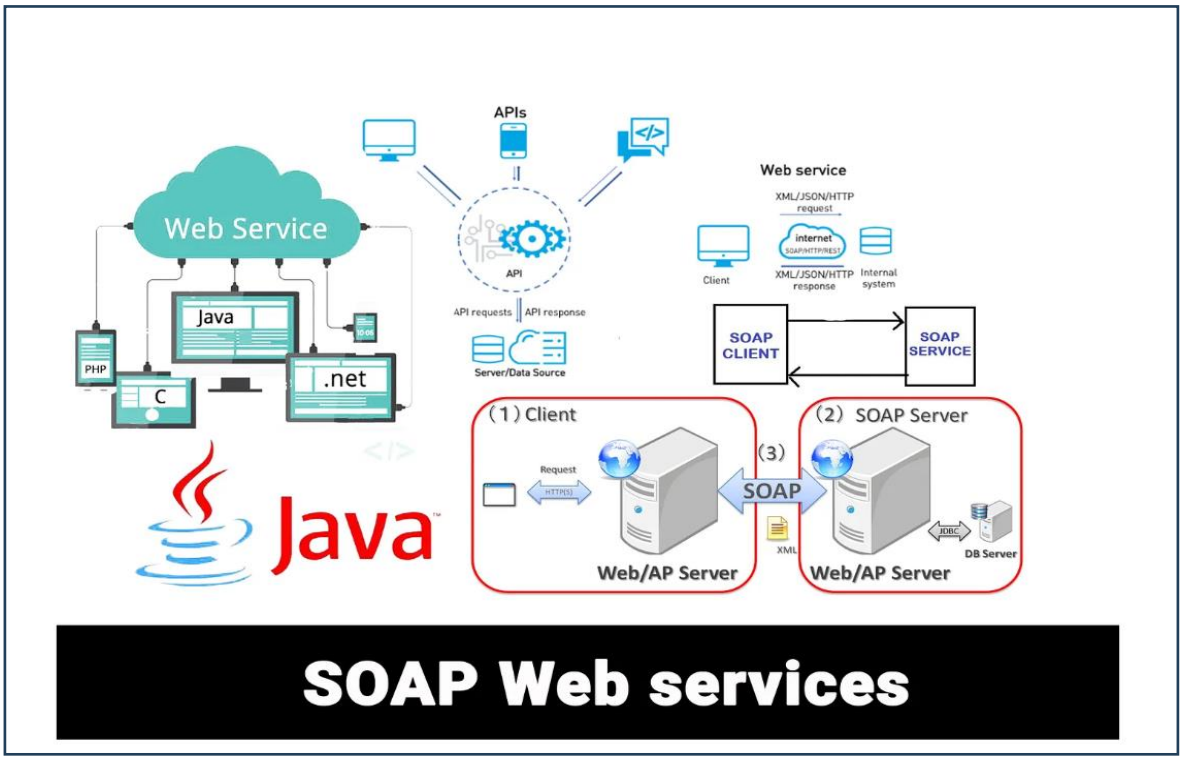
müxtəlif proqramlaşdırma dillərindən istifadə edilə bilər. Bundan əlavə, hər iki tərəfin əsas texnologiyaları əlaqəni pozmadan dəyişdirilə bilər (Deniz Buğa, 2023).

RESTful API necə işləyir?

RESTful API mahiyyətcə internetlə eyni şəkildə işləyir. İstifadəçi resursa ehtiyac duyduqda ondan istifadə etmək üçün API vasitəsilə serverə qoşulur. Server tərəfi API-dən istifadə tələbləri sənədlərdə tam izah olunur. İstənilən növ REST API sorğusunun edilməsində iştirak edən əsas proseslər aşağıdakılardır:

1. Server müştəridən sorğu alır. Sorğu, API sənədlərinə riayət etməklə serverin başa düşə biləcəyi şəkildə tərtib edilmişdir.
2. Server müştərinin şəxsiyyətini və istifadəçinin bu sorğu üçün uyğunluğunu təsdiq edir.
3. Sorğunu qəbul etdikdən sonra server onu içəridə yoxlayır.
4. İstifadəçi serverdən cavab alır. Cavabdakı məlumat sorğunun uğurlu olub-olmadığını göstərir. İstifadəçinin tələb etdiyi məlumat da cavaba daxil edilir (Deniz Buğa, 2023).

SOAP Veb servis

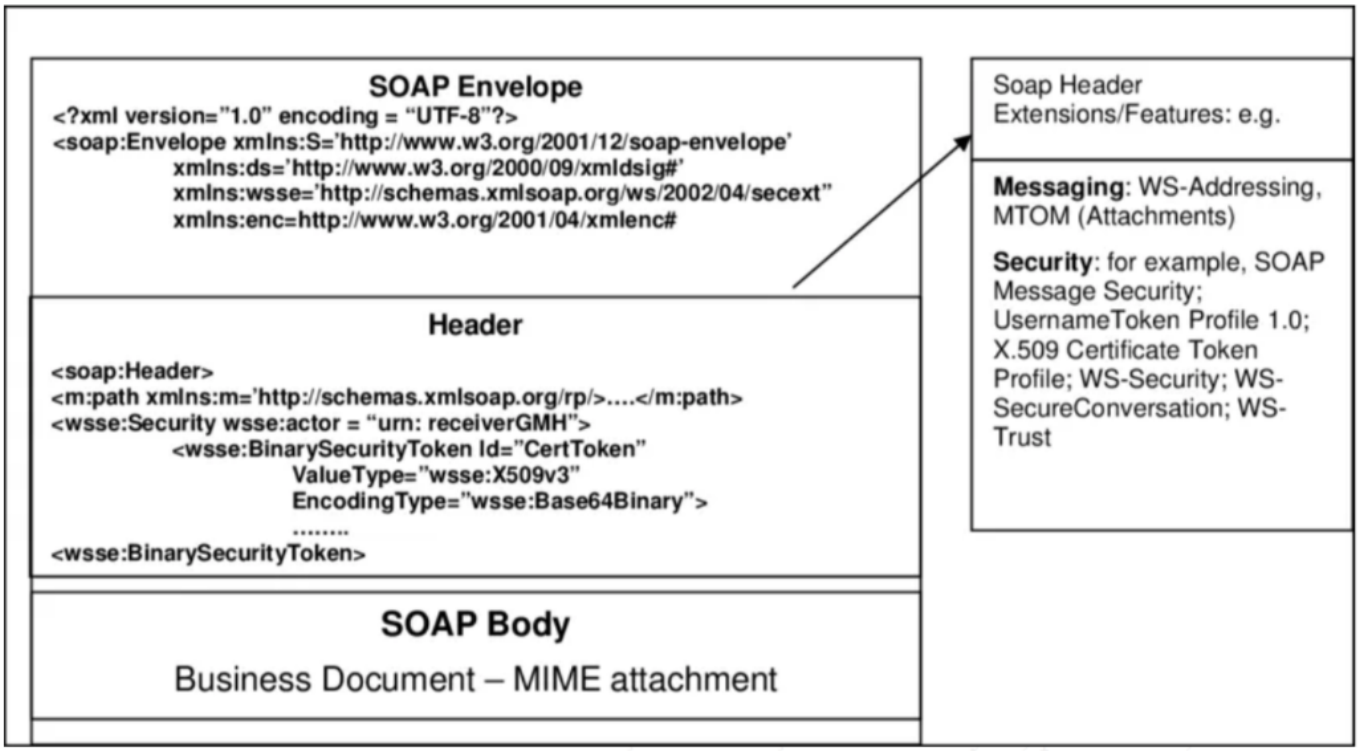


Şək.2.2. SOAP
(https://miro.medium.com/v2/resize:fit:1400/format:webp/1*ecBXg7uI6pr1TQU9dVBNew.jpeg)

Sadə Obyekt Giriş Protokolu və ya SOAP, veb xidmətləri yaratmaq üçün kompüter şəbəkələrində istifadə edilmək üçün hazırlanmış bir protokoldur.

1. SOAP-ın məqsədləri müstəqillik, detallılıq, neytrallıq və genişlənməkdir.
2. XML formatı SOAP protokolunda məlumat mübadiləsi üçün istifadə olunur.
3. HTTP (Hyper Text Transfer Protocol) adətən SOAP mesajlarını göndərmək üçün istifadə olunur, TCP/IP isə bəzən də istifadə olunur.
4. Biznes təşəbbüslərində, Java və.Net dilləri ilə əlaqə qurulduqda məlumat səhvlərini və inteqrasiyanı minimuma endirmək üçün SOAP-dan istifadə edilir.

SOAP-ın dörd əsas bölməsi var.



Şək.2.3. SOAP hissələri (https://www.researchgate.net/profile/Lucas-Venter/publication/220803160/figure/fig1/AS:669549062021138@1536644344878/The-SOAP-envelope-header-can-be-extended-with-standard-features.png)

1. ***Envelop:***

SOAP strukturunun kök elementi adlanır. Bu tələb olunur və istifadəsi məcburdir.

2. ***Header:***

HTML-də baş etiketinə bənzəyir. Bu, autentifikasiya və SOAP ilə əlaqəli digər tapşırıqların tamamlandığı bölmədir. Bu hissə tərtibatçıdan asılıdır və istəyə bağlıdır.

3. ***Body:***

SOAP-ın bədəni onun ən vacib komponentidir. Bir çox funksional nasazlıqların nəticələri və təfərrüatları bu bölmədə verilmişdir.

4. ***Fault:***

Bu, potensial SOAP səhvlərini idarə edən bölmədir. Səhv məlumatı və səhv mesajları bu bölmədə yer alır.

Veb xidmətlərindən istifadə etmək üçün funksiyalar, parametrlər, xidmət ünvanı və s. Sahələri bilməliyik. WSDL ilə biz bu vəzifələri yerinə yetirə bilirik.

SOAP sorğuları üçün WSDL sorğularından istifadə edilir. WSDL adlı dil XML əsaslı veb xidmətləri yaratmaq və bütün SOAP sorğularının qeydlərini saxlamaq üçün nəzərdə tutulmuşdur.

Veb xidməti aşağıdakı strukturlardan istifadə edir:

Növlər (**Types**): Mesajın nəzərdə tutulan məlumat növləri müəyyən edilir.

Mesaj (**Message**): Ünsiyyətdə istifadə ediləcək müəyyən mesaj növləri var.

Bağlama **(Binding):** Əməliyyatlarda və ya mesajlarda istifadə ediləcək məlumat formatları bağlama ilə müəyyən edilir.

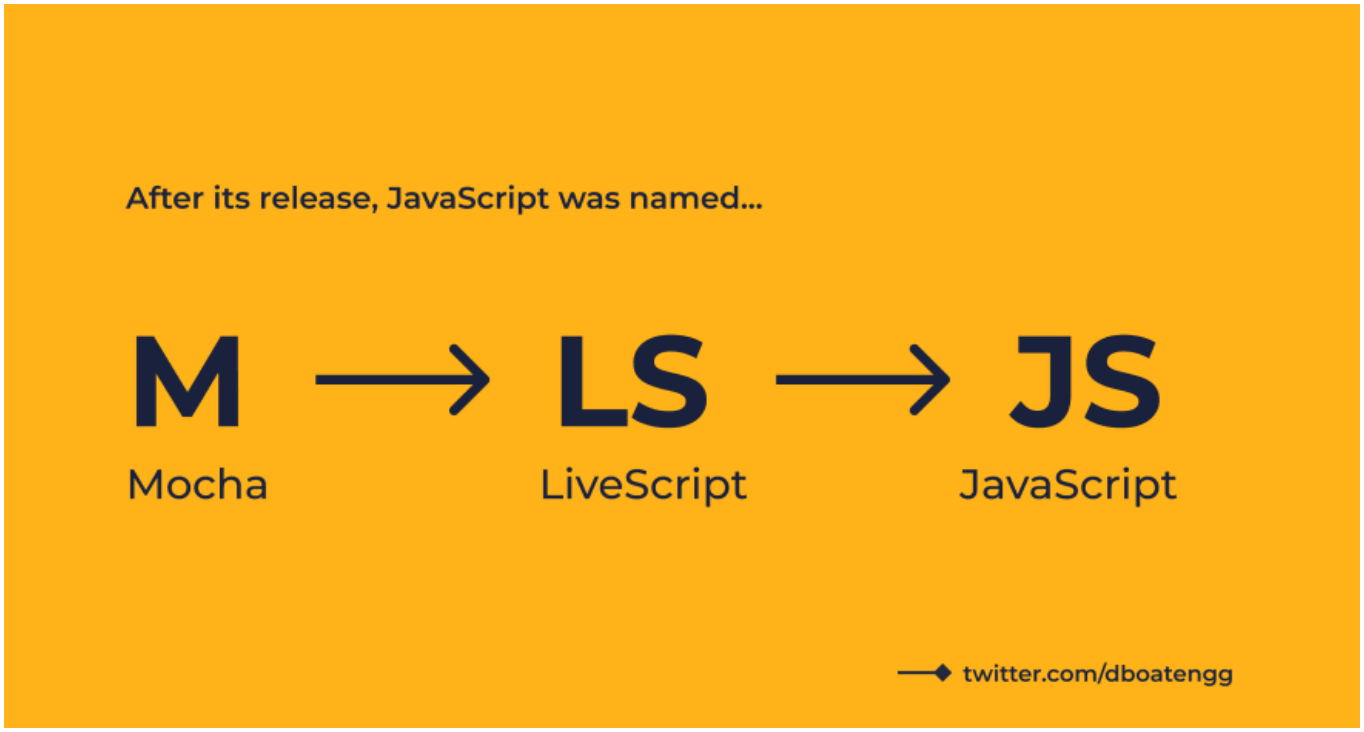
Xidmət (**Servise**): İstifadə edilməli olan portlar dəsti.

