**Bilgi Sistemleri ve Güvenliği**

**Bekir Öndeş 175541008**

# [Possible] BREACH Attack Detected (BREACH Saldırısı Tespit Edildi)



MEDIUM

## Özet

Invicti, bu web sitesinde BREACH (Browser Reconnaissance & Exfiltrasyon yoluyla Adaptive Compression of

Hypertext) saldırısının mümkün olduğunu tespit etti.

BREACH saldırısını mümkün kılan unsurlar nedeniyle, SSL/TLS korumalı trafik savunmasız kalır ve web sitesindeki bilgileri ortaya çıkarmak için saldırıya uğrayabilir.

Hangi SSL/TLS sürümünü kullanırsanız kullanın, saldırılar yine de mümkündür. Saldırılar, TLS katmanı sıkıştırması gerektirmez ve herhangi bir şifre paketine karşı çalışabilirler.

## Etki

SSL/TLS korumalı bir bağlantı kullansanız bile, saldırgan kurbanın şifreli trafiğini görüntüleyebilir ve kurbanın (görünmez çerçeveler kullanarak) savunmasız web sunucusuna HTTP istekleri göndermesine neden olabilir. Bu adımları izleyerek bir saldırgan web sitesinden bilgi çalabilir ve aşağıdakileri yapabilir:

• Ortaya çıkardıkları kısmi düz metni bir kurbanın isteklerine enjekte edin

• Şifrelenmiş trafiğin boyutunu ölçün.

## Düzeltme

Netsparker, hedef web sayfası bunu kolaylaştıran aşağıdaki koşulları karşıladığı için Olası bir İhlal Saldırısı sorunu bildirdi:

• HTTP düzeyinde sıkıştırma kullanan bir sunucudan sunulur (ör. gzip)

• HTTP yanıt gövdelerinde kullanıcı girdisini yansıtır

• HTTP yanıt gövdelerinde hassas bilgiler (CSRF belirteci gibi) içerir

Sorunu azaltmak için aşağıdaki çözümleri öneriyoruz:

1. Mümkünse, HTTP düzeyinde sıkıştırmayı devre dışı bırakın

2. Hassas bilgileri kullanıcı girişinden ayırın

3. Güvenlik açığı bulunan sayfaları CSRF belirteci ile koruyun. SameSite Cookie özniteliği bu sorunu azaltacaktır, çünkü bu sorundan yararlanmak için bir saldırgan, kurbanı görünmez çerçeveler kullanarak hedef bir web sitesini ziyaret etmeye zorlar. SameSite cookie özelliği eklendiğinde, hedefe ait cookie'ler üst düzey navigasyon içermeyen bir istek ile gönderilmeyecektir.

4. Yanıtlara rastgele sayıda bayt ekleyerek trafiğin uzunluğunu gizleyin.

5. Dakikada beş kez maksimum sayfaya ulaşılması için bir hız sınırı ekleyin.

# HTTP Strict Transport Security (HSTS) PolicyNot Enabled(HTTP Sıkı Aktarım Güvenliği (HSTS) Politikası Etkin Değil)



MEDIUM

## Özet

Invicti, HTTP Katı Aktarım Güvenliği (HSTS) ilkesinin etkinleştirilmediğini belirledi.

Hedef web sitesi yalnızca HTTPS'den değil, aynı zamanda HTTP'den de sunuluyor ve HSTS politika uygulamasından yoksun.

HTTP Strict Transport Security (HSTS), bir web sunucusunun, uygun kullanıcı aracılarının (bir web tarayıcısı gibi) kendisiyle yalnızca güvenli (HTTPS) bağlantılar kullanarak etkileşime gireceğini beyan ettiği bir web güvenlik politikası mekanizmasıdır. HSTS Politikası, sunucu tarafından kullanıcı aracısına "Strict-Transport-Security" adlı bir HTTP yanıt başlığı alanı aracılığıyla iletilir. HSTS Politikası, kullanıcı aracısının sunucuya yalnızca güvenli bir şekilde erişeceği bir süre belirtir.

Bir web uygulaması kullanıcı aracılarına HSTS Politikası yayınladığında, uyumlu kullanıcı aracıları aşağıdaki gibi davranır:

• Web uygulamasına atıfta bulunan güvenli olmayan (HTTP) bağlantıları otomatik olarak güvenli (HTTPS) bağlantılara dönüştürün. (Örneğin, http://example.com/some/page/, sunucuya erişmeden önce https://example.com/some/page/ olarak değiştirilecektir.)

