“ AZƏRBAYCAN HAVA YOLLARI ”

QAPALI SƏHMDAR CƏMİYYƏTİ

MİLLİ AVİASİYA AKADEMİYASİ



SƏRBƏST İŞ №4

Fakültə: “Aerokosmik”

İxtisas: “İnformasiya texnologiyalari”

Fənn: “Veb sistemləri və texnologiyaları ”

Mövzu : “Veb proqram kodlarının təhlükəsizliyi”

Qrup: 2441a

Tələbə: Şahmuradov Ilyas

Müəllim: Heydərzadə Nübar

BAKI 2024

**Veb proqram kodlarının təhlükəsizliyi**

Veb kodunun təhlükəsizliyi müasir informasiya dünyasında mühüm rol oynayır. Veb tərtibatçıları saytların müxtəlif növ kiberhücumlardan qorunmasına xüsusi diqqət yetirməlidirlər. Veb saytın sındırılması məxfi istifadəçi məlumatlarının oğurlanmasına, məlumat sızmasına və digər ciddi nəticələrə səbəb ola bilər. Veb kodunun təhlükəsizliyini təmin etmək üçün bir sıra vacib prinsiplərə əməl edilməlidir:

**1. Sanitar proqramlaşdırma**: Təhlükəsiz kodlaşdırma üsullarından istifadə, daxil edilmiş məlumatların süzülməsi, SQL inyeksiya və skript hücumları kimi zəifliklərdən qaçınmaq.

**SQL inyeksiya:** SQL injection zəiflikləri təcavüzkarlara verilənlər bazasında ixtiyari SQL kodunu icra etməyə imkan verir, istifadəçi icazələrindən asılı olmayaraq verilənlərə daxil olmağa, dəyişdirilməyə və ya silinməyə imkan verir. Uğurlu inyeksiya hücumu şəxsiyyətləri saxtalaşdıra, inzibati hüquqlara malik yeni şəxsiyyətlər yarada, serverdəki bütün məlumatlara giriş əldə edə və ya onları yararsız etmək üçün məlumatları məhv edə/dəyişdirə bilər. SQL inyeksiya növlərinə səhvə əsaslanan SQL inyeksiyası, məntiqi səhvə əsaslanan SQL inyeksiyası və zamana əsaslanan SQL inyeksiyası daxildir.

Bu cür hücumun qarşısını almaq üçün SQL sorğusuna ötürülən hər hansı istifadəçi məlumatının sorğunun xarakterini dəyişdirə bilməyəcəyinə əmin olmalısınız. Bunun bir yolu, SQL-də xüsusi məna daşıyan bütün istifadəçi daxiletmə simvollarından qaçmaqdır.

**2. Daimi yeniləmələr**: Məlum zəiflikləri bağlamaq üçün saytın bütün komponentlərini, o cümlədən proqramlaşdırma dillərini, çərçivələri və kitabxanaları müntəzəm olaraq yeniləmək vacibdir.

**3. CSRF və XSS hücumlarına qarşı qorunma:** Saytlararası sorğu saxtakarlığına (CSRF) və saytlar arası skriptlərə (XSS) qarşı qorunma mexanizmlərini tətbiq etmək lazımdır.

**Saytlararası Sorğu Saxtakarlığı (CSRF):** CSRF hücumları təcavüzkarın digər istifadəçinin etimadnaməsini istifadə edərək onun xəbəri və ya razılığı olmadan hərəkətlər etməyə imkan verir. Bu cür hücum ən yaxşı şəkildə bir nümunə ilə izah olunur. Con, müəyyən saytın daxil olmuş istifadəçilərə hesab adı və pul məbləğini özündə əks etdirən HTTP POST sorğusundan istifadə edərək müəyyən hesaba pul göndərməyə imkan verdiyini bilən təcavüzkardır. Con öz bank rekvizitlərini və pul məbləğini gizli sahələr kimi özündə əks etdirən forma yaradır və onu digər sayt istifadəçilərinə elektron poçtla göndərir ("Göndər" düyməsi "tez varlanmaq" saytına keçid kimi maskalanmışdır). İstifadəçi təqdim düyməsini klikləsə, əməliyyat təfərrüatları və brauzerin saytla əlaqələndirdiyi istənilən müştəri tərəfi kukilərindən ibarət serverə HTTP POST sorğusu göndəriləcək (sorğulara saytla bağlı kukilərin əlavə edilməsi brauzerin normal davranışıdır). Server kukiləri yoxlayacaq və istifadəçinin daxil olub-olmadığını və tranzaksiyanı tamamlamaq icazəsinin olub-olmadığını müəyyən etmək üçün onlardan istifadə edəcək.

Bu cür hücumların qarşısını almağın bir yolu, serverin xüsusi sayt üçün istifadəçi tərəfindən yaradılan sirri ehtiva edən POST sorğularını tələb etməsidir. Köçürmələr üçün istifadə edilən veb formanı təqdim edərkən sirr server tərəfindən təmin ediləcək. Bu yanaşma Conun öz formasını yaratmasına mane olur, çünki o, serverin istifadəçiyə verdiyi sirri bilməlidir. Hətta o, sirri tapıb konkret istifadəçi üçün forma yaratsa belə, bundan sonra hər istifadəçiyə hücum etmək üçün eyni formadan istifadə edə bilməyəcək.