Port (**Port**): Xidmət nöqtəsini müəyyən etmək üçün veb ünvanı və bağlamadan ibarət olan port istifadə olunur.

Port Növü (**PortType**): Veb xidmətin elan edilmiş funksiyaları və müxtəlif mesajlar içərisindədir (Shikhar Goel, 2023).

2.2 JavaScript

JavaScript 1995-ci ilin sentyabrında cəmi on gün ərzində Netscape proqramçısı Brendan Eich yeni proqramlaşdırma dili yaratdı. Əvvəlcə Mocha kimi tanınan o, tezliklə LiveScript, sonra isə JavaScript kimi populyarlıq qazandı (Boateng Dickson, 2022).

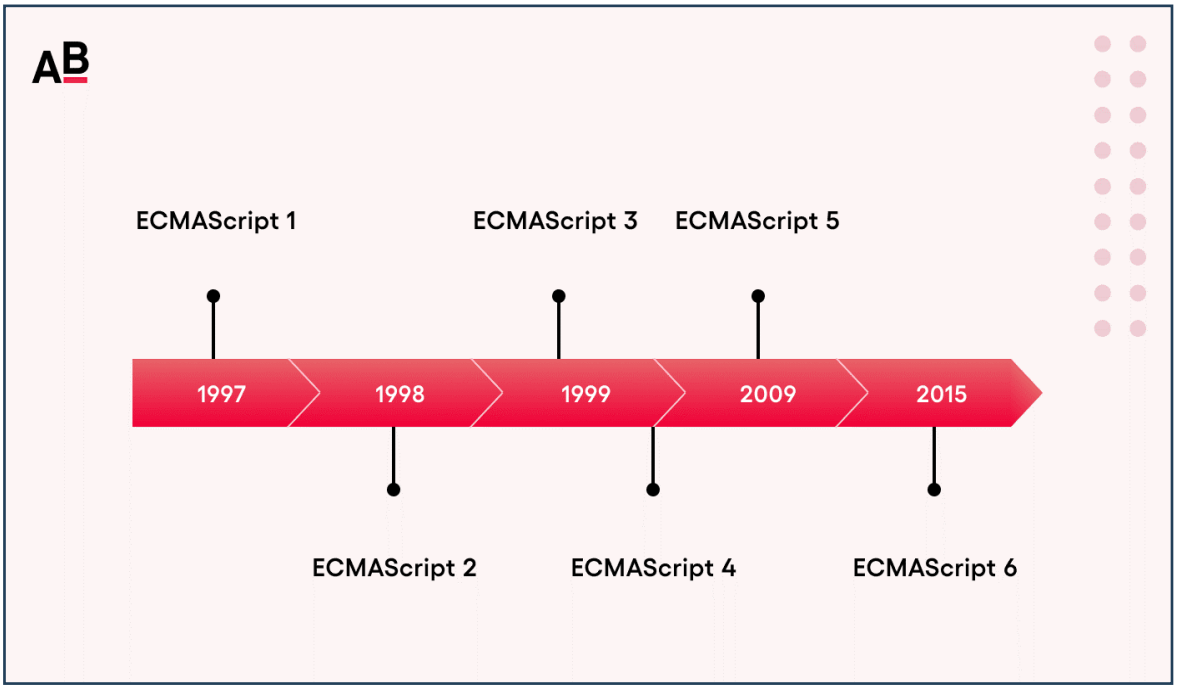


Şək.2.4. JavaScriptin tarixi (https://res.cloudinary.com/practicaldev/image/fetch/s--UAH--eoA-/c_limit%2Cf_auto%2Cfl_progressive%2Cq_auto%2Cw_880/https:/dev-to-uploads.s3.amazonaws.com/uploads/articles/eeiyxz1y851jd4eqxws0.png)

HTML və CSS ilə yanaşı, JavaScript yüksək səviyyəli, çox paradiqmalı proqramlaşdırma dilidir və daha çox vebsayt yaratmaq üçün istifadə olunur. Veb saytın strukturu və görünüşü HTML və CSS tərəfindən təmin edilir, lakin interaktiv xüsusiyyətlər və davranışlar yalnız JavaScript ilə əlavə edilə bilər. Bu, istifadəçilərin veb-saytla daha maraqlı üsullarla əlaqə saxlamasına imkan yaradır. Qeyd etmək lazımdır ki, dil obyekt yönümlü və prototip əsaslı proqramlaşdırma da daxil olmaqla imperativ, funksional və hadisəyə əsaslanan proqramlaşdırma üslublarını dəstəkləyir və hər hansı bir əməliyyat sistemi ilə məhdudlaşmır (Boateng Dickson, 2022).

JavaScript-in standartlaşdırılması (ECMAScript)

Netscape ilkin olaraq JavaScript-i təqdim edəndə bazardakı bütün brauzer provayderləri arasında müharibə gedirdi. JavaScript Microsoft və hər biri fərqli ad və sintaksisi olan çoxsaylı digər brauzer provayderləri tərəfindən çoxsaylı brauzer versiyalarında tətbiq edilmişdir. Bir brauzerdə yaxşı işləyən kod digərində tamamilə yararsız olduğu üçün bu, tərtibatçıların çoxlu baş ağrısına səbəb oldu. Bu, onların hamısını brauzerlərində eyni dildən, JavaScript-dən istifadə etmək qərarına gələnə qədər bir müddət davam etdi (Shikhar Goel, 2023).



Şək.2.5. ECMAScriptin tarixi (https://almablog-media.s3.ap-south-1.amazonaws.com/Frame_37_min_787b9dd30d.png)

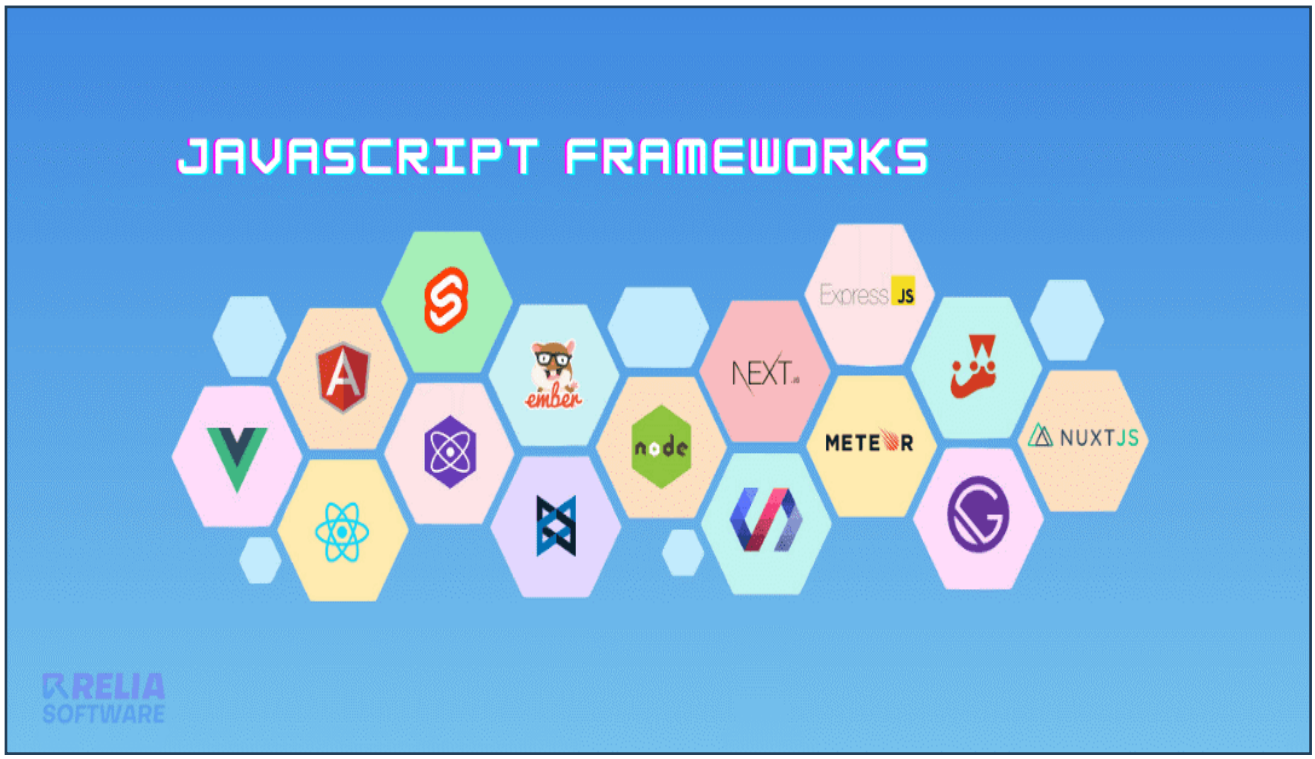
Nəticə etibarilə, Netscape dil üçün müvafiq texniki qulluq və yardımı təmin etmək üçün JavaScript-in standartlaşdırılması üçün Avropa Kompüter İstehsalçıları Assosiasiyasına (ECMA) müraciət etdi. ECMA JavaScript-i standartlaşdırdığı üçün o, rəsmi olaraq ECMAScript kimi tanındı.

İndiyə qədər ECMAScript sadəcə JavaScript rəsmiləşdirilib, lakin JScript və ActionScript kimi digər dillər də ECMAScript standartının üzərində qurulub. Onları bir mühərriki paylaşan üç fərqli nəqliyyat vasitəsi kimi düşünün.

Brauzerlər və Node.js JavaScript üçün iki əsas host mühitdir. Bu kontekstlər vasitəsilə dilə bir neçə API əlavə edilir. ECMAScript bütün xarici API-lər bu parametrlərdən silindikdə qalan şeydir. Sadə dillə desək, ECMAScript host mühiti olmayan sadəcə JavaScript-dir (Shikhar Goel, 2023).

2.3 JavaScript Framework-ləri

Framework-in istifadəsi ilə proqram təminatının inkişaf etdirilməsi prosesləri sürətləndirilə, standartlaşdırıla və daha effektiv edilə bilər. Proqram mühəndisləri müəyyən tapşırıqları yerinə yetirmək üçün framework-lərdə olan əvvəlcədən hazırlanmış alətlər, kitabxanalar, konvensiyalar və standartlardan istifadə edə bilərlər. Framework-lər müəyyən bir proqramlaşdırma dilinin və ya texnoloji yığının üstündə qurulur və tez-tez tərtibatçılara əsas tikinti blokları təklif edir (Akshay Kedari, 2024).



Şək.2.6. JavaScript Framework-ləri
(https://api.reliasoftware.com/uploads/best_javascript_frameworks_8e66b245c0.png)

Bir neçə JavaScript Framework-ləri

- **AngularJS**

AngularJS adlı pulsuz və açıq mənbəli JavaScript framework-i proqramçılara qabaqcıl onlayn proqramlar yaratmağa kömək edir. O, HTML-ə əlavə xüsusiyyətlər əlavə edir və tək səhifəli proqramlar (SPA) üçün idealdır.

2009-cu ildə istifadəyə verildiyi vaxtdan Google-un AngularJS framework-i veb inkişaf sənayesində mühüm rol oynamışdır. SPA-ların qurulması AngularJS-in parladığı yerdir. Bunlar brauzerə yalnız bir dəfə yüklənən və səhifənin tam yenilənməsini tələb etmədən məzmunu yeniləyən proqramlardır və istifadəçi təcrübəsini daha qüsursuz edir (Shikhar Goel, 2023).