• Bağlantının güvenliği sağlanamıyorsa (örneğin sunucunun TLS sertifikası kendinden imzalıysa), kullanıcı aracıları bir hata mesajı gösterir ve kullanıcının web uygulamasına erişmesine izin vermez.

## Düzeltme

Web sunucunuzu HTTP isteklerini HTTPS'ye yönlendirecek şekilde yapılandırın.Yani Apache için httpd.conf dosyasında değişiklik yapmalısınız..

**Uğur Umur Zelcek 170541058**

# Out-of-date Version (IIS), Güncel Olmayan Sürüm



MEDIUM

## Özet

Invicti, hedef web sitesinin IIS kullandığını belirledi ve güncel olmadığını tespit etti.

## Etki

Bu, yazılımın eski bir sürümü olduğundan saldırılara açık olabilir.

## Düzeltme

IIS'yi daha yüksek bir sürüme yükseltmek, bağımsız bir işlem değildir. IIS sürümü, büyük ölçüde sunucu makinenizde kullandığınız Windows işletim sistemi sürümüne bağlıdır.

Bu tür bir nedenle IIS'yi daha yüksek bir sürüme yükseltmek mümkün değilse, satıcı tarafından yayınlanan yamaları izlemenizi ve uygulamanızı şiddetle tavsiye ederiz.

Lütfen tüm IIS güncellemelerinin ve yamalarının Windows Güncellemeleri olarak geldiğini unutmayın. Ayrıca, hangi güncelleme paketinin/paketlerinin uygulanacağını da seçebilirsiniz.

# Out-of-date Version ( jQuery) Güncel Olmayan Sürüm



MEDIUM

## Özet

Invicti, hedef web sitesinin jQuery kullandığını belirledi ve güncel olmadığını tespit etti.

## Etki

Bu, yazılımın eski bir sürümü olduğundan saldırılara açık olabilir.

## Düzeltme

Lütfen, jQuery kurulumunuzu en son kararlı sürüme yükseltiniz.

**Mertcan Taşkıran - 195541303**

# Active Mixed Content over HTTPS (HTTPS Üzerinden Etkin Karmaşık İçerik)



MEDIUM

## Özet

Invicti, bir HTTPS sayfasında HTTP üzerinden etkin bir içeriğin yüklendiğini tespit etti.

## Etki

Aktif İçerik, sayfanızın bağlamında çalışabilen ve ayrıca tüm sayfayı değiştirebilen bir kaynaktır. HTTPS sayfası, normal, açık metin HTTP aracılığıyla alınan komut dosyaları veya stil sayfaları gibi etkin içerik içeriyorsa, bağlantı yalnızca kısmen şifrelenir. Şifrelenmemiş içeriğe koklayıcılar erişebilir. Ortadaki adam saldırganı, HTTP içeriği talebini engelleyebilir ve ayrıca kötü niyetli kodlar eklemek için yanıtı yeniden yazabilir. Kötü amaçlı etkin içerik, kullanıcının kimlik bilgilerini çalabilir, kullanıcı hakkında hassas veriler elde edebilir veya kullanıcının sistemine kötü amaçlı yazılım yüklemeye çalışabilir (örneğin, tarayıcıdaki veya eklentilerindeki güvenlik açıklarından yararlanarak) ve bu nedenle bağlantı artık korunmaz.

## Düzeltme

Karışık içerik sorunlarına karşı savunmak için iki teknoloji vardır:

 HTTP Strict Transport Security (HSTS), kullanıcı hataları (80 numaralı bağlantı noktasından web sitenize erişme girişimi) ve uygulama hataları (geliştiricileriniz güvenli bir sayfaya güvenli olmayan bir bağlantı yerleştirir) karşısında bile güvenli kaynak alımını zorlayan bir mekanizmadır.

 İçerik Güvenliği Politikası (CSP), üçüncü taraf web sitelerinden güvenli olmayan kaynak alımını engellemek için kullanılabilir.