**Saytlararası Skriptlər (XSS):** XSS (Saytlararası Skript) təcavüzkarın vebsayt vasitəsilə digər istifadəçilərin brauzerlərinə zərərli kodu yeritməyə imkan verən hücum növünü təsvir etmək üçün istifadə edilən termindir. Yerləşdirilən kod brauzerə saytdan gəldiyi üçün etibarlıdır və istifadəçinin icazə kukisini təcavüzkara göndərmək kimi hərəkətləri yerinə yetirə bilər. Təcavüzkar kukiyəsahib olduqdan sonra o, istifadəçi kimi sayta daxil ola və kredit kartı məlumatlarına daxil olmaq, əlaqə məlumatlarına baxmaq və ya parolları dəyişdirmək kimi istifadəçinin edə biləcəyi hər şeyi edə bilər. XSS zəiflikləri saytın yeridilmiş kodu brauzerə necə qaytarmasından asılı olaraq əks olunan və saxlanılanlara bölünür. Əks olunan XSS zəifliyi serverə təqdim edilən istifadəçi məzmunu brauzerdə nümayiş etdirilmək üçün dərhal dəyişdirilmədən qaytarıldıqda baş verir. Orijinal istifadəçi məzmununda olan hər hansı skript yeni səhifə yükləndikdə işləyəcək. Davamlı XSS ​​zəifliyi zərərli skript vebsaytda saxlandıqda və sonra digər istifadəçilərin istəmədən onu icra edə bilməsi üçün dəyişdirilmədən yenidən göstərildikdə baş verir.

**4. Girişə Nəzarət**: Məlumatı qorumaq üçün autentifikasiya və avtorizasiyadan istifadə edərək həssas məlumatlara girişin məhdudlaşdırılması.

**5. Məlumatların Şifrələnməsi:** Müştəri və server arasında ötürülən həssas məlumatların ələ keçirilməsinin qarşısını almaq üçün şifrələmək vacibdir.

Veb kodunun qorunması daimi diqqət və səy tələb edən bir prosesdir. Müasir təhlükəsizlik standartlarına və düzgün proqramlaşdırmaya riayət etmək bir çox təhlükələrin qarşısını almağa və veb proqramları hücumlardan qorumağa kömək edəcək. Yadda saxlamaq lazımdır ki, hər hansı bir veb layihəsini hazırlayarkən təhlükəsizlik mühüm aspekt olmalıdır.

Veb kodunun təhlükəsizliyi ilə bağlı məlumatlara əlavə olaraq, veb tətbiqinizi təhlükəsiz saxlamağa kömək edə biləcək bəzi əlavə məsləhətlər və üsullar bunlardır:

1. **Çox faktorlu autentifikasiya:** Təhlükəsizliyi yaxşılaşdırmaq üçün çox faktorlu autentifikasiyanı həyata keçirin. Bu, parol, SMS kodu, biometrik məlumatlar və s. birləşməni əhatə edə bilər.

2. **Təhlükəsizlik Monitorinqi**: Potensial təhdidlərə tez cavab vermək üçün sayta giriş qeydlərini müntəzəm olaraq izləmək və şübhəli fəaliyyəti aşkar etmək vacibdir.

3. **Nüfuz sınağı**: Veb tətbiqindəki zəiflikləri təcavüzkarlar tərəfindən istismar edilməzdən əvvəl müəyyən etmək üçün nüfuz testinin keçirilməsi.

4. **Kadr hazırlığı:** Veb tətbiqi ilə işləyən işçilərin informasiya təhlükəsizliyi məsələləri və İnternetdə təhlükəsiz davranış qaydaları üzrə təlimi.

5. **Daimi Məlumat Yedəkləmələri**: Hücum və ya uğursuzluq halında məlumatların bərpa oluna bilməsi üçün saytınızın məlumatlarının müntəzəm ehtiyat nüsxəsini çıxarmaq vacibdir.

Bütün bu tövsiyələrə əməl etmək veb tətbiqinizin təhlükəsizliyini yaxşılaşdırmağa və onu müxtəlif təhlükələrdən qorumağa kömək edəcək. Təhlükəsizlik veb-layihələri inkişaf etdirən və saxlayan hər kəs üçün prioritet olmalıdır.

Həmçinin əlavə əlaqədər qarşılaşa biləcəyiniz digər bəzi təhlükələr haqqında məlumat verək:

**1. Clickjacking:** Bu hücumda təcavüzkar yuxarı səviyyəli görünən sayt üçün nəzərdə tutulan klikləri ələ keçirir və onları aşağıdakı gizli səhifəyə yönləndirir. Bu üsul, məsələn, qanuni bank veb-saytını göstərmək, lakin təcavüzkar tərəfindən idarə olunan görünməz <iframe> daxilində giriş etimadnaməsini ələ keçirmək üçün istifadə edilə bilər. Clickjacking, istifadəçini görünən saytda bir düyməyə klikləmək üçün aldatmaq üçün də istifadə edilə bilər, lakin əslində istəmədən tamamilə fərqli bir düyməni klikləyir. Müdafiə olaraq, saytınız müvafiq HTTP başlıqlarını təyin etməklə özünün başqa saytdakı iframe-ə daxil edilməsinin qarşısını ala bilər.

**2. Denial of Service (DoS):** DoS adətən hədəf saytı saxta sorğularla doldurmaqla əldə edilir ki, qanuni istifadəçilər üçün sayta giriş pozulsun. Sorğular sadəcə olaraq çoxsaylı ola bilər və ya fərdi olaraq böyük miqdarda resurs istehlak edə bilər (məsələn, böyük faylların yavaş oxunması və ya endirilməsi). DoS mühafizəsi adətən qanuni mesajların keçməsinə icazə verərkən “pis” trafiki müəyyən etmək və bloklamaqla işləyir. Bu qorumalar adətən veb-serverin qarşısında və ya üzərində yerləşir (onlar veb proqramın özünün bir hissəsi deyil).

**3. Directory Traversal:** Bu hücumda təcavüzkar veb serverin fayl sisteminin daxil olmamalı olan hissələrinə giriş əldə etməyə çalışır. Bu zəiflik istifadəçi fayl sisteminin naviqasiya simvollarını (məsələn, ../../) ehtiva edən fayl adlarını ötürə bildikdə baş verir. Həll yolu istifadə etməzdən əvvəl girişi təmizləməkdir.

**4. File Inclusion:** Bu hücumda istifadəçi serverə göndərilən məlumatda "təsadüfən" göstərmək və ya icra etmək üçün faylı təyin edə bilər. Yükləndikdən sonra bu fayl veb serverdə və ya müştəri tərəfində icra oluna bilər (nəticədə XSS hücumu). Həll yolu istifadə etməzdən əvvəl girişi dezinfeksiya etməkdir.

**5. İnyeksiya əmrləri:** Komanda inyeksiya hücumları təcavüzkara ana əməliyyat sistemində ixtiyari sistem əmrlərini yerinə yetirməyə imkan verir. Həll istifadəçi daxiletməsini sistem zənglərində istifadə edilməzdən əvvəl sanitarlaşdırmaqdır.

Yuxarıda göstərilən təhlükəsizlik istismarlarının demək olar ki, hamısı veb tətbiqi brauzer məlumatlarına etibar etdikdə uğurlu olur. Veb saytınızın təhlükəsizliyini yaxşılaşdırmaq üçün nə etsəniz də, istifadəçilərdən gələn bütün məlumatları brauzerdə göstərilməzdən, SQL sorğularında istifadə edilməzdən və ya əməliyyat sisteminə və ya fayl sistemi çağırışına keçməzdən əvvəl təmizləməlisiniz. Ata biləcəyiniz bir sıra digər xüsusi addımlar var:

**1.** Daha effektiv parol idarəçiliyindən istifadə edin. Güclü parolların müntəzəm olaraq dəyişdirilməsini təşviq edin. Saytınız üçün iki faktorlu autentifikasiyanı nəzərdən keçirin ki, istifadəçi parola əlavə olaraq başqa identifikasiya kodunu da daxil etməlidir (adətən bir kod yalnız istifadəçinin malik olacağı bəzi fiziki avadanlıq vasitəsilə verilir, məsələn, telefonuna göndərilən SMS-dəki kod. ) .

**2.** Veb serverinizi HTTPS və HTTP Strict Transport Security (HSTS) istifadə etmək üçün konfiqurasiya edin. HTTPS müştəri və server arasında göndərilən məlumatları şifrələyir. Bu, giriş etimadnamələrinin, kukilərin, POST sorğu məlumatlarının və başlıq məlumatlarının təcavüzkarlar üçün asanlıqla əlçatan olmamasını təmin edir.

**3.** Ən populyar təhdidlərə diqqət yetirin (cari OWASP siyahısı buradadır) və ilk növbədə ən çox yayılmış zəiflikləri həll edin.

**4.** Saytınızda avtomatlaşdırılmış təhlükəsizlik testini həyata keçirmək üçün zəifliyin skan alətlərindən istifadə edin.

**5.** Yalnız sizə lazım olan məlumatları saxlayın və göstərin. Məsələn, istifadəçiləriniz kredit kartı məlumatı kimi həssas məlumatları saxlamalıdırlarsa, kart nömrəsinin bir hissəsini yalnız istifadəçi tərəfindən müəyyən oluna bilməsi üçün göstərin və təcavüzkar onu kopyalayıb başqa saytda istifadə edə bilməsi üçün kifayət deyil. Bu gün ən çox yayılmış nümunə kredit kartı nömrəsinin yalnız son 4 rəqəmini göstərməkdir.