- **Vue.js**

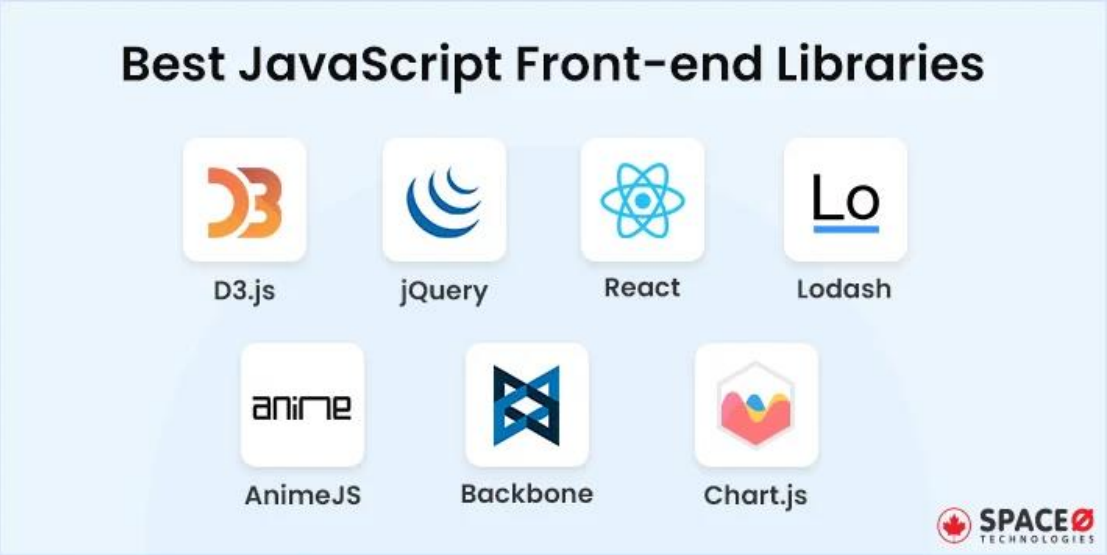
Vue.js istifadəsi asan, çevik və güclü mütərəqqi JavaScript framework-dir. Vue-nin daxili istifadəçi interfeysi məlumat dəyişdikdə avtomatik olaraq yenilənən mütərəqqi framework-dir. Asanlıq, səmərəlilik və genişlənmə qabiliyyəti ilə tərtibatçılar Vue.js ilə müasir, interaktiv onlayn proqramlar yarada bilərlər. O, istifadəçi interfeysləri yaratmaq üçün sürətli deklarativ, komponent əsaslı proqramlaşdırma metodologiyasını təklif edir və ümumi HTML, CSS və JavaScript-dən istifadə edir. (Glen Willims, 2024).

- **Next.js**

İnanılmaz imkanları sayəsində React platformasında qurulmuş açıq mənbəli veb inkişaf framework-i Next.js kifayət qədər populyarlaşdı. Vercel tərəfindən yaradılmış Next.js, mürəkkəb axtarış sisteminin optimallaşdırılması (SEO) və server tərəfində göstərilməsi (SSR) daxil olmaqla, güclü xüsusiyyətləri ilə diqqət çəkir. Next.js-in daxili marşrutlaşdırma xüsusiyyəti tətbiqinizdə naviqasiyaya nəzarət etməyi və dinamik marşrutlar qurmağı asanlaşdırır (Shikhar Goel, 2023).

2.4 JavaScript kitabxanası

JavaScript kitabxanası, tərtibatçıların veb inkişaf işlərində istifadə edə biləcəyi əvvəlcədən yazılmış funksiyaların, metodların və siniflərin məcmusudur. O, inkişaf prosesini asanlaşdırır və kodun təkrar istifadəsinə imkan verir. O, həm də vaxta və səyə qənaət etdiyi üçün istifadə olunur və brauzerlər arası qarşılıqlı fəaliyyət təklif edir (Shikhar Goel, 2023).



Şək.2.7. JavaScript Libraries (https://almablog-media.s3.ap-south-1.amazonaws.com/Frame_37_min_787b9dd30d.png)

Bir neçə JavaScript Libraries-ləri

- **JQuery**

JQuery adlı yüngül, pulsuz və açıq mənbəli JavaScript alət dəsti hadisələrlə işləməyi, Document Object Model-ni (DOM) manipulyasiya etməyi və dinamik onlayn təcrübələr yaratmağı asanlaşdırır. jQuery-nin əsas məqsədi veb səhifələrdə JavaScript istifadəsini asanlaşdırmaqdır. Bir sətirli funksiyaları ilə jQuery çətin JavaScript əməliyyatlarını asanlaşdırır və kodunuzun oxunaqlılığını və davamlılığını

yaxşılaşdırır. Veb saytlarınızın cavab vermə qabiliyyəti və qarşılıqlı əlaqəsi jQuery-ni başa düşməklə əhəmiyyətli dərəcədə artırıla bilər (Shikhar Goel, 2023).

- **React.js**

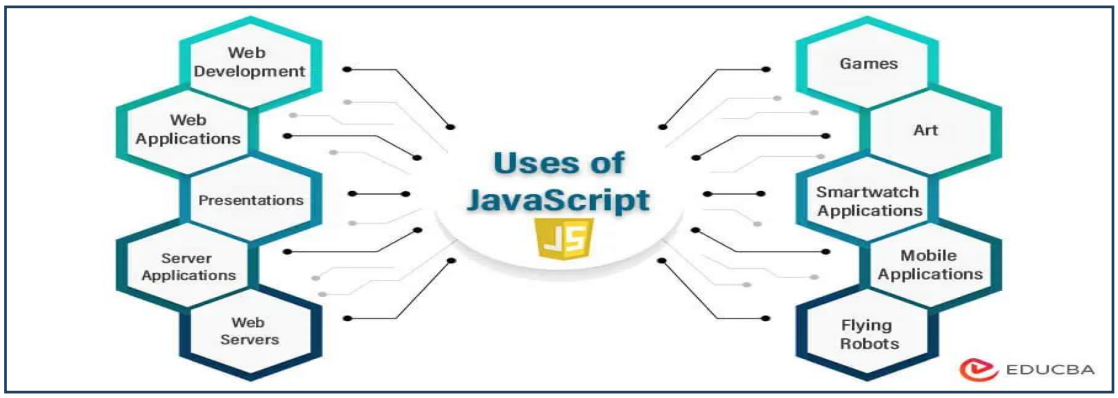
React adlı güclü JavaScript paketi dinamik və interaktiv istifadəçi interfeysləri (UI) yaratmaq üçün istifadə edilə bilər. Onu inkişaf etdirən Facebookdur. React, təkrar istifadə edilə bilən istifadəçi interfeysi elementlərinin yaradılmasına imkan verməklə böyük onlayn tətbiqləri idarə etməyi və saxlamağı asanlaşdıran komponent əsaslı dizaynı ilə məşhurdur. React bir səhifəlik proqramlar yaratmaq üçün bir vasitədir (Whitle Dustin, 2014).

- **Lodash**

Lodash adlı JavaScript kitabxanası underscore.js üzərində işləyir. Onun sayəsində massivlər, sətirlər, obyektlər, rəqəmlər və s. ilə işləmək asanlaşır. O, bizə bir sıra daxili funksiyalar təklif edir və funksional proqramlaşdırma metodologiyasından istifadə edir ki, bu da JavaScript kodlamasını başa düşməyi asanlaşdırır, çünki işlər təkrarlanan funksiyaların işlənib hazırlanmasını tələb etməkdənsə, bir kod sətri ilə tamamlana bilər. Bundan əlavə, o, xüsusilə mürəkkəb obyektlərlə işləyərkən JavaScript obyektinin manipulyasiyasını asanlaşdırır (Shikhar Goel, 2023).

2.5 JavaScript-in ən yaxşı 10 praktik tətbiqi

JavaScript veb-saytlar və veb proqramlar yaratmaq üçün geniş istifadə olunur. Müxtəlif seqmentlərdə JavaScript-in bəzi praktik tətbiqlərini müzakirə edək (1)



Şək.2.8. JavaSicript-in istifadəsi (https://cdn.educba.com/academy/wp-content/uploads/2018/10/Uses-of-JavaScript-1.jpg.webp)

- **Veb inkişafı**

Veb səhifələri yaratmaq üçün istifadə olunan skript dili JavaScript adlanır. Netscape üçün hazırlanmış JavaScript-in köməyi ilə proqramçılar istifadəçilərə qarşılıqlı əlaqədə olmaq və mürəkkəb tapşırıqları yerinə yetirmək imkanı verən dinamik və interaktiv veb-saytlar yarada bilərlər. Bundan əlavə, istifadəçilərə bütün səhifəni yeniləmədən sənədə material əlavə etməyə imkan verir. JavaScript əksər vebsaytlar tərəfindən PDF sənədləri, vidjetlər və Flash proqramları kimi xarici proqram dəstəyi və təsdiqi üçün istifadə olunur. Google, YouTube, Facebook, Wikipedia, Yahoo, Amazon, eBay, Twitter və LinkedIn kimi bir neçə tanınmış vebsayt JavaScript istifadə edənlər arasındadır.

- **Veb Tətbiqləri**

Güclü onlayn proqramlar müxtəlif JavaScript framework-lərindən istifadə etməklə hazırlanır. Bütün istifadəçilər Google Maps kimi bir tətbiqdə xəritəni araşdırmaq üçün etməlidirlər ki, ətraflı görünüş əldə etmək üçün siçanı klikləyib sürükləyin. JavaScript serverlərlə əlaqə saxlamadan brauzerlə əlaqə saxlayaraq bunu gücləndirir. Angular və Vue, onlayn proqramların inkişafına kömək edən məşhur JavaScript front-end framework-ləridir. Tətbiq Proqramlaşdırma İnterfeysləri (API) və AngularJS JavaScript framework-i Netflix və PayPal-ın inkişafında istifadə edilmişdir.

- **Təqdimatlar**

Təqdimatlarla interaktiv veb-səhifələr yaratmaq üçün JavaScript-dən istifadə nisbətən ümumi istifadə halıdır. HTML istifadə edərək, RevealJs və BespokeJs kitabxanaları ilə veb-əsaslı slayd lövhələri yaradıla bilər. RevealJs aləti ilə bütün CSS rəng formatlarında keçid üslubları, mövzuları və slayd fonları olan interaktiv slayd göyərtələri hazırlana bilər. BespokeJs framework-i dinamik marker siyahıları, ölçüsünün dəyişdirilməsi, sintaksis vurğulanması və s. kimi imkanlarla doludur. JavaScript ilə istifadəçilər proqramlaşdırma dillərini çox yaxşı bilməsələr belə vebsaytlar və təqdimatlar yarada bilərlər.

- **Server Proqramları**

JavaScript həmçinin server tərəfi proqramlar yaratmaq üçün açıq mənbəli Node.js iş vaxtı mühiti ilə istifadə olunur. Kodlayıcılar genişlənən və sürətli şəbəkə proqramları üçün kod yaratmaq, sınaqdan keçirmək və sazlamaq qabiliyyətinə malikdir. JavaScript HTTP sorğularının idarə edilməsini və məzmunun yaradılmasını asanlaşdırır. Uber, Walmart, PayPal, GoDaddy və bir çox başqaları kimi tanınmış korporasiyalar server arxitekturası üçün Node.js-i qəbul ediblər.

- **Veb serverləri**

Node.js istifadə edərək tərtibatçılar JavaScript ilə veb server yarada bilərlər. Node.js hadisəyə əsaslandığı üçün növbəti zəngə keçməzdən əvvəl əvvəlki zəngin cavabını gözləmir. Serverlər çox sürətlə buferləşdirmədən böyük həcmdə məlumat göndərirlər. CreateServer () funksiyası HTTP modulu tərəfindən serverlər yaratmaq üçün istifadə olunur.

- **Oyunlar**

JavaScript-in digər əhəmiyyətli istifadəsi onlayn oyunların yaradılmasıdır. JS-dən istifadə edərək oyunlar yaratarkən, HTML5 və JavaScript birlikdə olduqca yaxşı işləyir. Oyunlar üçün zəngin qrafika EaselJS kitabxanası vasitəsilə təmin edilir. Tam onlayn giriş HTML5 ilə Flash kimi digər plaginlərə ehtiyac olmadan mümkündür. HTML5 və JavaScript Tower Building, CrossCode və HexGL kimi mürəkkəb brauzer oyunları üçün əsas yaradır.

- **İncəsənət**

JavaScript-də HTML5-ə daha yeni əlavə olan canvas elementi veb-səhifədə 2D və 3D qrafika yaratmağı asanlaşdırır. Çoxsaylı brauzer əsaslı rəqəmsal sənət layihələri bundan faydalanmışdır. Öz JavaScript kodunuzu yazaraq rəqəmsal rəssam olun.

- **Ağıllı saat proqramları**

Pebble proqramçılara Pebble saatları üçün JavaScript proqramları yazmağa imkan verən Pebble.js adlı JavaScript framework-ini təklif edir. Ağıllı saat proqramı yaratmaq üçün sadə JavaScript kodu yazın.

- **Mobil Proqramlar**

Qeyri-veb parametrləri və ya İnternetdə olmayan obyektlər üçün proqramların hazırlanması JavaScript-in ən güclü istifadələrindən biridir. Mobil cihazların bütün zamanların ən yüksək populyarlığı ilə, Windows, iOS və Android daxil olmaqla bir neçə əməliyyat sistemi üçün mobil proqramlar yaratmağı asanlaşdırmaq üçün JavaScript çərçivələri hazırlanmışdır. Platformalar arası mobil tətbiqetmənin inkişafı React Native framework-i sayəsində mümkün olur ki, bu da tərtibatçılara həm iOS, həm də Android üçün universal ön hissəyə çıxış imkanı verir.

- **Uçan Robotlar**

Onu da qeyd edək ki, JavaScript istifadə edərək uçan robotu proqramlaşdıra bilərsiniz. İstifadəçilər Node.js mühitinin köməyi ilə müxtəlif miniatür robotları, ixtiraçı layihələri və Əşyaların İnterneti cihazlarını idarə edə bilərlər. JavaScript ilə işləyən dronların, uçan robotların və kvadrokopterlərin füsunkar dünyasını kəşf edin (Padelis Kefalidis, 2024).

2.3 DevOps

Nəticələri vaxtında və etibarlı şəkildə inkişaf etdirmək, sınaqdan keçirmək və buraxmaq üçün proqram təminatının inkişaf etdirilməsi prosesləri DevOps kimi tanınan inteqrasiya edilmiş fəaliyyətlər və ya təcrübələr toplusu vasitəsilə İT tərtibatçıları ilə əlaqələndirilir. İnteqrasiya edilmiş "DevOps" sözü həm inkişafa, həm də əməliyyatlara, həm də mədəni baxımdan aiddir (Allan Joaquin, 2023).

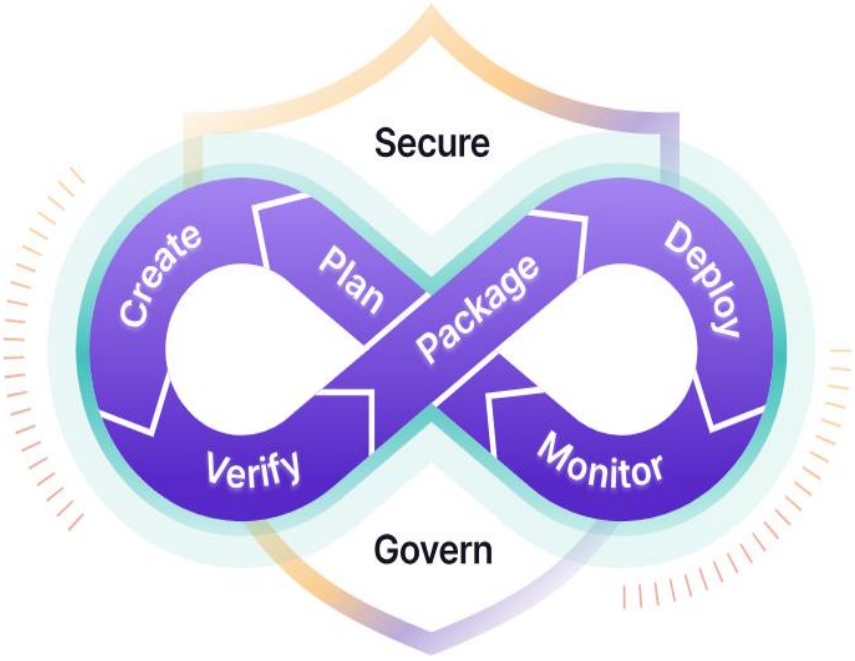
İnkişaf və əməliyyat komandalarının inteqrasiyasına bir sıra amillər kömək etdi. Konsepsiya qrupların rollarının nə qədər fərqli bölündüyünə görə qrupların uzun müddətdir davam edən problemlərindən və maneələrindən qaynaqlanır ki, bu da onların əməliyyatlarını daha mürəkkəb edir (Allan Joaquin, 2023). Mürəkkəbliyin sürətli artması problemləri fərdi nöqteyi-nəzərdən həll etmək qabiliyyətinin azalması ilə əlaqələndirilir. Birləşmə ilə nəticələnən bir neçə mühüm hərəkət böyük bir şəbəkə üzərində virtual kompüterlər təklif edir, genişləndirilmiş şəbəkə cihazlarını və serverlərini mürəkkəb şəkildə qurur və müxtəlif proqramları istifadəyə verir (Allan

Joaquin, 2023). Üstəlik, logların toplanması və yığılması, xidmət monitorinqi, şəbəkə performansının monitorinqi və proqram performansının monitorinqi mürəkkəb problemlərə çevrilib.

Birlikdə tərtibatçıların və operatorların unikal qarışığı faktiki dünyada mövcud olan əməliyyat problemlərinin həlli üçün hörmətli strategiya yaratdı (Whitle Dustin, 2014). Bu mürəkkəb proseslər daha asan başa düşmək üçün idarə oluna bilən hissələrə bölünmüşdür. Bunlar jurnalın idarə edilməsi, konfiqurasiyanın idarə edilməsi, yerləşdirmənin avtomatlaşdırılması, performansın idarə edilməsi və monitorinqdən ibarətdir.

Keçmişdə əməliyyat parametrlərinin əksəriyyətində istehsal sahəsinə kimin daxil ola biləcəyini, dəyişikliklər edə biləcəyini və onların nə vaxt həyata keçiriləcəyini tənzimləyən ciddi siyasət və prosedurlar var idi. Xüsusilə məlumat mərkəzlərində bu dəyişikliklər aparatdan istifadə etməklə əl ilə edilməli idi.

Ənənəvi üsullarla müqayisədə, DevOps inkişaf sahələrini (Dev) və əməliyyatları (Ops) birləşdirərək proqram təminatının hazırlanması və çatdırılmasının səmərəliliyini, sürətini və təhlükəsizliyini artırır.



Şək.2.9. Devops mentiqi
(https://images.ctfassets.net/xz1dnu24egyd/2S16xLgZGnBkxXgFVQOrxv/24e5808aba2b4c7024c15daa6b6ef5f7/loop-white.svg)

Müəssisələr və onların müştəriləri daha çevik proqram təminatının işlənməsinin həyat dövründən rəqabət üstünlüyü əldə edirlər.

DevOps-u təsvir etməyin ən yaxşı yolu təhlükəsiz proqramı tez bir zamanda dizayn etmək, inkişaf etdirmək və çatdırmaq üçün birgə səydir. Proqram təminatının hazırlanması (inkişaf) və əməliyyatlar (əməliyyatlar) komandaları DevOps prinsiplərinin köməyi ilə avtomatlaşdırma, komanda işi, sürətli rəy və təkrar təkmilləşdirmə yolu ilə çatdırılmanı sürətləndirə bilər. Proqram təminatının hazırlanması üçün Çevik metodologiyadan irəli gələn DevOps proseduru, qurulmasının fənlərarası strategiyasını genişləndirir və tətbiqləri daha sürətli və iterativ olaraq buraxır (Whitle Dustin, 2014).

İnkişaf dövrü ərzində daha çox əməkdaşlıq mühitini təşviq etməklə, siz DevOps inkişaf metodologiyasını tətbiq etməklə tətbiqin axını və dəyərin çatdırılmasını gücləndirməyi seçirsiniz. DevOps İT mədəniyyətinin düşüncə tərzində dəyişiklik deməkdir. Çevik, yalın və sistemlər nəzəriyyəsinə əsaslanaraq, DevOps proqram təminatının artımlı inkişaf yolu ilə miqyasda çatdırılmasını vurğulayır. Müvəffəqiyyət üçün hesabatlılıq mühitini, təkmil komanda işi, empatiya və şirkət nəticələri üçün ortaq cavabdehliyi inkişaf etdirmək bacarığı vacibdir.

Proqram təminatının inkişafı (dev) və əməliyyatlar (ops) DevOps yaratmaq üçün birləşdirilir. O, əməliyyatlar və inkişaf qruplarının işini birləşdirmək üçün ortaq hesabatlılıq və əməkdaşlıq mədəniyyətini inkişaf etdirməyə çalışan proqram mühəndisliyi texnikası kimi təsvir edilir (Whitle Dustin, 2014).

DevOps yanaşması

DevOps texnikasının məqsədi sistemlərin inkişaf dövrünün müddətini azaltmaqla yüksək keyfiyyətli proqram təminatını davamlı olaraq təmin etməkdir. O, sürətli geribildirim dövrlərinə, avtomatlaşdırmaya, inteqrasiyaya və komanda işinə böyük diqqət yetirir. Bu keyfiyyətlər proqram təminatını daha tez və etibarlı şəkildə quran, sınaqdan keçirən və buraxan mədəniyyətə kömək edir.

Bu texnikanın dörd əsas prinsipi tətbiqin inkişafı və yerləşdirilməsinin effektivliyini və səmərəliliyini tənzimləyir. Aşağıdakı prinsiplər siyahısı müasir proqram təminatının inkişafının ən yaxşı xüsusiyyətlərinə diqqət yetirir.

Əsas DevOps konsepsiyaları

Proqram təminatının işlənməsinin həyat dövrü avtomatlaşdırılır. Bu, sınaq, quraşdırma, buraxılışlar, inkişaf mühitlərinin təmin edilməsi və digər əl əməliyyatları kimi proqram təminatının çatdırılması prosesinə insan səhvini maneə törədə və ya təqdim edə bilən əl proseslərini avtomatlaşdırmağı əhatə edir.

Avtomatlaşdırma güclü DevOps komandasının xüsusiyyətidir, lakin güclü komanda işi və ünsiyyət də vacibdir.

Tullantıların azaldılması və davamlı təkmilləşdirmə. Yüksək performanslı DevOps komandaları, təkrarlanan tapşırıqların avtomatlaşdırılması və ya buraxılış müddətlərini və ya bərpa müddətini qısaltmaq yolları üçün performans göstəricilərinin monitorinqi olsun, həmişə təkmilləşdirmə yollarını axtarır.

İstifadəçi tələblərinə və qısa geribildirim dövrlərinə hiper diqqət.

DevOps komandaları real istifadəçilərin əslində nə istədiklərinə və avtomatlaşdırma, təkmil kommunikasiya və əməkdaşlıq və davamlı təkmilləşdirmə vasitəsilə bunu necə təmin etmələrinə diqqət yetirə bilirlər.

Təşkilatlar bu konsepsiyaları həyata keçirməklə tətbiqin planlaşdırılmasını təkmilləşdirə, bazara çıxma vaxtı sürətləndirə və kod keyfiyyətini artıra bilər (Whitle Dustin, 2014).

DevOps dörd faza

DevOps-un inkişafında hər biri təşkilati strukturda və texnologiyada dəyişikliklərlə xarakterizə olunan dörd əsas dövr olmuşdur. Bu təkamül DevOps-un əsasən iki əsas tendensiyaya görə necə çətinləşdiyini göstərir:

I. **Mikroservislərə keçid:** Şirkətlər monolit dizayndan daha çevik mikroservis arxitekturasına keçdikcə ixtisaslaşdırılmış DevOps alətlərinə ehtiyac artıb. Bu dəyişiklik mikroxidmətlərin təmin etdiyi daha böyük çeviklik və qranularlıqdan istifadə etməyə çalışır.

II. **Alət İnteqrasiyasında artım:** Layihələr çoxaldıqca əlavə DevOps alətlərinə olan tələbat da artır və bu da layihələr arasında alət inteqrasiyalarının sayında nəzərəçarpacaq artımla nəticələnir. Təşkilatlar indi bu mürəkkəbliyə görə DevOps həllərindən istifadə və inteqrasiya üçün fərqli strategiyanı nəzərdən keçirirlər.

DevOps-un inkişafında hər biri dörd əsas mərhələ olmuşdur

DevOps tarixində dörd əsas mərhələ meydana gəldi, hər biri artan ehtiyaclara və proqram təminatının inkişafı və çatdırılmasının mürəkkəbliyinə cavab verir.

Bu dörd mərhələ aşağıdakılardır:

Mərhələ 1: BYOD (Öz DevOplarınızı gətirin)

Bring Your Own DevOps mərhələsində hər komanda öz avadanlığını seçir. Bu üsulla əməkdaşlıq etməyə çalışan komandalar bir-birlərinin texnologiyaları ilə tanış olmadıqlarından çətinliklərlə qarşılaşdılar. Bu mərhələ layihənin daha qüsursuz idarə edilməsi və komandanın qarşılıqlı fəaliyyətini təmin etmək üçün daha vahid alətlər toplusunun zəruriliyini aydınlaşdırdı.

Mərhələ 2: Sənayedə aparıcı DevOps

Təşkilatlar fərqli alətlərdən istifadə etməklə yaranan problemləri həll etmək üçün öz sinfində ən yaxşı DevOps kimi tanınan ikinci mərhələyə keçirlər. Təşkilatlar DevOps həyat dövrünün hər mərhələsi üçün fərqli alət seçərək, bu mərhələ boyu vahid alətlər

dəsti üzərində standartlaşdırılıb. Nəticədə komandalar birlikdə daha yaxşı işləyə bildilər, lakin problem hər addımda müxtəlif alətlər arasında proqram yeniləmələrinin ötürülməsinə gəldikdə yarandı.

Mərhələ 3: DIY DevOps-u əhatə edir

Təşkilatlar bu problemi həll etmək üçün mövcud alətlər üzərində və onların arasında quraraq DIY DevOps-u qəbul etdirlər. Onların DevOps nöqtə həlləri inteqrasiya etmək üçün geniş ixtisaslaşmış iş tələb edirdi. Buna baxmayaraq, bu alətlər heç vaxt tam uyğun gəlmir, çünki onlar inteqrasiya nəzərə alınmaqla ayrıca yaradılmışdır. DIY DevOps-un saxlanması bir çox təşkilat üçün çoxlu əmək və artan xərc tələb edirdi, çünki mühəndislər əsas proqram məhsulunu inkişaf etdirməkdənsə, alətlərin inteqrasiyasına daha çox vaxt sərf etməli idilər.

Mərhələ 4: DevOps üçün platforma

Tək tətbiq platformasından istifadə həm korporativ məhsuldarlığı, həm də komanda təcrübəsini artırır. DIY DevOps bütün DevOps həyat dövrü üzərində görmə və nəzarəti təmin edən DevOps platforması ilə əvəz olunur.

DevOps platforması DevOps-un tam potensialına nail olmaq yolunda böyük bir addımdır, çünki o, bütün komandalara - İnkişaf, Əməliyyatlar, İT, Təhlükəsizlik və Biznes - proqram təminatını planlamaq, qurmaq, qorumaq və çatdırmaq üçün birlikdə işləməyə imkan verir. vahid sistemi bitir (Shikhar Goel, 2023).

DevOps mühitinin üstünlükləri

İstehsal mühitinin təkmilləşdirilməsi yolu ilə proqram təminatının çatdırılma sürətinin artırılması və davamlı təkmilləşdirmə DevOps-un biznes dəyəri və DevOps mədəniyyətinin üstünlükləridir. Siz bazarı pozanları qabaqcadan görməli və onları aradan qaldırmaq üçün tez hərəkət etməlisiniz. Bu, komandaların Agile proqram təminatının inkişaf etdirilməsi yanaşmasında təqdim edə biləcəyi muxtariyyət və sürət sayəsində mümkün olur ki, bu da tamamlanmamış işin həcmini azaldır. Komandalar bu baş verdikdən sonra bazarın sürəti ilə tələblərə reaksiya verə bilərlər.

DevOps-un nəzərdə tutulduğu kimi işləməsi üçün bir neçə əsas ideya həyata keçirilməlidir, məsələn:

1. Xüsusilə bir komandanın əsas performans göstəriciləri (KPI) və digər komandanın uğur ölçüləri birbaşa ziddiyyət təşkil etdikdə, maneələr və məhdudiyyətlər təmin edən institusional silosları və təhvil-təslimləri aradan qaldırın.
2. Komandalar arasında əməkdaşlığa və paylaşmağa imkan verən vahid alətlər zənciri kimi xidmət edən vahid proqram yaradın. Komandalar daha tez işləyə və nəticədə bir-birinə tez rəy bildirə biləcəklər (Shikhar Goel, 2023).

DevOps həyat dövrü və DevOps necə işləyir ?

Proqram təminatının inkişafının başlanğıcından çatdırılma, texniki xidmət və təhlükəsizliyə qədər DevOps kimi tanınan bir həyat dövrü var. DevOps həyat dövrü aşağıdakı mərhələlərdən ibarətdir:

- ❖ Planlaşdırın: Tapşırıqları toplayın, onları sıralayın və onların yerinə yetirilməsinə nəzarət edin.
- ❖ Yaradın: Layihə məlumatlarını və kodunu yazmaq, dizayn etmək, inkişaf etdirmək və təhlükəsiz idarə etmək üçün komandanızla birlikdə işləyin.
- ❖ Doğrulayın: İdeal olaraq, kodunuzun keyfiyyət standartlarınıza uyğun olduğuna və nəzərdə tutulduğu kimi işlədiyinə əmin olmaq üçün avtomatlaşdırılmış testdən istifadə edin.
- ❖ Paket: Artefaktlar yaradın, konteynerləri idarə edin, proqramlarınızı və asılılıqlarınızı paketləyin.
- ❖ Təhlükəsiz: Potensial zəiflikləri müəyyən etmək üçün asılılıq skanı, qeyri-səlis testi, statik və dinamik test və digər üsullardan istifadə edin.
- ❖ Buraxılış: Proqramı son istifadəçilər üçün əlçatan edin.
- ❖ Konfiqurasiya edin: Tətbiqlərinizin yedəklənməsi üçün lazım olan infrastruktura nəzarət edin və quraşdırın.
- ❖ Monitor: Problemlərin şiddətini və tezliyini azaltmağa kömək etmək üçün performans göstəricilərini və səhvləri izləyin.
- ❖ İdarə et: Siyasət uyğunluğuna, təhlükəsizlik qüsurlarına və təşkilati təhlükəsizliyə nəzarət edin (Shikhar Goel, 2023).

DevOps alətləri, prinsipləri və əsasları

Tətbiq ömrü boyu geniş təcrübələr DevOps tərəfindən əhatə olunur. DevOps ilə uğur qazanmaq üçün komandalar tez-tez bu üsullardan biri və ya bir neçəsi ilə başlayır.

Versiya idarəetməsi (**Version control**), mənbə koduna və digər fayllara edilən bütün dəyişiklikləri izləmək və nəzarət etmək üçün əsas prosedurdur. Mənbə kodun idarə edilməsi və versiyaya nəzarət bir-biri ilə sıx əlaqəli anlayışlardır.

Çevik inkişaf (**Agile)** layihənin çatdırılmasını sürətləndirmək və sadələşdirmək üçün artan, arıq və iterativ metodların tətbiqidir.

Davamlı İnteqrasiya (CI) (**Continuous Integration)** qurmaların avtomatik başlaması, modifikasiyaların sınaqdan keçirilməsi və bütün kod dəyişikliklərinin tez-tez əsas filiala inteqrasiyası prosesidir.

Davamlı Çatdırılma (CD) (**Continuous Delivery)** İnfrastruktur təminatı və tətbiqlərin yerləşdirilməsi prosesini avtomatlaşdırmaq üçün fasiləsiz çatdırılma davamlı inteqrasiya ilə əməkdaşlıq edir. Birlikdə onlar tez-tez CI/CD adlanır.

Shift left inkişaf prosesində çox əvvəl sınaq və təhlükəsizliyin gətirilməsini təsvir etmək üçün istifadə edilən bir ifadə. Bunu etməklə, daha yüksək standart kod qoruyarkən inkişaf daha sürətli gedə bilər (Shikhar Goel, 2023).

III FƏSİL. Simulyasiya və Nessus haqqında məlumat

3.1 Nessus kibertəhlükəsizlik cəmiyyətində ən çox istifadə edilən zəiflik skanerlərindən biridir. Bəs bu proqram necə işləyir?

Nessus potensial təhlükəsizlik qüsurları üçün şəbəkəyə qoşulmuş cihazları skan etmək üçün nəzərdə tutulmuş hər şeyi əhatə edən proqramdır. Təşkilatlar kibertəhlükəsizlik imkanlarını tədricən artırmaq üçün Nessus brauzerini düzgün konfiqurasiya etməli və istifadə etməlidirlər.

Nessusun tarixi

Renaud Deraison 1998-ci ildə açıq mənbəli zəifliyi skan edən Nessus alətini yaratdı.

Nessus dünya miqyasında milyonlarla yükləmə əldə edib və özünü kibertəhlükəsizlik sektoru üçün ən vacib vasitələrdən biri kimi təsdiqləyib.

Nessusun populyarlığı artdıqca, Tenable Network Security onu satın aldı və kommersiya istifadəsi üçün qapalı mənbə ilə təmin etdi, yəni daha ətraflı yeniləmələr və dəstək aldı.

Nessus qapalı mənbəli kommersiya proqramıdır, lakin geniş xüsusiyyətlər dəsti və dinamik plagin strukturuna görə o, mövcud zəiflikləri skan edən ən etibarlı vasitələrdən biri olmaqda davam edir.

Nessusun əsasları

Tenable Network Security zəiflikləri skan etmək üçün alət olan Nessus-u hazırlayır və yeniləyir. Müxtəlif növ təhlükəsizlik qüsurlarını müəyyən etmək üçün sənaye standartı kimi qəbul edilir. İnformasiya təhlükəsizliyi mütəxəssisləri serverlərdə, şəbəkə cihazlarında, sistemlərdə və proqramlarda həyata keçirilən skan əməliyyatları nəticəsində aydın təhlükəsizlik statusu hesabatı alırlar.

Kibertəhlükəsizlik mütəxəssisləri indi Nessus-u böyük verilənlər bazası və istifadə rahatlığına görə ən çox seçdikləri vasitələrdən biri hesab edirlər. Alətin əməliyyat sistemlərində və parametrlərdə çox yönlü olması onun üstünlüklərindən biridir. Məsələn, Linux, Mac OS X və Windows ilə istifadə etmək asandır. Bundan əlavə, qurumların potensial təhlükəsizlik pozuntularına daha yaxşı hazır olmasına kömək

edən NESL (Nessus Scripting Language) istifadə edərək yaradılmış minlərlə unikal skan imzaları sayəsində təhlükəsizlik zəiflikləri tez aşkar edilir.

Əsas elementlər və əməliyyatlar

Çoxsaylı müəssisələr və insanlar bütün dünyada təhlükəsizliyin skan edilməsi üçün hərtərəfli zəifliyin aşkarlanması vasitəsi olan Nessus-dan istifadə edirlər.

- Zəifliyin skan edilməsi şəbəkəyə qoşulmuş cihaz konfiqurasiyalarında zəif cəhətləri tapır.
- Zərərli proqram aşkarlanması kompüter sistemlərində mümkün zərərli proqramları tapır.
- Konfiqurasiya idarəetməsi sistemlərin təhlükəsizlik parametrlərinə cavabdehdir.
- Uyğunluq auditləri bir sıra qaydalara və təlimatlara əməl edilib-edilmədiyini yoxlayır.
- Şəbəkə skanı: Hədəf şəbəkənin cihazlarını və xidmətlərini görünən edir. Bu skanlar təşkilatın kibermüdafiəsini gücləndirmək üçün vacibdir.

Nessus uyğunlaşa bilən arxitekturası sayəsində müxtəlif əməliyyat sistemlərində və şəbəkə topologiyalarında yaxşı işləyə bilir. Sahib olduğu minlərlə unikal skan imzası ilə o, təhlükəsizlik qüsurlarını çox dərindən tapır və təfərrüatlandırır (Fahri Ulutaş, 2024).

Əsas Sistem Tələbləri

Nessus düzgün işləməsi üçün müəyyən sistem tələbləri toplusunu tələb edir. Maksimum performans üçün təklif olunan əməliyyat sistemlərinə, aparat xüsusiyyətlərinə və şəbəkə parametrlərinə diqqət yetirmək vacibdir. Bunlar taramanın sabitliyi və effektivliyi üçün vacibdir. Nessusun dəstəklədiyi Linux, Mac OS və ya Windows paylamalarından birinin yenilənmiş versiyası əməliyyat sistemi kimi istifadə edilməlidir. 64-bit arxitektura uyğunluğu sistem üçün tələbdir.

Axtarış prosesinin ölçüsü və mürəkkəbliyi serverin yaddaş (RAM) tutumuna və prosessorun (CPU) performansına təsir göstərir. Optimal performans üçün sizə çoxnüvəli prosessor və kifayət qədər operativ yaddaş lazımdır.

Nessus həm verilənlər bazasını, həm də skan nəticələrini saxlamaq üçün kifayət qədər böyük disk sahəsinə ehtiyac duyur. Saxlama seçimlərinin genişləndirilməsi üçün böyük verilənlər bazası və uzunmüddətli qeydlər qiymətləndirilməlidir.