 Son olarak, kullanıcının hangi protokole bağlı olduğuna bağlı olarak, kullanıcının tarayıcısının HTTP veya HTTPS'yi uygun şekilde otomatik olarak seçmesini sağlamak için "protokol göreli URL'leri" kullanabilirsiniz. Örneğin:

Bir stil yüklemek için protokole göre bir URL şöyle görünür:

<link rel="stylesheet" href="//example.com/style.css"/>

Komut dosyaları için:

<script type="text/javascript" src="//example.com/code.js"></script>

Tarayıcı, hangisi uygunsa, URL'nin başına otomatik olarak;

"http:" veya "https:" ekler.

**Mustafa Ali Kağan Küçük 170542018**

# Weak Ciphers Enabled (Zayıf Şifreler Etkin)



MEDIUM

## Özet

Invicti, güvenli iletişim (SSL) sırasında zayıf şifrelerin etkinleştirildiğini tespit etti.

Ziyaretçilerinizle güvenli iletişimi korumak için web sunucunuzda yalnızca güçlü şifrelere izin vermelisiniz.

## Etki

Saldırganlar, sunucunuz ve ziyaretçileriniz arasındaki SSL trafiğinin şifresini çözebilir.

## Yapılacak İşlemler

1. Apache için, httpd.conf içindeki SSLCipherSuite yönergesini değiştirmelisiniz.

SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4

2. Lighthttpd yazılımına bu kısımları eklemeliyiz:

ssl.honor-cipher-order = "enable" ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"

3. Microsoft IIS için sistem kayıt defterinde bazı değişiklikler yapmalısınız. Kayıt defterini yanlış düzenlemek, sisteminize ciddi şekilde zarar verebilir. Kayıt defterinde değişiklik yapmadan önce, bilgisayarınızdaki tüm değerli verileri yedeklemelisiniz.

a. Başlat'a tıklayın, Çalıştır'a tıklayın, yazın regedt32veya yazın regeditve ardından Tamam'a tıklayın.

b. Kayıt Defteri Düzenleyicisi'nde aşağıdaki kayıt defteri anahtarını bulun: HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders

c. Aşağıdaki kayıt defteri anahtarları için "Etkin" DWORD'yi "0x0" olarak ayarlayın:

YÜKSEK - Triple-DES kullanan tüm şifreler etkinleştirilir ORTA - 128 bit şifrelemeli tüm şifreler etkinleştirilir.

SCHANNEL\Ciphers\DES 56/56 SCHANNEL\Ciphers\RC4 64/128 SCHANNEL\Ciphers\RC4 40/128 SCHANNEL\Ciphers\RC2 56/128 SCHANNEL\Ciphers\RC2 40/128 SCHANNEL\Ciphers\NULL SCHANNEL\Hashes\MD5

!aNULL - Boş kimlik doğrulama devre dışı bırakılır ve daha sonra bir "+" (artı) işareti aracılığıyla dizeye yeniden eklenemez

!MD5 - MD5 karma devre dışı bırakılır ve daha sonra "+" (artı) işaretiyle dizeye yeniden eklenemez.

$ openssl ciphers 'HIGH' ADH-AES256-SHA:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:AES256- SHA:ADH-AES128-SHA:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:

AES128-SHA:ADH-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DES-CBC3- SHA:DES-CBC3-MD5

$ openssl ciphers 'MEDIUM' ADH-SEED-SHA:DHE-RSA-SEED-SHA:DHE-DSS-SEED-SHA:SEED-SHA:ADH- RC4-MD5:RC4-SHA:RC4-MD5:RC2-CBC-MD5:RC4-MD5

$ openssl ciphers 'LOW' ADH-DES-CBC-SHA:EDH-RSA-DES-CBC-SHA:EDH-DSS-DES-CBC-SHA:DES-

CBC-SHA:DES-CBC-MD5

tüm SSLCipherSuite dizesini 'openssl şifreleri' ile de belirtebilirsiniz.

"güçlü" tanımı, istediğiniz kullanım durumlarına, tehdit modellerinize ve kabul edilebilir risk seviyelerinize bağlıdır. Apache HTTP Sunucusu ekibi bunları sizin için belirleyemez. Güçlü şifrelemenin tek başına

güçlü güvenlik sağlamadığını lütfen unutmayın.

Lighttpd(lighty), hızlı, güvenli, esnek ve standartlara sadık olacak şekilde tasarlanmış bir web sunucusudur. Hızın çok önemli olduğu ortamlar için optimize edilmiştir. Bunun nedeni, diğer sunuculardan daha az CPU ve RAM tüketmesidir.

Lighttpd, yük sorunları olan herhangi bir sunucu için uygundur. Ücretsiz bir yazılımdır ve BSD lisansı altında dağıtılmaktadır. Resmi olarak GNU / Linux ve UNIX üzerinde çalışır.

Microsoft Windows için Kevin Worthington tarafından sağlanan Lighttpd For Windows olarak bilinen bir dağıtım vardır.

## Düzeltme

Web sunucunuzu zayıf şifrelerin kullanılmasına izin vermeyecek şekilde yapılandırın.

# ViewState is not Encrypted ViewState Şifreli değil



**LOW**

## Özet

Invicti, ViewState Şifrelemenin devre dışı bırakıldığını tespit etti. ASP.NET görünüm durumu, geri göndermeler arasında bir Web Formunun durumundaki değişiklikleri sürdürmek için bir ASP.NET Web sayfası tarafından kullanılan tekniktir. Varsayılan olarak, görünüm durumu verileri sayfada gizli bir alanda depolanır ve base64 kodlaması kullanılarak kodlanır.

Görüntüleme durumu verileri şifrelenmez, bu nedenle hala görüntülenebilir ve potansiyel olarak kötü niyetli kullanıcılar tarafından ele geçirilip okunabilir. Ancak bazı durumlarda kontroller, hiçbir kullanıcının erişmemesi gereken bilgileri depolamak için görünüm durumunu kullanabilir. Örneğin, sayfa, öğe tanımlayıcılarını (veri anahtarlarını) görünüm durumunda saklayan bir veriye bağlı denetim içerebilir.

## Etki

Saldırgan, olası güvenlik açıkları için uygulamanın durum yönetimi mantığını inceleyebilir; uygulamanız, uygulama açısından kritik bilgileri ViewState'te saklarsa, bu bilgiler de ortaya çıkar.

## Düzeltme

ASP.NET, ViewState parametreleri için şifreleme sağlar.

Sayfa tabanlı koruma için, aşağıdaki yönergeyi etkilenen sayfanın en üstüne yerleştirin.

<%@Page ViewStateEncryptionMode="Always" %>

web.configBu seçeneği, dosyaları kullanarak tüm uygulama için de ayarlayabilirsiniz

. web.configUygulamanızın dosyası için aşağıdaki yapılandırmayı uygulayın.

<System.Web>

<pages viewStateEncryptionMode="Always">

</System.Web>

**Muhammed Talha BAYSAL - 170541052**

# Phishing by Navigating Browser Tabs (Tarayıcı Sekmelerinde Gezinerek Oltalama) (Tabnabbing)



**LOW**

## Özet

Netsparker, olası bir tarayıcı sekmelerinde gezinerek oltalama saldırısı belirledi.

Target="\_blank" etiketli normal href'lere sahip açık pencereler, window.opener.location'ı değiştirebilir ve ana web sayfasını farklı bir kaynakta bile olsa başka bir şeyle değiştirebilir.

## Tanım

Tabnabbing, 2010'dan bu yana denenmiş ve gerçek kimlik avı biçimlerinden biri olmuştur. Saldırgan, birçok insanın meşgul olduğu ve tarayıcılarında açtıkları sekmelere dikkat etmediği gerçeğinden yararlanır. Siber suçlunun güvenilir web sitesi görünümlerini taklit etme yeteneği, kullanıcının kimlik bilgilerini girebilmesi için bir web sayfası gibi görünmelerine olanak tanır. Tabnabbing adı, bir güvenlik araştırmacısı ve tasarım uzmanı Aza Raskin tarafından oluşturulmuştur.

## Etki

Bu güvenlik açığı, komut dosyası yürütülmesine izin vermese de, üst sekmenin sessizce yerini alan oltalama saldırılarına izin verir. Eğer bağlantılar rel=”noopener noreferrer” özelliğinden yoksunsa, üçüncü taraf bir site window.opener.location.assign kullanarak kaynak sekmenin URL’sini değiştirebilir. Bu sayede kullanıcıları hala güvenli bir sitede oldukları yönünde kandırabilir ve hassas verilerini kötü amaçlı websitesine girmelerine sebep olabilir.

## Düzeltme

•Sayfaların window.opener'ı kötüye kullanmasını önlemek için bağlantılara rel=noopener ekleyin. Bu, sayfanın Chrome ve Opera tarayıcılarında window.opener özelliğine erişememesini sağlar.  
•Daha eski tarayıcılar ve Firefox'ta, Referer başlığını ek olarak devre dışı bırakan rel=noreferrer ekleyebilirsiniz.

**Tahir Bayraktar - 170541061**

# [Possible] Cross-site Request Forgery (Siteler Arası İstek Sahteciliği)



**LOW**

## Özet

Suçlu, olası bir Siteler Arası İstek Sahteciliği tespit etti. CSRF çok yaygın bir güvenlik açığıdır. Kullanıcıyı, o anda kimliğinin doğrulandığı bir web uygulamasında istenmeyen eylemler yürütmeye zorlayan bir saldırıdır.

## Tanım

CSRF (Cross Site Request Forgery) genel yapı olarak bir web sitesinin açığından faydalanarak site kullanıcılarının istekleri dışında sanki o kullanıcıymış gibi erişerek işlem yapılması sürecini içerir. Genellikle GET requestleri ve SESSION işlemlerinin doğru kontrol edilememesi durumlarındaki açıklardan saldırganların faydalanmasını sağlamaktadır.

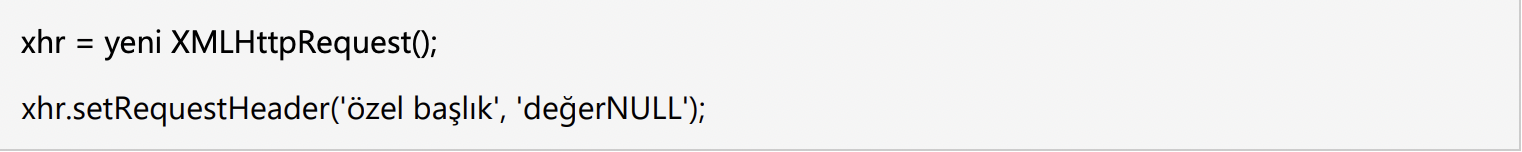
## Etki

Saldırgan, uygulamaya bağlı olarak, kullanıcı ekleme, içerik değiştirme, veri silme gibi kullanıcı tarafından yapılabilecek eylemlerden herhangi birini yükleyebilir. Kurbanın kullanabileceği tüm işlevler saldırgan tarafından kullanılabilir. Bu kuralın tek istisnası, yalnızca meşru kullanıcının bilebileceği (kullanıcının şifresi gibi) ek bilgiler gerektiren bir sayfadır.

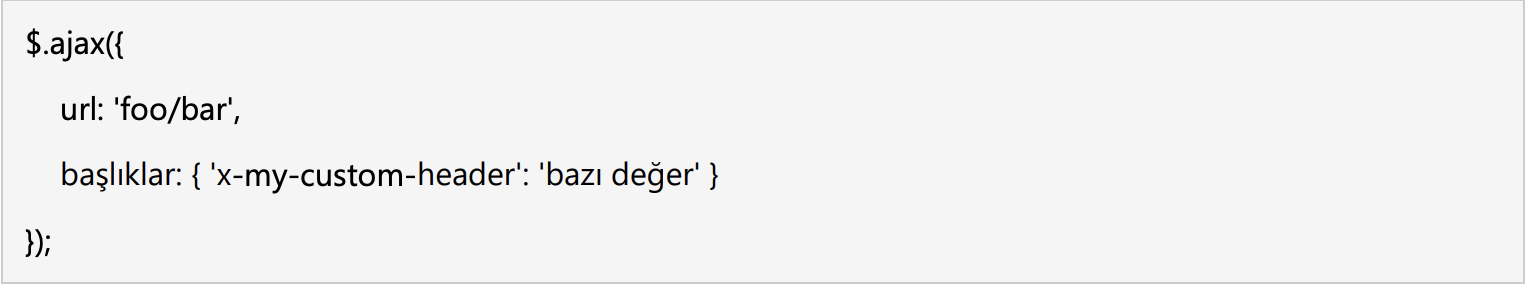
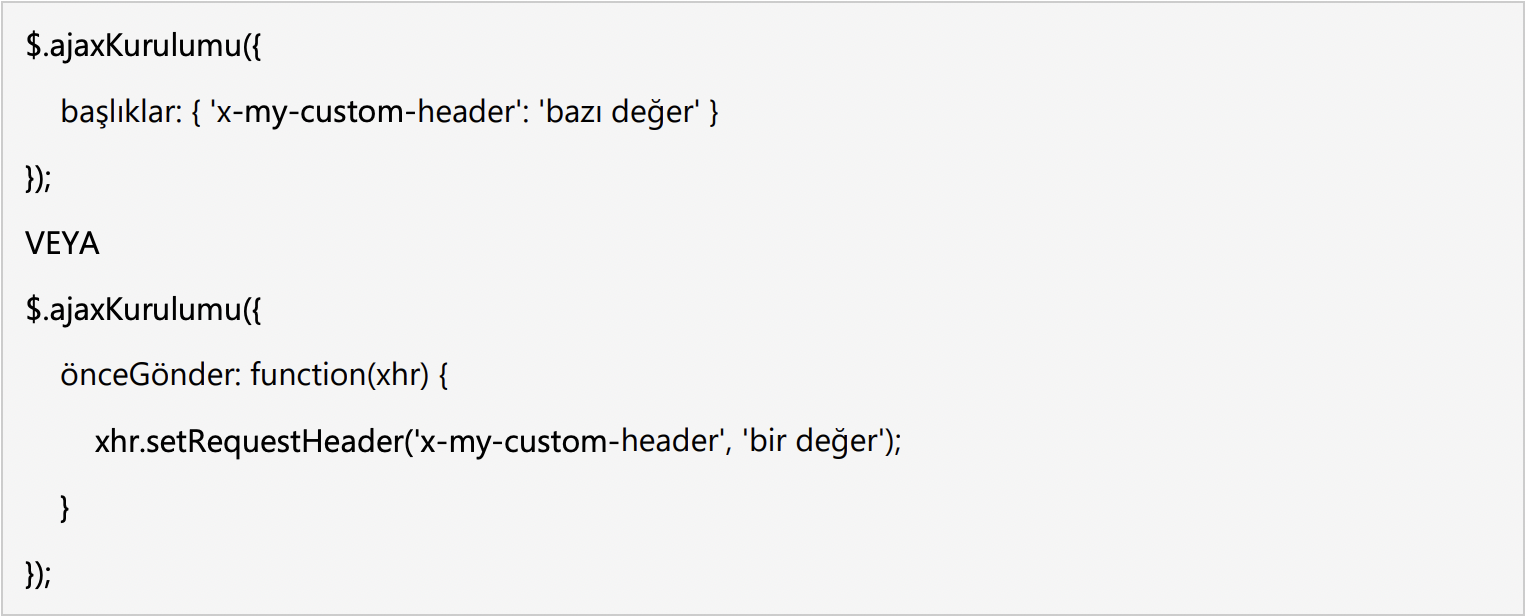
## Düzeltme

İsteğin yetkili bir kaynaktan gelip gelmediğini belirlemek için kullanılabilecek her HTTP isteğinde ek bilgiler gönderin. Bu "doğrulama belirteci", kullanıcının hesabına halihazırda erişimi olmayan saldırgan için tahmin edilmesi zor olmalıdır. Bir istekte doğrulama belirteci eksikse veya belirteç beklenen değerle eşleşmiyorsa, sunucu isteği reddetmelidir

Ajax isteğinde form gönderiyorsanız, tarayıcı sitelerin başka bir siteye özel HTTP üstbilgileri göndermesini engellediği, ancak sitelerin XMLHttpRequest kullanarak kendilerine özel HTTP üstbilgileri göndermesine izin verdiği için CSRF'yi önlemek için özel HTTP üstbilgileri kullanılabilir. o JavaScript'teki yerel XMLHttpRequest (XHR) nesnesi için;



JQuery için, özel bir başlık (veya başlık kümesi) eklemek istiyorsanız,

1. Bireysel İstek  
   
2. Her İstek  
   

**Hasan ÇAKMAKÇI - 170542019**

# Stack Trace Disclosure-Yığın İzleme Açıklaması (ASP.NET)



**LOW**

## Özet

Invicti, hedef web sunucusunun HTTP yanıtında bir yığın izleme ifşası (ASP.NET) belirledi.

## Etki

Bir saldırgan aşağıdakiler gibi bilgileri alabilir:

• ASP.NET sürümü.

• