****

**Siber Olaylara Müdahale Merkezi (SOME) Birimi**

**ZAFİYETLER - ÇÖZÜM ÖNERİLERİ**

**KILAVUZU v1.0**

**HAZIRLAYAN**

Mehmet CAVLAMAZ

(Birim Sorumlusu)

**05.05.2019**

 **ZAFİYETLER ve ÇÖZÜM ÖNERİLERİ**

**Zafiyet:** SQL Enjeksiyonu

**Açıklama:** SQL Enjeksiyonu, web uygulamasında kullanılan parametrelere ek sorgular ekleyerek arka plandaki veri tabanına erişilmesine ve SQL sorgulamaları yapılmasına olanak tanıyan açıktır. Genellikle web uygulaması üzerindeki formlar veya URL içindeki parametreler kullanılarak bu açıktan yararlanılır.

Eğer web uygulamasının herhangi bir veri tabanı ile bağlantısı varsa ve uygulama üzerinde gerekli önlemler alınmamış ise bu tip saldırılar yapmak mümkün olabilir.

 Örneğin, GET metodu ile veri tabanında yer alan kayıtları, tanımlı sayfaid değerine göre web sayfa çıktısı olarak üreten dinamik bir koda sahip bir web uygulamamız olduğunu düşünelim.

 [www.example.com/index.asp?sayfaid=0001](http://www.example.com/index.asp?sayfaid=0001)

 Yukarıdaki örnekte 0001 değeri web uygulaması tarafından alınacak ve kod içinde tanımlanan SQL ifadesinde çalıştırılarak, arka plandaki veri tabanına sorulacaktır. Veri tabanı 0001 numaralı kaydını yollayarak, ekranda görüntülenmesini sağlayacaktır.

 Eğer uygulama yollanan her türlü veriyi SQL ifadesi içinde kullanacak şekilde tasarlanmış, özel karakterlere göre herhangi bir filtreleme yapmıyor ise gelen her veriyi SQL ifadesi olarak kullanabilir. Bu durumda aşağıdaki bir ifade kullanılarak arka plandaki veri tabanı üzerinde kendi SQL sorgulamalarımızı çalıştırmamız mümkün olabilir.

www.example.com/index.asp?sayfaid=0001’ UNION SELECT sayfa\_id FROM web\_sitesi WHERE '1'='1

 Bir uygulamanın SQL enjeksiyonu açığı içerip içermediğini anlamak için uygulama içindeki formlara veya URL’de bulunan parametrelere ‘, “, \* , or, 1=1, and, having gibi SQL ifadeleri yerleştirmek ve dönen cevabı incelemek yeterli olacaktır. Eğer geriye dönen cevap “Internal Server Error 500” ise veya herhangi bir veri tabanı hatası ise uygulama büyük ihtimalle SQL enjeksiyonu açığına sahiptir.

**Çözüm/Öneri:**

SQL Enjeksiyonu genellikle web uygulaması kodlarındaki problemden ortaya çıkmaktadır. Dolayısı ile söz konusu açıktan korunmanın en kesin çözümü, uygulama geliştirirken güvenli kodlama teknikleri kullanmaktır. Uygulama geliştirirken aşağıda belirtilen yöntemleri uygulamak, güvenli kodlara sahip olmak açısından yardımcı olacaktır.

* Veri tipinin yanı sıra yollanan veri büyüklüğünün de izin verilen aralıkta olup olmadığı kontrol edilmelidir. Bu kontrol özellikle gelen veri uygulama tarafından işlenmeden önce veri tipine göre mutlaka kontrol edilmelidir. Eğer olması gereken haricinde bir veri tipine sahip veri yollanmış ise istek kabul edilmemelidir. Örneğin sayısal bir değer girilen yere alfa numerik karakterler yollanmış ise bu istek kabul edilmemelidir. Veri tipi kontrolü mutlaka yapılmalıdır.
* Buffer OverFlow tipindeki atakların önüne geçecektir.
* Uygulamaya yollanabilecek karakterlerin ne olduğu tanımlanmalı, bunlar dışındaki her karakter işlenmeden önce veri içinden temizlenmeli veya istek komple reddedilmelidir. İzin verilmeyen karakterlerin tanımlanması yerine sadece izin verilen karakterlerin belirlenmesi daha başarılı bir koruma sağlayacaktır.

Benzer şekilde veri tabanı kullanıcısına ait hakların kısıtlanması işletim sistemi üzerinde gerçekleştirilebilecek aktiviteleri sınırlayacağı için, sistemin komple ele geçirilmesini büyük oranda engelleyecektir.

**Ek Bilgi**: <http://www.owasp.org/index.php/SQL_Injection>

**Zafiyet:** Kör SQL Enjeksiyon (Blind SQL Injection)

**Açıklama:** Yetkisiz kişilerin , veritabanı tarafından yazılımsal olarak verilen doğru mu yada yanlış mı cevapları ile veritabanı bilgilerinin alınmasını sağlayan bir açıklıktır . Basitçe bir evet-hayır oyununa benzer .

**Çözüm Önerisi:** Yazılımsal olarak düzeltilmesi gerekmektedir .

**Zafiyet:** Boolean SQL Enjeksiyon (Boolean SQL Injection)

**Açıklama:** Kullanıcı veri girişleri veritabanı tarafından SQL komutları olarak algılanır/yorumlanır.

**Çözüm Önerisi:** Yazılımsal olarak düzeltilmesi gerekmektedir .

**Zafiyet:** Desteklenmeyen Apache versiyonu

**Açıklama:** Çeşitli açıklıklardan faydalanarak web sitesinin/sunucunun ele geçirilmesi sonucunu meydana getirebilir .

**Çözüm Önerisi:** Apache güncel ve en son sürümüne yükseltilmelidir .

**Zafiyet:** HTTP üzerinden oturum açma işlemi - Clear Text Basic (HTTP Üzerinden Login / Clear Text Parola ile Basic) Authentication

**Açıklama:** HTTP üzerinden aktarılan bilgiler herhangi bir şifreleme işlemine tabi tutulmadığı için, ağın dinlenmesi durumda rahatlıkla yetkisiz kullanıcılar tarafından ele geçirilebilir. Bu sebeple login için gerekli kullanıcı adı ve parola bilgilerinin http üzerinden aktarılması büyük bir risk oluşturmaktadır.

**Çözüm Önerileri:** Çözüm önerisi olarak, kritik bilgilere ulaşımda kullanılan her türlü bilgi (Kullanıcı Adı, Parola, SessionID vb.) transfer edilirken mutlaka şifrelenmelidir. Bu sebeple bu tip sistemlere HTTPS gibi SSL şifrelemesi yapan protokoller ile bağlanılması güvenlik açısından yararlı olacaktır. Yine yönetim amaçlı kullanılan sayfalara erişimin kısıtlanması, (IP adresine göre erişim izni verilmesi) karşılaşılabilecek güvenlik risklerini azaltacaktır.

**Ek Bilgi:**

**Asp.NET**

Öncelikle IIS in bulunduğu sunucuya <https://www.iis.net/downloads/microsoft/url-rewrite> eklentisinin kurulması gereklidir.

Kurulumdan sonra uygulamalarınızda webconfig dosyasına aşağıdaki kod satırlarını ekleyip kaydediniz.

<system.webServer>

 <!--Rewrite blogu - Http isteklerini Https'e yönlendiriyor.-->

 <rewrite>

 <rules>

 <rule name="https redirect" enabled="true">

 <match url="(.\*)" ignoreCase="false" />

 <conditions>

 <add input="{HTTPS}" pattern="off" ignoreCase="false" />

 </conditions>

 <action type="Redirect" redirectType="Found" url="https://{HTTP\_HOST}{REQUEST\_URI}" />

 </rule>

 </rules>

 </rewrite>

Linkler İncelenebilir;

<https://justinacton.com/2015/11/14/iis-8-redirect-http-to-https-3/>

<https://blogs.technet.microsoft.com/dawiese/2016/06/07/redirect-from-http-to-https-using-the-iis-url-rewrite-module/>

2- PHP ile geliştirilen uygulamalarınızda .htaccess dosyasına aşağıdaki kod satırlarını ekleyip kaydediniz.

<IfModule mod\_rewrite.c>

 RewriteRule ^(.\*)$ https://domainadı/$1 [R,L]

 RewriteRule ^(.\*)$ www// domainadı/$1 [R,L]

 RewriteRule ^(.\*)$ http:// domainadı/$1 [R,L]

</IfModule>

Domainadı yazan kısma örnek; RewriteRule ^(.\*)$ https://usakism.saglik.gov.tr/$1 [R,L]

**Önemli Hatırlatma!!!** Bakanlığımızın Hosting Platformunu kullanan uygulama geliştiricilerimiz, Web Sitelerinde (uygulamalarında) HTTP’den girişleri HTTPS’e çevirmek için aşağıdaki adımları takip edebilirler.

\* Plesk Yönetim arayüzünden Web Siteleri ve Alan Adları >>> Barındırma Ayarları >>> Güvenlik >> SSL Desteği Açık işaretlenmelidir.

 Sitenizde kontrolleri sağladığınızda hem HTTP hem de HTTPS üzerinden erişim sağladığınızı göreceksiniz. Her iki şekilde erişimde zafiyet ortadan kalkmaz. Bu durum da HTTP isteklerinizi HTTPS’e çevirmeniz gereklidir.

**Zafiyet:** Desteklenmeyen PrettyPhoto versiyonu

**Açıklama:** Çeşitli açıklıklardan faydalanarak kullanıcı adı ve parola bilgisinin ele geçirilmesi sonucunu meydana getirebilir .

**Çözüm Önerisi:** PrettyPhoto güncel ve en son sürümüne yükseltilmelidir.

**Zafiyet:** Desteklenmeyen PHP versiyonu

**Açıklama:** Çeşitli açıklıklardan faydalanarak web sitesinin/sunucunun ele geçirilmesi sonucunu meydana getirebilir .

**Çözüm Önerisi:** PHP güncel ve en son sürümüne yükseltilmelidir.

**Zafiyet:** Cross Site Scripting (XSS) (Siteler Arası Betik Çalıştırma)

**Açıklama:**

Dinamik olarak oluşturulan web sayfalarında, kullanıcı tarafından sağlanan girdilerin uygulama tarafından düzgün bir şekilde işlenemediği veya kontrol edilmediği durumlarda, yetkisiz kişiler istedikleri script kodlarının, görüntülenecek sayfada çalışmasını sağlayabilirler. Söz konusu problem Cross Site Scripting (XSS) olarak adlandırılır.

Bu problem ile daha çok aşağıdaki durumlarda karşılaşılır:

* Girilen arama metnini tekrar tarayıcıda görüntüleyen arama motorları,
* Hataya yol açan metni tekrar eden hata mesajları,
* Girilen değerlerin sonradan kullanıcıya görüntülendiği formlar,
* Kullanıcıların kendi mesajlarını yayınlayabilmelerine izin veren sayfalar

XSS açığı ile saldırgan, HTML kodlarının arasına istemci tabanlı kod ekleyerek kullanıcıların tarayıcısında istediği istemci tabanlı kodu çalıştırabilir. Bu açık sayesinde saldırgan, açığın bulunduğu domain ve sayfanın bilgilerini, oturum bilgilerini ve diğer obje değerlerini çalabilir. Örneğin, bir kullanıcının oturum bilgilerini çalan saldırgan o kullanıcının yetkisi dâhilindeki her işlemi yapabilir.

**Çözüm/Öneri:**

Bu saldırıları önlemek için birkaç yol mevcuttur. Bu yöntemler kullanıcı girdilerinin (URL, form vs.) filtrelenmesi esasına dayanmaktadır. Kullanıcıların girdilerindeki zararlı karakterler filtrelenmelidir. Bu yöntem anı zamanda SQL enjeksiyonu, uzaktan komut çalıştırma, bellek taşması gibi diğer ataklarını da engeller.

Aşağıda XSS için alınabilecek önlemlere değinilmiştir.

* Değişik parametreler aracılığı (cookie, formlar, http başlığı, kod içindeki gizli (hidden) alanları sorgu parametreleri) ile GET ve POST metodu ile yollanan veriler içinde nelere izin verilmeyeceği mutlaka belirlenmelidir.
* Oluşturulan çıktı mutlaka kontrol edilmeli ve uygun şekilde filtrelenmelidir.
* XSS için kullanılabilecek <,>,/”, (gibi tüm özel karakterler istek içinden silinmelidir. İstek içinde yollanacak veri sadece alfa numerik karakterler olacak şekilde düzenlendikten sonra uygulamaya yollanmalıdır.
* Üretilen çıktıda özel karakterlerin silinmesi için değişik uygulamalar tarafından geliştirilmiş fonksiyonlar kullanılabilir.

Örneğin PHP tabanlı uygulamalarda **htmlspecialchars** ASP.NET tabanlı uygulamalarda ise **HTMLEncode** fonksiyonları bu amaçla kullanılabilir.

PHP: string htmlspecialchars (string string [, int quote\_style])

ASP / ASP.NET: Server.HTMLEncode (strHTML String)

 Çok sayıda uygulamada benzer problemlerin olması, kodların kısa bir süre içinde güvenlik gereksinimlerini karşılayacak şekilde değiştirilmesini oldukça güçleştirebilir.

 Bu sebeple web uygulamasına yönelik güvenlik yazılımları kullanmak, kodlar değiştirilmeden de belirtilen açıklara karşı önlem almanıza yardımcı olacaktır. Buna ek olarak sunucu hata mesajlarının standart hale getirilmesi de söz konusu açıktan doğacak riski minimuma indirecektir.

Aşağıdaki karakterlerin filtrelenmesi tavsiye edilmektedir:

[1] | (pipe)

[2] & (ampersand)

[3] ; (noktalı virgül)

[4] $ (dolar)

[5] % (yüzde)

[6] @ (at)

[7]' (tek tırnak)

[8] " (çift tırnak)

[9] \' (backslash-tek tırnak)

[10] <> (üçgen parantezler)

[11] () (parantezler)

[12] + (artı)

[13] CR (Carriage return, ASCII kodu: 0x0d)

[14] LF (Line feed, ASCII kodu: 0x0a)

[15] , (virgül)

[16] \ (backslash)

[17] \" (backslash-çift tırnak)

**Ek Bilgi:** [https://www.owasp.org/index.php/Cross-site\_Scripting\_(XSS)](https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29)

 **Detaylı anlatım:** [**https://www.gokhankesici.com/x-xss-protection/**](https://www.gokhankesici.com/x-xss-protection/)

**Zafiyet:** Secure (Güvenli) olarak işaretlenmemiş çerez tespiti

**Açıklama:** Çerezler (Cookies), istemci tarafında web uygulamasıyla ilgili herhangi bir bilgiyi, oturum numarasını vs. saklamaya yarayan ve sadece kendisinin yollandığı alan adı (domain) için bilgilerini veren bir araçtır. HTTPS üzerinden yollanan çerezler eğer “Secure” olarak işaretlenmez ise HTTP üzerinden şifresiz olarak da gönderilirler. Bütün bir uygulama boyunca oturum işlemlerinin geçtiği her adımda HTTPS kullanılan bir sistemde çerezleri “Secure” olarak işaretlememek, bir saldırganının oturum bilgisini HTTP üzerinden elde etmesi ile sonuçlanabilir. Ağ iletişiminde araya girilerek çerezlerin elde edilmesi sonucunu meydana getirebilir.

**Çözüm Önerisi:** Çerezin Secure olarak işaretlenmesi tavsiye edilmektedir. Bu önlem man-in-the-middle ataklarına karşı koruma sağlamak için önemlidir. Eğer bütün uygulama kapsamında HTTPS üzerinden şifreli bir bağlantı gerçekleştiriliyorsa, HTTP başlıklarındaki Set-Cookie özelliğine Secure deyimi eklenmelidir.

**Ek Bilgi:**

**IIS web.config**

Aşağıdaki kodları **web.config** altına ekleyiniz.

|  |  |
| --- | --- |
| 123 | <system.web><HttpCookies httpOnlyCookies="true" requireSSL="true" lockItem="true" /></system.web> |

**Apache (Versiyon: 2.2.4 ve üstü)**

* + “mod\_headers.so” modülünün etkin olması gerekmektedir .
	+ “httpd.conf” dosyasına şu satırı eklemelisiniz:

Header edit Set-Cookie ^(.\*)$ $1;HttpOnly;Secure .

* + Apache’yi yeniden başlatın.

**Apache (Versiyon: 2.2.4 altındaki)**

* + “httpd.conf” dosyasına şu satırı eklemelisiniz:

Header set Set-Cookie HttpOnly;Secure.

* + Apache’yi yeniden başlatın.

Nginx:

Having **HTTPOnly** and **Secure** in HTTP response header can help to protect your web applications from cross-site scripting and session manipulation attacks.

There are 2 (two) ways to **configure HTTPOnly Secure Cookie** Attribute in **Nginx**.

## **By using proxy\_cookie\_path**

Add the following directives to  ssl.conf or default.conf

|  |  |
| --- | --- |
| 1 | proxy\_cookie\_path / "/; HTTPOnly; Secure"; |

Restart the Nginx to see the results.

## **By using “nginx\_cookie\_flag\_module” Module**

A **Nginx module** called **nginx\_cookie\_flag** by Anton Saraykin let you quickly set cookie flag as **HTTPOnly** and **Secure** in Set-Cookie HTTP response header.

To implement **Secure Cookie** by this way you need to build **Nginx** from the source code by adding the module.

Add this flag to your configure directives:

|  |  |
| --- | --- |
| 1 | --add-module=/path/to/nginx\_cookie\_flag\_module |

Once **Nginx** is built with the above module, you can add the following line either in location or serverdirective in respective configuration file

|  |  |
| --- | --- |
| 1 | set\_cookie\_flag HttpOnly secure; |

Restart **Nginx** to verify the results.

## Verify **HTTPOnly Secure Cookie**

To verify that changes have been applied you can use “Developer Tools” in Chrome or Firefox to examine the request headers.

**Zafiyet:** SSLv2 desteği tespit edilmesi CVE-2016-0800 (Drown Attack-Decrypting RSA using Obsolete and Weakened Encryption)

**Açıklama:** SSLv2 protokolünün açıklığından faydalanarak şifrelenmiş bir trafikte araya girerek şifreyi çözmesine olanak sağlar .

**Çözüm Önerisi:** SSLv2 protokolünün sunucu tarafında devre dışı bırakılması ya da SSL yerine TLS kullanılması önerilmektedir. SSLv3 POODLE zafiyetindeki çözüm önerisindeki adımlar yapılmalıdır.

**Ek Bilgi**: <https://drownattack.com/>

**Zafiyet:** SSLv3 POODLE (Padding Oracle On Downgraded Legacy Encryption)

**Açıklama:** Zafiyet bir çeşit kriptografik tasarım zafiyetidir. Zafiyet esas olarak SSLv3’ün kripto bloğunun sonuna eklenen padding bitlerinin içeriğini belirlememesi ve önce deşifre sonra kimlik doğrulama işlemini yapmasından kaynaklanmaktadır. Bu açık sayesinde bir saldırgan 1/256 ihtimalle SSLv3 ile şifrelenmiş trafiği başarılı bir şekilde deşifre edebilir. Bu ihtimal bir byte içindir, dolayısıyla toplamda N byte’lık bir veriyi deşifre etmek için ortalama 256\*N adet deneme yapmak gerekmektedir. Zafiyetin en uygulanabilir saldırı vektörü MITM (man in the middle) durumudur. İstemci ile sunucu arasındaki trafiğin kendi üzerinden geçmesini sağlayan saldırgan, SSL el sıkışması adımlarına müdahele edip bağlantıda SSLv3 kullanılmasını zorlayabilir. Tabi istemci ve sunucunun, daha güncel protokollerin yanısıra SSLv3’ü de destekliyor olması gerekmektedir.

**Çözüm Önerisi:** SSLv3 kullanımı devre dışı bırakılıp daha yüksek bitlerde şifreleme algoritmaları kullanılmalıdır.

**Windows:**

Kayıt Defteri (Registry) açıyoruz.

Başlat > Çalıştır > regedit

Aşağıda ki dizine gidiyoruz ;

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\Schannel\Protocols\



Protocols sağ tıklayıp **New > Key** diyoruz.



Yeni Key’e **SSL 3.0** ismini veriyoruz.

**SSL 3.0** sağ tıklayıp tekrardan **New > Key** diyoruz.



Oluşturduğumuz key’in adını **Client** olarak değiştiriyoruz.

**SSL 3.0** altında tekrardan **New > Key** diyoruz.



Oluşturduğumuz keyin adını **Server** olarak değiştiriyoruz.

Client sağ tıklayıp **New > DWORD (32-bit) Value** diyoruz ve bir dword kaydı oluşturuyoruz.



Oluşturduğumuz Dword kaydının adını **DisabledByDefault** olarak değiştiriyoruz.



**DisabledByDefault** kaydının **Decimal** değerini **1** yapıyoruz.

Bu işlemi yaptıktan sonra **Server** sağ tıklayarak **New > DWORD (32-bit) Value** kaydı ekliyoruz.



Eklediğimiz DWORD (32-bit) Value kaydını editleyip ismini **Enabled** olarak değiştiriyoruz.



Enabled kaydının Decimal değerinin 0 olduğundan emin oluyoruz.

Bu işlemleri yaptıktan sonra sunucunuzu tekrar başlatarak işlemi tamamlayınız.

**Apache:**

SSL v3.0 ve v2.0 kapatmak için;

SSLProtocol All -SSLv2 -SSLv3

Bu komut SSL v2.0 ve SSL v3.0’ı protokollerini kapatacaktır. Apache’deki şifreleme standartı **en az** TLS 1.1 üzerinden yapılması tavsiye edilmektedir. Eğer web server’da bu implementasyon yapılmamışsa TLS desteği vermeniz önerilir.

Son kontrolleri aşağıdaki komut ile kontrol ediyoruz.

apachectl configtest

ardından restart ediyoruz.

sudo service apache2 restart

**Ngix:**

Nginx’de de SSL v3.0 desteğini kapatmak oldukça basittir. Hangi protokolleri kullanacağımızı söylemek  ve ardından servisi restart etmek yeterlidir. Ngix’deki şifreleme standartı **en az** TLS 1.1 üzerinden yapılması tavsiye edilmektedir

ssl\_protocols TLSv1.1 TLSv1.2;

sudo nginx -t

sudo service nginx restart

**Zafiyet:** Zayıf Şifreleme Algoritmasının Tespit Edilmesi

**Açıklama:** **Şifreleme,** sadece doğru şifreleme anahtarını bilen bir kişi tarafından düzenleme veya okuma işlemlerini gerçekleştirmek için, ileti veya dosyanın içeriğini karıştırarak güvenliğini artırmanın bir yoludur. Günümüzde şifreleme yöntemleri, matematiksel temelli hesaplamalara dayanan bilgisayar, elektronik, fizik gibi birçok bilim dalını ilgilendiren bir sistematiğe bürünmüştür. Şifreleme ve şifre çözme işleminde kullanılan bir çok algoritma, teknik ve yaklaşım bulunmaktadır.

**Çözüm Önerisi:** Sistemde MD5 ve SHA1 gibi zayıf şifreleme algoritmalarının kullanılmaması için yeniden konfigüre edilmesi önerilmektedir.

**Windows:**

Bununla birlikte **DES, 3DES, RC4** desteğinin de kapatılması gereklidir;

Kayıt Defteri (Registry) açıyoruz.

Başlat > Çalıştır > regedit

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers]

Dizinine gidiyoruz.

**Chiphers** sağ tıklayarak **New > Key**diyoruz ve yeni eklediğimiz Key’in adını **RC4 128/128** olarak değiştiriyoruz.

**RC4 128/128** sağ tıklayıp **New > DWORD (32-bit) Value** diyoruz ve eklediğimiz kaydın adını **Enabled** olarak değiştirip **Decimal** değerinin **0**olduğundan emin oluyoruz.

Aynı işlemi ;
DES 56/56
RC4 40/128
RC4 56/128

RC4 64/128

Triple DES 168/168

Key’leri içinde yapıyoruz, yukarıdaki keyleri oluşturup Enabled Dword ekleyip Decimal değerini 0 yapıyoruz.

Sunucuyu restart ederek işlemimizi tamamlıyoruz.

**Ngix ve Apache**

[**https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/**](https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/)

[**https://www.acunetix.com/blog/articles/tls-ssl-cipher-hardening/**](https://www.acunetix.com/blog/articles/tls-ssl-cipher-hardening/)

**Zafiyet:** Desteklenmeyen Jquery tespit edilmesi

**Açıklama:** Jquery bir javascript kütüphanesidir ya da bir javascript frameworküdür denilmektedir. Daha çok sayfalardaki animasyonlar için Flash alternatifi olarak kullanılan bu teknoloji ile, slaytlar, foto galeriler, tab menüler yapmak oldukça kolaydır.

**Çözüm Önerisi:** Desteklenen ve en güncel sürüme yükseltilmesi önerilmektedir.

**Ek Bilgi**: <http://jquery.com/download/>

**Zafiyet:** Desteklenmeyen Jquery UI Dialog tespit edilmesi

**Açıklama:** Sitemizde bir uyarı vereceğimizde .alert() kullanırız genellikle tasarımları sitemize uymaz yada güzel değildir. Alert kullanmak yerine kendimiz diyalog kutusu yapabiliriz. Jquery UI de bunun için kolay ve özelliştirmelere serbest bırakmayı sağlar.

**Çözüm Önerisi:** Desteklenen ve en güncel sürüme yükseltilmesi önerilmektedir.

**Ek Bilgi**: <http://jqueryui.com/download/>

**Zafiyet:** Desteklenmeyen Angular JS tespit edilmesi

**Açıklama:** Google’nin desteklediği, javascript MVC kütüphanesidir. MVC (Modal View Controller) verinin, görünümün ve kontrollerin ayrı tutulmasına dayanır.

**Çözüm Önerisi:** Desteklenen ve en güncel sürüme yükseltilmesi önerilmektedir.

**Ek Bilgi**: <https://angularjs.org/>

**Zafiyet:** Query String üzerinden gönderilen parola tespiti

**Açıklama:** QueryString ASP.Net te sayfalar arasında veri transfer etmeye yarar. ASP.Net te sayfalar arası veri transfer etmek için birden fazla yol vardır. Bu yollardan bir taneside QueryString ‘dir. Fakat bu yöntemi kullanmanızı pek tavsiye edilmemektedir. Güvensiz bir yöntem olduğu için gönderdiğimiz veriler tarayıcı çubuğunda açıkça gözükmektedir.

**Çözüm Önerisi:** Parolaların query string üzerinden gönderilmemesi önerilmektedir. Bu önlem man-in-the-middle ataklarına karşı koruma sağlamak için önemlidir.

**Zafiyet:** Desteklenmeyen Jquery Migrate tespit edilmesi

**Açıklama:** Yeni bir değişiklik çıktığında JQuery gibi kütüphanelerinin versiyonlarının yükseltilmesi (upgrade) çok iş yükü getirebilir. jQuery Migrate, kaldırılan API'leri geri yükleyerek bunu daha kolay hale getirir ve ek olarak kaldırıldığında ve / veya kullanımdan kaldırılan API'lar kullanılırken tarayıcı konsolunda uyarılar gösterir. (yalnızca jQuery Geçiş sürümünü geliştirir.)

Bu şekilde, artık jQuery Migrate'a uzun süre ihtiyaç duymadan ve onu kaldırarak hataları düzeltebilirsiniz.

**Çözüm Önerisi:** Desteklenen ve en güncel sürüme yükseltilmesi önerilmektedir.

**Ek Bilgi**: <http://jquery.com/download/>

**Zafiyet:** Geçersiz SSL Sertifikası Tespit Edilmesi

**Açıklama:** SSL'in açılımı Secure Sockets Layer (güvenli giriş katmanı) dır. SSL, server ile alıcı iletişimi esnasında verilerin şifrelenerek yapılması işlemidir. En bilinen kullanımı ise, web sitesindeki veri alışverişi esnasında, server ile internet tarayıcısı arasındaki iletişimi şifrelenmesidir.

SSL, standart bir algoritmadan oluşmaktadır. Güvenli veri iletişimi için birçok web sitesi SSL teknolojisini kullanmaktadır. SSL işleminin çalışması için server tarafında bir anahtar ve alıcı tarafında çalışacak bir sertifika olması gerekmektedir.

**Çözüm Önerisi:** Güncel Sertifikanın alınması gereklidir.

**Ek Bilgi**: Bakanlığımız barındırma (hosting) hizmetinden yararlanan kurumlarımızın web sitelerinde bir veri girişi alanı (Kullanıcı adı ve şifre vb…) bulunuyorsa Plesk Panel üzerinden sertifika desteğini aktif etmelidir. Bununla birlikte uygulamaları Bakanlığımız sunucularında barındırılıyorsa , Sistem Yöneticilerinden sunucularına sertifika kurulumunu talep edebilir. Sertifikanın geçersiz olması durumları geliştiricilerin değil Sistem Yöneticilerinin sorumluluğundadır.

**Zafiyet:** HSTS (HTTP Strict Transport Security) Başlığının Kullanılmaması

**Açıklama:** HTTP Strict Transport Security (HSTS) kullanıcıların internet tarayıcılarını her talep için HTTPS kullanmaya zorladığı için downgrade adı verilen saldırılara karşı tüm trafiğin güvenliğini sağlamak için kullanılmalıdır.

**Çözüm Önerisi:** HSTS'i aktif etmek için aşağıdaki satırı HTTP cevabının başlık bilgisine eklenmelidir.

**Ek Bilgi**:

***Strict-Transport-Security: max-age=16070400;***

Bu sayede HSTS belirlenen domain veya subdomain için HTTPS everywhere aktif hale getirilmiş olacaktır. Bazı durumlarda www.domain.com için tanımlama yapılıp tüm subdomainler dahil HSTS ön tanımlı olarak gelmesi istenildiğinde aşağıdaki eklemenin yapılması gereklidir.

***Strict-Transport-Security: max-age=16070400; includeSubDomains***

Eğer sertifikanın Chrome, Firefox gibi tarayıcılar tarafından ön tanımlı olarak kontrol listesinde yer almasını istiyorsanız, aşağıdaki düzenlemeyi yapıldıktan sonra https://hstspreload.appspot.com adresinden web site girilerek ücretsiz başvuru yapabilirsiniz.

***Strict-Transport-Security: max-age=10886400; includeSubDomains; preload***

(Not: max-age değeri 6 hafta olarak önerilmektedir.)
Domain için HTTPS hizmet verilecek ise HSTS kullanılmalıdır. Tüm bu HTTP başlık bilgilerini kod katmanında yapmak yerine Apache/IIS/Nginx/Tomcat vb… web sunucu uygulamaları üzerinde yapmanız gereklidir.

**Apache'de HSTS aktifleştirmek için;**

***# Apache HSTS
LoadModule headers\_module modules/mod\_headers.so***

***<VirtualHost \*:443>***

 ***Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"***

***</VirtualHost>***

**Nginx'de HSTS aktifleştirmek için;**
***add\_header Strict-Transport-Security max-age=63072000; includeSubdomains***

**IIS web.config ile HSTS aktifleştirmek için**
*<system.webServer>*

 *<httpProtocol>*

 *<customHeaders>*

 *<add name="Strict-Transport-Security" value="max-age=31536000"/>*

 *</customHeaders>*

 *</httpProtocol>*

*</system.webServer>*

**Zafiyet: robots.txt** Bilgi Açığı Ortaya Çıkarılması

**Açıklama:** Arama motorlarının indeksleme robotları bir web sitesine ulaştıklarında ilk olarak kök dizinde bulunan robots.txt dosyasını ararlar. Bu dosya robotlara hangi sayfaların indekslenip hangilerinin indekslenmeyeceğini anlatmak için yerleştirilir. Bununla birlikte **robots.txt** dosyası herkes tarafından okunabileceği için yetkisiz kullanıcılara web sunucu üzerindeki dosya yapısı hakkında da detaylı bilgi sunabilir.

**Çözüm Önerisi:** Güvenlik açısından bakıldığı zaman **robots.txt** dosyasının kullanımı riskli görülmektedir. Eğer **robots.txt** kullanımı gerekiyorsa dışarıdan görülmesini istemediğiniz bilgiler alt dizinler altına kopyalanarak koruma sağlanabilir.

Örneğin dizin/sifre.txt, dizin/onemli\_dosyalar/ gibi bilgilerin saklanması isteniyorsa robots.txt dosyası aşağıdaki gibi oluşturulabilir.

User-agent: \*

Disallow: /dizin/

Bir başka yöntem ise saklanmak istenen bilgilerin başına belirli ifadeler ekleyip, bu ifadeyle başlayan tüm dosyaların indekslenmesini engellemek olabilir.

Örneğin sifre.txt dosyasını giz\_sifre.txt gibi adlandırıp robots.txt dosyasını aşağıdaki gibi oluşturmak yeterli olacaktır.

User-agent: \*

Disallow: /giz\*

**Zafiyet:** HTTP TRACE/TRACK

**Açıklama:** HTTP TRACE/TRACK metodları istenilen sayfanın aynısının web sunucu tarafından gönderilmesi sağlanarak hata ayıklamak için kullanılmaktadır. TRACE/TRACK yöntemini destekleyen sistemler XST (Cross Site Tracing) saldırılarına maruz kalabilmektedir.

Etkilenen sunucu sistemleri HTTP TRACE ya da TRACK erişim yöntemlerinden en az birisini desteklemektedir.

**Çözüm Önerisi:** Yalnızca gerekli HTTP metodları kullanılmalıdır.

**Ek Bilgi**:

**IIS web.config aşağıdaki ayarlar yapılmalıdır.**
***<system.webServer>***

 ***<httpProtocol>***

 ***<customHeaders>***

 ***<add name="Allow" value="GET,POST,HEAD" />***

 ***<add name="Public" value="GET,POST,HEAD" />***

 ***</customHeaders>***

 ***</httpProtocol>***

***</system.webServer>***

**Zafiyet:** Web Uygulama ve Web Sunucu Sürümlerinin Açığa Vurulması

**Açıklama:** HTTP yanıt paketleri içerisinde web uygulaması ve web sunucusu sürümlerinin açığa vurulduğu tespit edilmektedir. Bu bilgilerden faydalanan bir saldırgan hedefe ait bilgi toplayabilir ve saldırılarını özelleştirebilir.

**Çözüm Önerisi:** Uygulama ve sunucu sürümlerinin HTTP cevap paketlerinde aşağıda detayları verilen sürüm bilgileri barındırılmamalıdır.

Server – Web Server versiyonunu belirtir.

X-Powered-By – Web Sitesininde geliştirildiği mimariyi gösterir. "powered by ASP.NET.”

X-AspNet-Version – Geliştirilirken kullanılan ASP.NET sürümünü belirtir.

**Detaylar için;**

<https://www.acunetix.com/blog/articles/configure-web-server-disclose-identity/>

<https://blogs.msdn.microsoft.com/varunm/2013/04/23/remove-unwanted-http-response-headers/>

[**https://www.saotn.org/remove-iis-server-version-http-response-header/**](https://www.saotn.org/remove-iis-server-version-http-response-header/)

**Ek Bilgi**:

**1- X-AspNet-Version başlığının kapatılması;**

**IIS sistemler için web.config dosyasına aşağıdaki kısım eklenmelidir.**

 <system.web>

 <httpRuntime enableVersionHeader="false" />

</system.web>

**Apache için httpd.conf dosyasında aşağıdaki değişiklikler yapılmalıdır.**

Header unset Server

ServerSignature Off

ServerTokens Prod

**Nginx için nginx.conf dosyasında aşağıdaki değişiklik yapılmalıdır.**

server\_tokens off;

**2- Server başlığının engellenmesi;**

**IIS Sistemlerinde;**

Kayıt defteri (Regedit) açılır

Aşağıdaki kayıt defteri anahtarında **DisableServerHeader** adlı bir DWORD girdisi oluşturun ve değerini **1** olarak ayarlayın.

**HKLM\SYSTEM\CurrentControlSet\Services\HTTP\Parameters**

****

**Diğer Bir Yol Olarakta;**

**URL Rewrite kuralına;**

<rewrite>

 <outboundRules rewriteBeforeCache="true">

 <rule name="Remove Server header">

 <match serverVariable="RESPONSE\_Server" pattern=".+" />

 <action type="Rewrite" value="" />

 </rule>

 </outboundRules>

</rewrite>

**3- X-Powered-By başlığının engellenmesi;**

**IIS Sistemlerinde;**

Düzeltme yapmak istediğiniz siteyi açın ve ardından **HTTP Response Headers** seçeneğini tıklayın.

**X-Powered-By** başlığına tıklayın ve ardından yanıttan çıkarmak için Eylemler Bölmesinde Kaldır'ı tıklayın.

**Diğer bir yol olarak;**

Web.config dosyasına aşağıdaki satırları ekleyebilirsiniz.

<httpProtocol>

 <customHeaders>

 <remove name="X-Powered-By" />

 </customHeaders>

</httpProtocol>

**HTTP Başlıklarını Global.asax Dosyasında kaldırma;**

IHTTP Modülüne aşağıdaki kodları ekleyebilirsiniz;

protected void Application\_PreSendRequestHeaders()

{

 // Response.Headers.Remove("Server");

 Response.Headers.Set("Server","My httpd server");

 Response.Headers.Remove("X-AspNet-Version");

 Response.Headers.Remove("X-AspNetMvc-Version");

}

**BeginRequest olayında kullanılması;**

protected void Application\_BeginRequest(object sender, EventArgs e)

{

 var application = sender as HttpApplication;

 if (application != null && application.Context != null)

 {

 application.Context.Response.Headers.Remove("Server");

 }

}

**X-AspNetMvc-Version başlığının Global.asax Dosyasında Kaldırmak için Application\_Start olayını oluşturun yada bulduğunuzda aşağıdaki satırları ekleyiniz.**

protected void Application\_Start()

{

 MvcHandler.DisableMvcResponseHeader = true;

}

**Zafiyet:** Kullanıcı Giriş Mekanizmasının Kaba Kuvvet (Brute Force) Ataklarına Açık Olması

**Açıklama:** Toplu form gönderme ve spam form gönderme saldırılarından korunmak için form girişlerinde CAPTCHA kontrolü eklenmektedir. Captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) basitçe kullanıcının karışık fakat okunabilir harf (veya harf rakam kombinasyonu) içeren resimde gösterilenleri girmesini gerektirir. Buradaki fikir bilgisayarın grafikte gizli olan kelimeyi bulamayacağı ve insanların kolaylıkla ayırt edebileceğidir. Captcha kullanımı, kullanıcının spesifik işlemler öncesi, örneğin iletişim formunu göndermeden önce resimdeki bilgiyi girmesi ile olur. Güçlü bir captcha kullanarak spam formların gönderilmesine ve servis engelleme ataklarına karşı önlem alınmış olur.

**Çözüm Önerisi:** Web sitelerinin kötüye kullanılmasını engellemek için Captcha’nın kullanılması tavsiye edilmektedir. Günümüzde üretilen pek çok Captcha uygulaması bulunmaktadır. Captcha kullanılırken şu özelliklere dikkat edilmesi gerekir:

• Captcha’nın Anlaşılabilir Olması: Captcha’nın kullanıcılar tarafından görülebilecek ve anlaşılabilecek bir şekilde oluşturulması gerekmektedir.

• Resimlerin Yeterince Bozulması: Günümüzde görüntü tanıma ve metin tanıma (image recognition, text recognition, OCR) teknikleri oldukça gelişmiştir. Bu teknikler kullanılarak yeterince bozulmaya uğratılmamış captcha’da yer alan yazılar tespit edilebilir. Bunu engellemek için captcha’nın yeterince bozulmasını sağlayacak algoritmanın tanımlanması gerekir.

• Resimlerin İnsan Dışı Varlıklar Tarafından Okunmaması: Web sayfasının kaynak kodunda captcha değerlerine işaret edecek bir parametrenin bulunmaması önerilmektedir. Örneğin 5 rakamına ait capcha resminin ismi 5.png olmamalı, öyle olsa bile bu değeri kaynak kodda robotların okuyup kullanabileceği biçimde gösterilmeyecek bir captcha metodu kullanılmalıdır.

**Ek Bilgi**: Genel Müdürlüğümüzde aktif olarak kullanılan NoCaptcha ve ReCaptcha Google kütüphaneleri Güvenli Captcha Uygulaması olarak kullanılması gereklidir. Ayrıca bilgiguvenli.saglik.gov.tr sitesindeki talimatlar kısmından Güvenlik Kodu (Captcha) Kullanım Talimatınında incelenmelidir.

<https://www.paperpencilwriteup.com/step-by-step-guide-to-integrate-google-recaptcha-2-0.html>

<https://developers.google.com/recaptcha/intro>

**Zafiyet:** Dizin Listeleme (Directory Listening)

**Açıklama:** index.php, default.asp ve benzeri index dosyalarının klasörlerde olmaması neticesinde web sunucuları tarafından dizin içeriklerinin listelenmesini sağlayan bir özelliktir. Örneğin dizin listeleme özelliğinin açık olduğu web uygulaması üzerindeki dizine yapılan bir istek neticesinde eğer ilgili dizinde herhangi bir index dosyası yoksa uygulamanın üzerinde koştuğu web sunucusu, dizin içeriğini listeleyecektir.

Dizin listeleme sonucunda hassas/kritik bilgilerin sızması mümkün olabilir. Örnek bir senaryo üzerinden ilerleyecek olursak: ilgili uygulamanın veritabanı bağlantı bilgilerinin tutulduğu config.php dosyası config klasöründe olsun ve bu dosyanın config.php.bak şeklinde yedeği de bu klasörde bulunuyor olsun. Saldırgan tarafından .config klasörünün varlığı crawling (link keşfetme) ya da fuzzing işlemi sonucu tespit edilir ve dizin listeleme ile config.php dosyasının yedeği ele geçirilirse, ilgili uygulamanın bağlantı bilgileri elde edilecek, bu bağlantı bilgileri ile web uygulamasına ya da veritabanı sunucusuna zarar verilebilecektir.

**Çözüm Önerisi:** Dizin listeleme özelliği açık olduğu halde dizinlerinizin listelenmesini istemiyorsanız, ilgili dizinlere içerisi boş veya sizin belirlediğiniz içeriğe sahip index dosyası yerleştirebilirsiniz. Ama test ya da geliştirme ortamından canlı ortamına geçiş yaparken ya da uygulamanıza yeni dizinler eklemeniz ve bu index dosyalarını oluşturmayı unutmanız ihtimalinde ise uygulamanızın güvenliğini riske atmış olacaksınız.

En kalıcı ve güvenli çözüm olarak, web sunucunuzu dizinlerin içeriğini listelemeyecek şekilde ayarlamak olacaktır.

**Detaylar için;**

<https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/> <https://www.netsparker.com.tr/blog/web-guvenligi/sunucu-uzerinde-dizin-listeleme-ozelligi-ve-getirdigi-riskler/>

<https://docs.microsoft.com/en-us/iis/configuration/system.webserver/directorybrowse>

**Ek Bilgi**:

**IIS sistemler için web.config dosyasına aşağıdaki kısım eklenmelidir.**

IIS web sunucusundaki dizin listeleme özelliği ön tanımlı olarak kapalı değerini almaktadır. Fakat bir regresyon ya da konfigürasyon değişikliği neticesinde dizin listeleme açılmış ise IIS web sunucusunun konfigürasyon arayüzünden dizin listelemeyi kapatmak mümkündür.

IIS 7.0 ve üzeri için dizin listelemeyi kapatmak için IIS Manager’ı açtıktan sonra Directory Browsing ayarlarına erişmeniz gerekir;



Ya da komut satırından **appcmd set config /section:directoryBrowse /enabled:false** komutunu çalıştırarak dizin listelemeyi devre dışı bırakabilirsiniz.

**Apache için httpd.conf dosyasında aşağıdaki değişiklikler yapılmalıdır.**

Apache web sunucusunda dizin listelemeyi kapatmak için ilgili uygulama üzerinde **.htaccess** dosyası oluşturulması ya da Apache web sunucusuna ait konfigürasyon detaylarının bulunduğu **httpd.conf** dosyasına aşağıdaki satırların eklenmesi ya da var olan satırların aşağıdaki gibi düzeltilmesi gerekmektedir;

<Directory /{YOUR DIRECTORY}>

Options FollowSymLinks

</Directory>

Kod örneğinden de anlaşılacağı üzere güvenli konfigürasyon yukarıdaki gibidir. Indexes ve MultiViews ifadelerinin kaldırılmasıyla dizin listeleme özelliği güvenli bir şekilde kapatılmış olacaktır.

**Nginx için nginx.conf dosyasında yapılması gerekenler;**

Nginx üzerinde dizin listeleme (directory listing) özelliği **ngx\_http\_index\_module** tarafından yönetilir. Dizin listeleme, nginx konfigurasyon dosyası üzerinde varsayılan olarak kapalı gelmektedir. Fakat bir regresyon ya da konfigürasyon değişikliği neticesinde dizin listeleme açılmış ise bunu kapatmak mümkündür.

Dizin listeleme özelliğinin açık olmasını veya kapalı olmasını sağlayan nginx parametre adı **autoindex** olarak isimlendirilmiştir ve location segmenti ile birlikte kullanılır.

Konfigurasyon dosyası varsayılan olarak **nginx.conf** adını almaktadır. Bu dosyanın varsayılan olarak bulunabileceği dizin yolları /usr/local/nginx/conf, /etc/nginx yada /usr/local/etc/nginx şeklindedir.

Eğer ön tanımlı olarak gelen değer değiştirildiyse aşağıdaki yapıya benzer bir ayar görebilirsiniz;

server {

 listen 80;

 server\_name domain.com www.domain.com;

 access\_log /var/...........................;

 root /path/to/root;

 location / {

 index index.php index.html index.htm;

 }

 location /somedir {

 autoindex on;

 }

}

Bu kısımda belirleyici parametre yukarıda bahsettiğimiz gibi “autoindex on;” özelliğidir. Bu örnekte sadece “somedir” dizini altında listeleme yapılmaktadır. Eğer herhangi bir dizin adı yazılmazsa (Örneğin location / {autoindex on;}) tüm klasörlere etki eder. Dizin listelemeyi kapatmak için de autoindex parametre değerini off haline getirmeliyiz.

Son olarak değişikliklerin geçerli olması için aşağıdaki komutu çalıştırmamız gerekiyor;

service nginx restart