Hədəf sistemlər şəbəkə bağlantısı üzərindən davamlı və etibarlı şəkildə əlçatan olmalıdır, bu da kifayət qədər bant genişliyinə malik olmalıdır. Bundan əlavə, firewallların və digər şəbəkə təhlükəsizliyi həllərinin Nessus-un işləyəcəyi şəbəkə seqmentinə uyğun olduğunu təsdiqləmək çox vacibdir.

Nəhayət, Nessus skanlarından gələn trafikin lüzumsuz olaraq dayandırılmadığından əmin olmaq üçün firewall və antivirus parametrləri müvafiq olaraq tənzimlənməlidir. Skanlama zamanı yanlış pozitivlərin və hər hansı ünsiyyət probleminin qarşısını almaq üçün müəyyən parametrlər çox vacibdir (Fahri Ulutaş, 2024).

Nessusun sadə istifadəsi

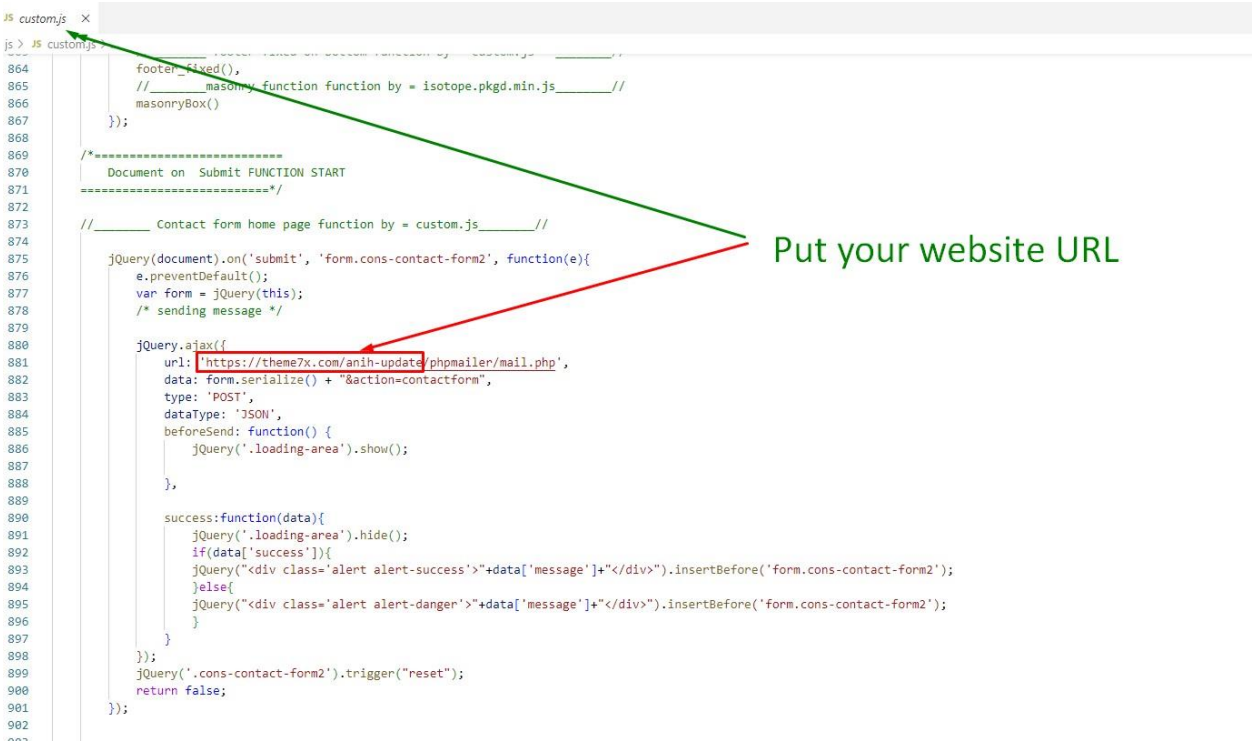
Nessus, şəbəkə təhlükəsizliyinin skan edilməsi üçün üç addımda istifadə olunan hərtərəfli bir vasitədir. Quraşdırma və konfiqurasiya mərhələlərini bitirməzdən əvvəl şirkətin və onun istifadəçilərinin tələbləri əsasında skan siyasətlərinə qərar verilir.

Hədəf şəbəkə seqmentləri, sistemləri və ya cihazları müəyyən edilmiş qaydalara uyğun olaraq növbəti mərhələdə skan edilir. Tarama zamanı fəaliyyət, portun skan edilməsi, zəifliyin aşkarlanması və konfiqurasiya xətalarının təhlili daxil olmaqla bir çox testlərdən istifadə olunur. Skan tamamlandıqdan sonra Nessus öz məlumatını ətraflı hesabata toplayır.

Nessus hesabatını nəzərdən keçirdikdən sonra istifadəçi son mərhələdə təhlükəsizlik qüsurlarını aşkar edir və onları aradan qaldırmağa üstünlük verir. Müvafiq düzəldici strategiyaların yaradılması da bu araşdırmaya əsaslanır (Fahri Ulutaş, 2024).

3.2 Skriptlərin tətbiqi və load-balans texnologiyası

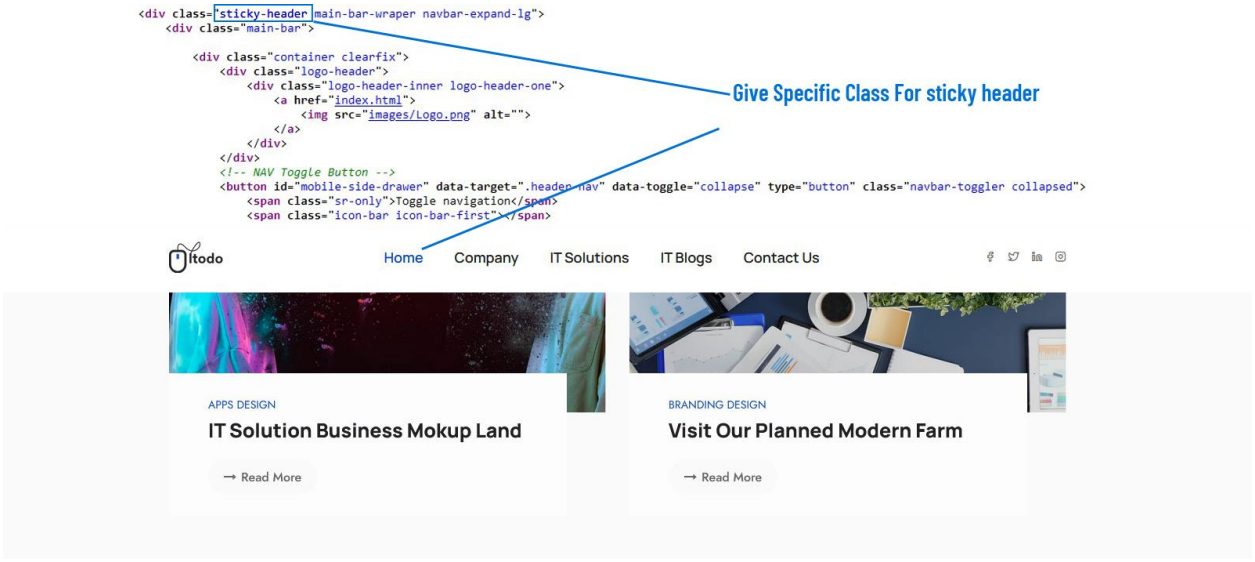
Burada mailin inteqrasiyasını ve SMTP-ilə əlaqəsini görəcəyik.



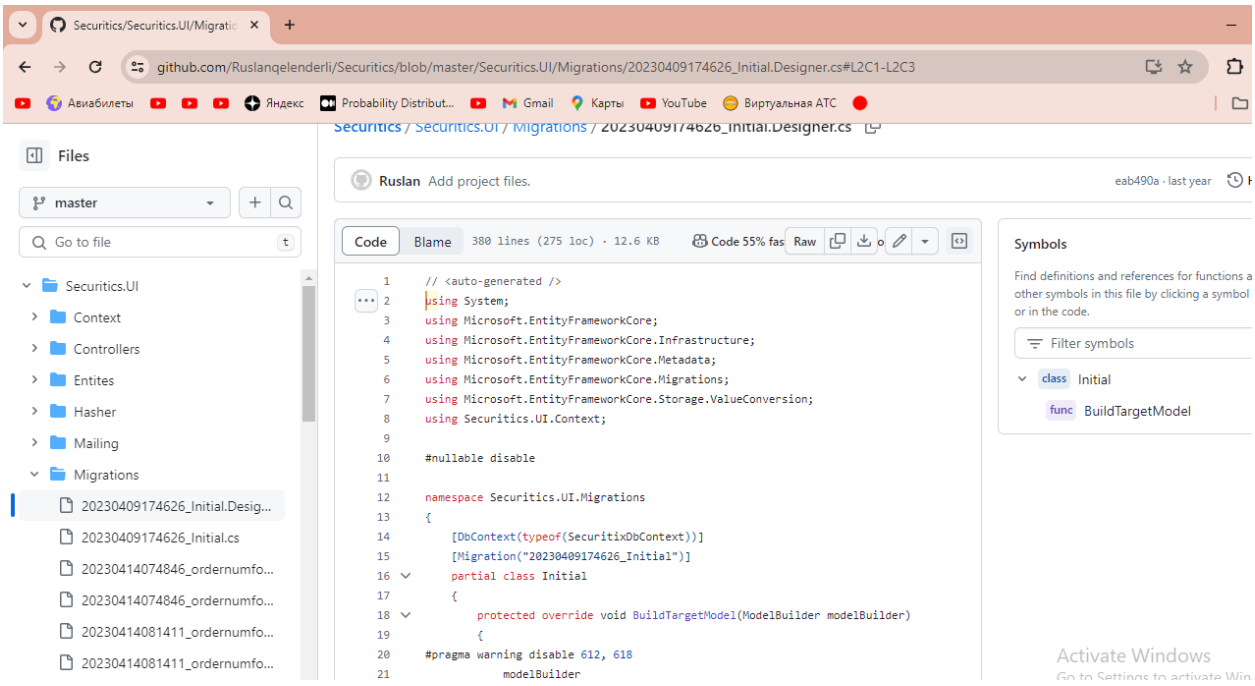
Şək.3.1. Mail ilə Appin əlaqəsi

Şək.3.2. SMTP ilə kontakın əlaqəsi

Saytda olan başlıqların necə hazırlandığını aşağıda göstəririk:



Şək.3.3.Saytın seçimlərinin yazılması



Şək.3.4. Miqrasiyanın avtomatlaşdırılması

Deploy ve Load Balans

Bir idarə olunan Kubernetes xidməti DigitalOcean Kubernetes (DOKS) adlanır. DigitalOcean yük balanslaşdırıcıları və həcmləri, yüksək əlçatanlıq, avtomatik ölçmə və tam idarə olunan idarəetmə müstəvisi ilə yerli inteqrasiya ilə Kubernetes klasterlərini quraşdırın. DigitalOcean API və CLI, eləcə də ümumi Kubernetes alət zəncirləri DOKS klasterləri ilə uyğun gəlir.

Tətbiqi bir Kubernetes klasterindən digərinə köçürdükdə onunla birlikdə gələn yük balanslaşdırıcı xidmətini köçürə bilərsiniz. Bunu etməklə, yük balanslaşdırıcısının tətbiqinizin DNS qeydi ilə göstərilən xarici IP ünvanı qorunur (Cillium Hubble, 2024). Yük balanslaşdırıcısı əvvəlcə öz xidmətini cari xidmətdən imtina etməklə və sonra onun ID-sinə yeni klasterin xidmətində istinad etməklə köçürülə bilər. Yük balanslaşdırıcının yaradılması, yenilənməsi və silinməsi kimi xidmət vasitəsilə idarə olunan bütün mutasiya əməliyyatları, yük balanslaşdırıcısı mövcud xidmətdən imtina edildikdə əməliyyatsız olur.

Yük balanslaşdırıcısını bir klasterdən digərinə köçürmək üçün aşağıdakı prosedurdan istifadə edin. Tətbiq adlı yük balanslaşdırıcı xidmətini istehsal-v1 klasterindən istehsal-v2 adlı fərqli klasterə köçürmək istədiyinizi fərz edin.

Cari xidmətinizin yük balanslaşdırıcısı ilə əlaqəli səhv hadisələrinin olmadığına əmin olmaq üçün aşağıdakı əmrdən istifadə edin: (Cillium Hubble, 2024).

kubectl xidməti <xidmət adı> təsviri

Xidməti sabitləşdirmək üçün hər hansı bildirilmiş problemləri həll edin.

Xidmət konfiqurasiya faylına aşağıdakı annotasiya əlavə edin və yük balanslaşdırıcısını rədd etmək üçün onu doğru olaraq təyin edin: (Rufulllazada Rəhman, 2018).

kind: Service

```
apiVersion: v1
metadata:
name: app
annotations:
  kubernetes.digitalocean.com/load-balancer-id: c16b0b29-217b-48eb-907e-93cf2e01fb56
  service.kubernetes.io/do-loadbalancer-disown: "true"
spec:
  selector:
    name: app
  ports:
    - name: http
    protocol: TCP
    port: 80
  type: LoadBalancer
```

Dəyişiklik qüvvəyə mindikdən sonra xidmət vasitəsilə idarə olunan və yük balanslaşdırıcısına yönəlmiş bütün mutasiyaya uğrayan sorğular nəzərə alınmır.

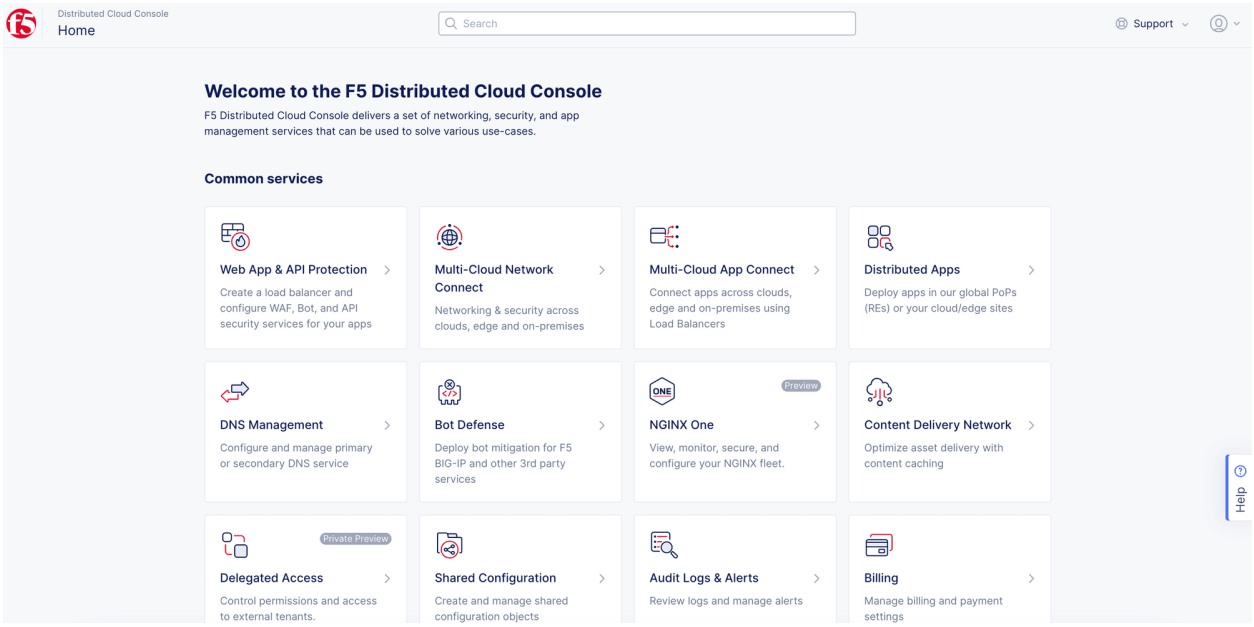
Product-v2 klasterində yeni xidmət konfiqurasiya faylı yaradın və köhnə xidmət konfiqurasiya faylından yük balanslaşdırıcı id-dən istifadə edin (Rufulllazada Rəhman, 2018).

```
kind: Service
apiVersion: v1
metadata:
  name: app
  annotations:
    kubernetes.digitalocean.com/load-balancer-id: c16b0b29-217b-48eb-907e-93cf2e01fb56
```

```
spec:
  selector:
    name: app
  ports:
    - name: http
      protocol: TCP
      port: 80
  type: LoadBalancer  (Rufulllazada Rəhman, 2018).
```

3.3 WAF texnologiyası

Tətbiqlərinizi müdafiə etmək üçün WAF-dan istifadə etmək üçün əvvəlcə F5 Paylanmış Bulud Konsolunda (Konsolda) WAF obyekti qurmalı və onu istifadə etmək istədiyiniz tətbiqi dəstəkləyən HTTP/HTTPS yük balanslaşdırıcısına qoşaraq aktivləşdirməlisiniz (Shikhar Goel, 2024).



Şək.3.5. WAF konsol

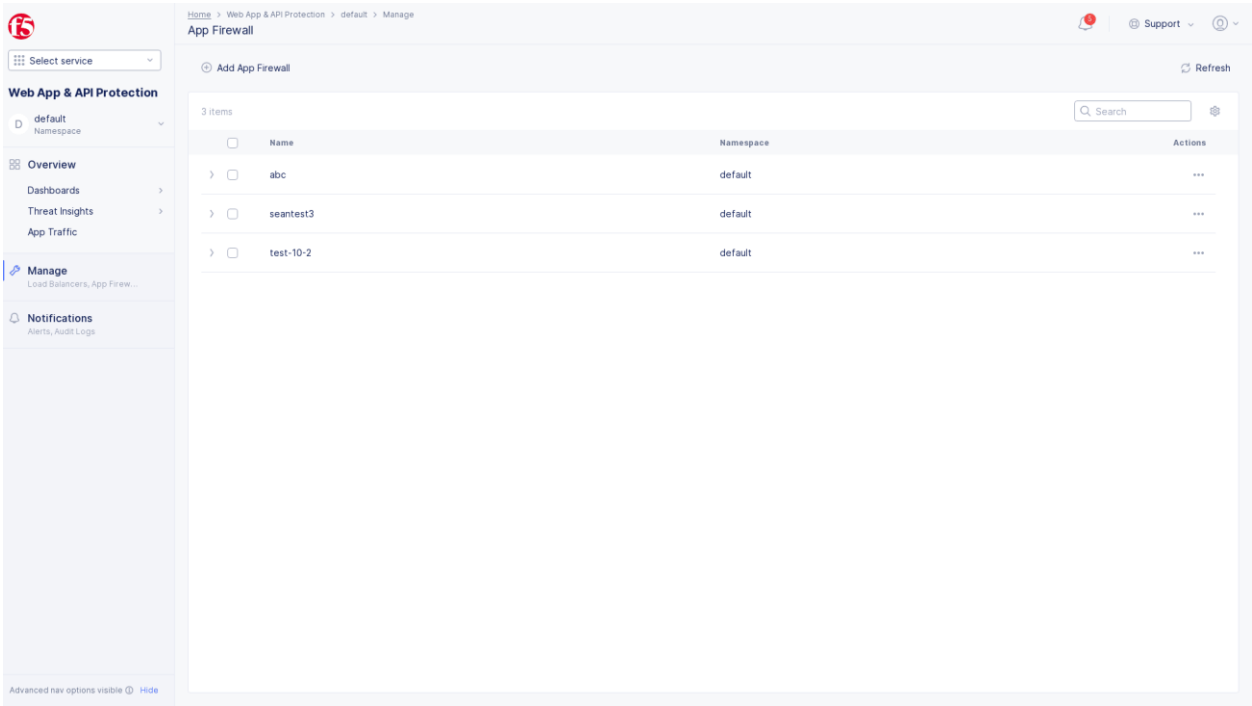
Qeyd: Tətbiq firewallunun yaradılması lazım olan ad məkanında siz onu da yarada bilərsiniz.

1. Konsolun əsas səhifəsindən İdarəetmə xidmətini seçin və sonra Şəxsi İdarəetmə > Ad məkanlarım seçin. Ad əlavə etdikdən sonra Ad sahəsi əlavə et seçin və yadda saxlayın.
2. İdarəetmə > Tətbiq Firewall bölməsinə keçin. WAF yaradılması formasına baxmaq üçün Tətbiq Firewall əlavə et düyməsini klikləyin.
3. Tələb olunan və ulduz (*) ilə göstərilən məlumatları daxil edin:

Metadata bölməsinin altındakı Ad qutusunu doldurараq WAF obyektinə ad verin.

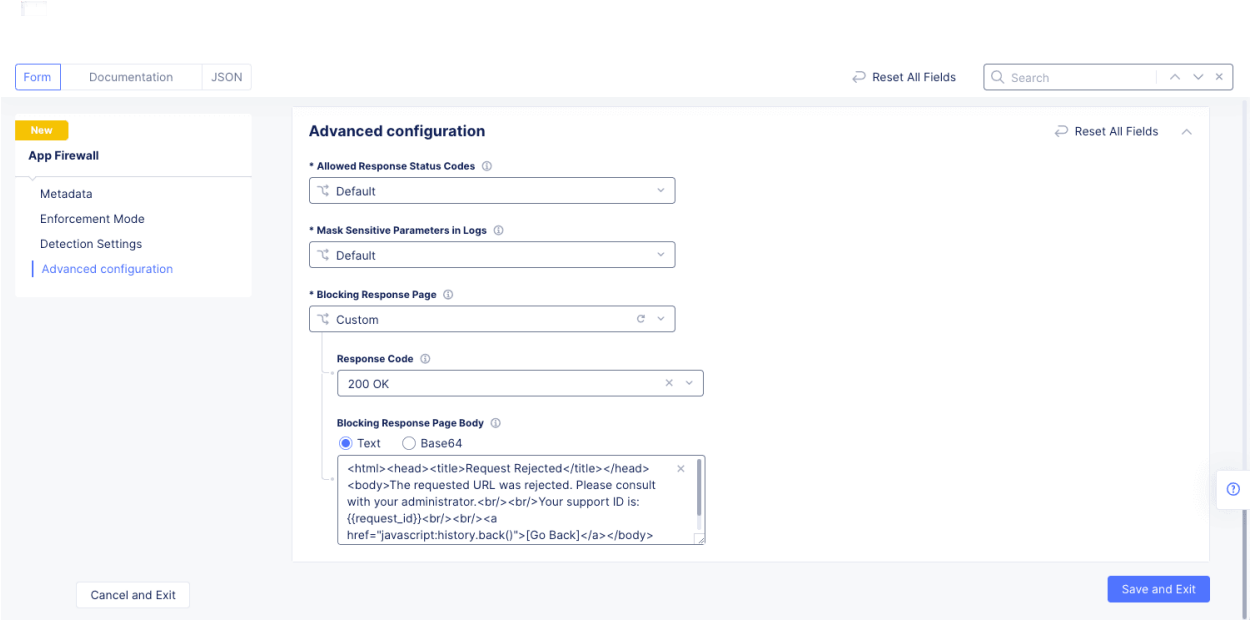
Tətbiq rejimi bölməsində, açılan menyudan istifadə edərək WAF-ın yalnız trafikə nəzarət etməsini və ya qadağan etməsini istədiyinizi seçin.

4. Bloklama: Zərərli trafik qeyd olunmaqla yanaşı qadağan edilir.
5. Monitorinq: Zərərli və şübhəli trafik təhlükəsizlik hadisələri (loqlar) ilə nəticələnsə də, trafik məhdudlaşdırılmır (Whitle Dustin, 2014).



Şək. 3.6. Applikasiya Firewallun qurulması

6. Qabaqcıl konfiqurasiya sahəsində Qabaqcıl Sahələri göstər seçimini aktivləşdirin, sonra aşağıdakı hərəkətləri edin:
7. İcazə Verilən Cavab Vəziyyəti Kodları: Müştərinin istifadəsinə icazə verilən HTTP cavab kodlarının siyahısını təqdim etmək üçün Xüsusi seçin. Bunlardan başqa HTTP cavabları qadağandır.
8. Sorğu Qeydlərində Həssas Parametr Dəyərlərini Gizlət: Defolt parametr həssas parametr dəyərlərini (kredit kartı nömrələri kimi) sorğu qeydlərində görünmədən gizlədəcək. Bu funksiyanı söndürmək və ya öz maskalama parametrlərinizi yaratmaq üçün Xüsusi istifadə etmək seçiminiz var.
9. Xüsusi seçim üçün Element Əlavə et üzərinə klikləyin, HTTP Başlığı, Sorğu Parametri və ya Kuki seçin və adı müvafiq şəkildə daxil edin. Bitirdikdən sonra Element əlavə et seçin. Element əlavə et seçimi birdən çox element əlavə etməyə imkan verir (Kost Edward, 2023).



Şək.3.7. HTTP cavablaını bloklama ayarlari

NƏTİCƏ

Tezis yerli şəbəkələrdə istifadə olunan veb təhlükəsizliyi texnologiyasının tədqiqinə yönəlmişdir. O, həmçinin vebdən istifadə edərək DevOps-un Kuberneti konfiqurasiya etmək və layihələndirmək yolu ilə əsas Veb Təhlükəsizliyini necə yaratmağı izah edir.

Administratorlar qərar qəbul etmək üçün veb təhlükəsizliyi idarəetmə strukturundan istifadə edirlər. İnternetdən icazəsiz girişi aşkarlaya və qarşısını ala bilər Quraşdırma bütün hostlar şəbəkə cihazları vasitəsilə idarə olunurdu.

Burada əsas vəzifə Şəbəkə Təhlükəsizliyi əsasları haqqında məlumat toplamaq və onların xüsusiyyətlərini, istifadə etdikləri protokolları və portları öyrənməkdir.

1-ci fəsildə biz Şəbəkənin təhlükəsizliyindən,qorunma üsullarından, təfərrüatlarına, növlərinə və nüanslarına baxdıq.

2-ci fəsildə müəyyən dərinliklərə enmək simulyasiya zamanı həyata keçiriləcək amilləri izah etmək üçün nəzərdə tutulub.

Burada istifadə olunan proqramlaşdırma dilləri,xüsusiyyətlərini, növlərini, və istifadə etdikləri xidmətləri, portlar, iş prinsiplərini dərindən izah etmək lazım idi.

Beləliklə, hər kodun kiçik nüansları nəzərə alındı. C# və JS-in mütəxəssislərinin həmişə tam başa düşməli olduğu bir məsələdir..

Hansı xidmətin hansı kitabxanada və necə əlaqələndircəyini bilməlidir.

3-cü fəsildə biz bu əməliyyatların praktikada necə olduğunu simulyasiya etdik.

Simulyasiyada WAF və kod,scriptlərdən istifadə edilib və hər bir clusterin parametrləri konfiqurasiya edilib. Simulyasiyası zamanı klusterlər arasındakı əlaqə nəzərdən keçirilmişdir

İSTİFADƏ OLUNMUŞ ƏDƏBİYYAT SİYAHISI

1. Akshay Kedari, (2024). What Is a Website? Understanding the Components and Different Categories. https://www.hostinger.com/tutorials/what-is-website/?ppc_campaign=google_search_generic_hosting_all&bidkw=defaultkeyword&lo=1000998&gad_source=1&gclid=CjwKCAjwouexBhAuEiwAtW_ZxxoksZf1kAH9YefMk3FWc48DQFr5o0sCjD6NU16p3zBrdf-cIXb7hRoCclcQAvD_BwE
2. Alanna Titterrington, (2024). Brute Force Attack: Definition and Examples. https://www.kaspersky.com/resource-center/definitions/brute-force-attack
3. Allan Joaquin, (2023). 10 Practical Applications of JavaScript & Tips for a Successful Career. https://www.simplilearn.com/applications-of-javascript-article#:~:text=JavaScript%20is%20everywhere%20on%20the,job%20as%20a%20web%20developer.
4. Armstrong Steven, (2016). DevOps for networking: Boost your organization's growth by incorporating
5. Boateng Dickson, (2022). A Brief History of JavaScript. https://dev.to/dboatengx/history-of-javascript-how-it-all-began-92a
6. Cillium Hubble, (2024). How to migrate load balancers. https://docs.digitalocean.com/products/kubernetes
7. Deeksha Karkera, (2024). Website Security Definition & How to Keep Your Site Protected. https://www.sitelock.com/blog/what-is-website-security/
8. Deniz Buğa, (2023). Restfull API. https://aws.amazon.com/tr/what-is/restful-api
9. Dhamendra Kewal, (2024). jQuery Tutorial. https://www.geeksforgeeks.org/jquery-tutorial/
10. Fahri Ulutaş, (2024). Nessus nedir ve nasıl çalışır? https://cyberskillshub.com/nessus-nedir-nasil-calisir/

11. Glen Willims, (2024). Application Firewall. https://docs.cloud.f5.com/docs/how-to/app-security/web-app-firewall
12. Gruver Gary, (2016). Start and scaling DevOps in the enterprise. Pennsauken, NJ: BookBaby
13. Gruver Gary, (2020). The devops lifecyle and how devops works. https://about.gitlab.com/topics/devops/
14. Khan İmaran, (2022). What is a website? https://www.linkedin.com/pulse/what-website-explained-detail-imaran-khan/
15. Korkmaz lker, (2022). Framework nedir? https://www.mediaclick.com.tr/blog/framework-nedir
16. Kost Edward, (2023). What are Web Shell Attacks? How to Protect Your Web Servers. https://www.upguard.com/blog/what-are-web-shell-attacks
17. Marcus Fields, (2024). Website Security & Protection: How to Secure a website. https://sucuri.net/guides/website-security/
18. Maricheva Alena, (2023). The history of JavaScript. https://softteco.com/blog/history-of-javascript
19. Padelis Kefalidis, (2024). 7 Security Tips to Protect Your Websites & Web Server From Hackers https://www.firewall.cx/tools-tips-reviews/security-articles/security-tips-how-to-protect-your-websites-and-webservers-from-hackers.html
20. Rufulllazada Rəhman, (2018). C#-ın yaranması və .NET haqqında məlumat. https://medium.com/kodera/c-%C4%B1n-yaranmas%C4%B1-v%C9%99-net-haqq%C4%B1nda-%C3%BCmumi-m%C9%99lumat-6da178620f9e#:~:text=.NET%20Framework%E2%80%99un(.NET,Anders%20Hejlsberg.
21. Shikhar Goel, (2023). Lodash. https://www.g_eeksforgeeks.org/lodash/
22. Shikhar Goel, (2024). React Tutorial. https://www.geeksforgeeks.org/react-tutorial/
23. Shikhar Goel, (2023). Next. Js Tutorial. https://www.geeksforgeeks.org/nextjs/

24. Shikhar Goel, (2023). JavaScript Libraries and Frameworks. https://www.geeksforgeeks.org/javascript-libraries-and-frameworks/

25. Shikhar Goel, (2023). What is website? https://www.geeksforgeeks.org/what-is-a-website/

26. Shikhar Goel, (2024). AngularJS Tutorial. https://www.geeksforgeeks.org/angularjs/

27. Shikhar Goel, (2024). Vue.js Tutorial. https://www.geeksforgeeks.org/vue-js/

28. Sirr Tim Berners-Lee, (2014). https://webfoundation.org/about/vision/history-of-the-web/

29. Sonu Meena, (2024). Devops & Configuration management tools. https://www.slideshare.net/sahilsk/configuration-management-stackexpress-20140610 networking in the DevOps culture. Birmingham, UK: Packt Publishing.

30. Thompson Ken, (2024). The 10 Most Common Website Security Attacks. https://www.tripwire.com/state-of-security/most-common-website-security-attacks-and-how-to-protect-yourself

31. Tim Berners-Lee, (2024). A short history of the web. https://home.cern/science/computing/birth-web/short-history-web

32. Whitle Dustin, (2014). An Introduction to DevOps .https://devops.com/introductiontodevops/

33. Whittle Dustin, (2014). An Introduction to DevOps. Retrieved February 26, 2021

Xülasə

Veb saytlarda olan mümkün bütün hucumların effektiv yollarını aşkar etmək, tətbiq edib bütün təhlükələri aradan qaldırmağı test etdik. Dissertasiyanın əsas məqsədi, ənənəvi davamiyyət metodlarının çatışmazlıqlarını aradan qaldırmaq və elektron sistemlərin üstünlüklərini nümayiş etdirməkdir.

Problemin Təsviri və Əhəmiyyəti

Veb saytlarda olan port açıqlıqlarını və saytda olan bugları bağlamaq, WAF vasitəsilə onları aşkar edib, yeni qaydaları tətbiq etməkdir.

Texnoloji Çərçivə

Dissertasiyada Veb saytların təhlükəsizliyini təmin etmək üçün istifadə olunan texnologiyalar ətraflı araşdırılmışdır:

WAF texnologiyası:

Bütün veb applikasiya,server,saytları qorumaq üçün lazım olan texnologiyadır.Boşluqları aradan qaldırıb,lazım olduqda load-balans etmək xüsusiyyəti daşıyır.

Devops Texnologiyaları:

Tətbiq ömrü boyu geniş təcrübələr DevOps tərəfindən əhatə olunur. DevOps ilə uğur qazanmaq üçün komandalar tez-tez bu üsullardan biri və ya bir neçəsi ilə başlayır.

Versiya idarəetməsi (**Version control**), mənbə koduna və digər fayllara edilən bütün dəyişiklikləri izləmək və nəzarət etmək üçün əsas prosedurdur. Mənbə kodun idarə edilməsi və versiyaya nəzarət bir-biri ilə sıx əlaqəli anlayışlardır.

Veb Tətbiq:

ASP.NET, React Native kimi çərçivələr veb interfeyslərinin yaradılması üçün istifadə olunur.

Bulud Texnologiyaları:

AWS, Google Cloud Platform, Microsoft Azure kimi bulud xidmətləri davamiyyət məlumatlarının uzaq serverlərdə saxlanmasını və işlənməsini təmin edir.

Təhlükəsizlik Texnologiyaları:

Məlumatların təhlükəsizliyini təmin etmək üçün şifrləmə və SSL/TLS protokolları istifadə olunur. OAuth, JWT və LDAP kimi texnologiyalar isə istifadəçi idarəçiliyi və avtorizasiya proseslərini təmin edir.

Praktik Tətbiq və Faydaları

Veb təhlükəsizliyi aradan qaldırmaq hər bir host sahibinə,şirkət,dövlət qurumu və bütün təşkilatlara üstünlüklər qazandırır.

Təhlükəsizlik və Məxfilik: Məlumatların şifrələnməsi və təhlükəsizlik protokolları vasitəsilə davamiyyət məlumatlarının məxfiliyi və təhlükəsizliyi təmin edilir.

Effektiv İdarəetmə: Məlumatların mərkəzləşdirilmiş şəkildə saxlanması və idarə olunması veb serverde nizamnamə proseslərini asanlaşdırır.

Summary

We have tested the effective ways of all possible attacks on web sites, applied them and eliminated all threats. The main goal of the thesis is to overcome the shortcomings of the traditional attendance methods and demonstrate the advantages of electronic systems.

Problem Description and Significance

Blocking open ports and bugs in websites, detecting them through WAF and applying new rules.

Technological Framework

In the thesis, the technologies used to ensure the security of websites are examined in detail:

WAF technology:

It is the technology needed to protect all web applications, servers, and sites.

Devops Technologies:

A wide range of practices throughout the application lifecycle are covered by DevOps. To succeed with DevOps, teams often start with one or more of these methods.

Version control is the basic procedure for tracking and controlling all changes made to source code and other files. Source code control and version control are closely related concepts.

Web Application:

Frameworks like ASP.NET, React Native are used to create web interfaces.

Cloud Technologies:

Cloud services such as AWS, Google Cloud Platform, Microsoft Azure provide storage and processing of attendance data on remote servers.

Security Technologies:

Encryption and SSL/TLS protocols are used to ensure data security. Technologies such as OAuth, JWT, and LDAP provide user management and authorization processes.

Practical Application and Benefits

Eliminating web security brings benefits to every host owner, company, government agency and all organizations.

Краткое содержание

Мы протестировали эффективные способы всех возможных атак на веб-сайты, применили их и устранили все угрозы. Основная цель дипломной работы — преодолеть недостатки традиционных методов посещаемости и продемонстрировать преимущества электронных систем.

Описание и значение проблемы

Блокировка открытых портов и ошибок на веб-сайтах, их обнаружение через WAF и применение новых правил.

Технологическая основа

В диссертации подробно рассматриваются технологии, используемые для обеспечения безопасности веб-сайтов:

Технология WAF:

Это технология, необходимая для защиты всех веб-приложений, серверов и сайтов.

Девопс-технологии:

DevOps охватывает широкий спектр практик на протяжении всего жизненного цикла приложения. Чтобы добиться успеха в DevOps, команды часто начинают с одного или нескольких из этих методов.

Контроль версий — это основная процедура отслеживания и контроля всех изменений, вносимых в исходный код и другие файлы. Контроль исходного кода и контроль версий — тесно связанные понятия.

Веб приложение:

Для создания веб-интерфейсов используются такие фреймворки, как ASP.NET, React Native.

Облачные технологии:

Облачные сервисы, такие как AWS, Google Cloud Platform, Microsoft Azure, обеспечивают хранение и обработку данных о посещаемости на удаленных серверах.

Технологии безопасности:

Для обеспечения безопасности данных используются протоколы шифрования и SSL/TLS. Такие технологии, как OAuth, JWT и LDAP, обеспечивают процессы управления пользователями и авторизации.

Практическое применение и преимущества

Устранение веб-безопасности принесет пользу каждому владельцу хоста, компании, государственному учреждению и всем организациям.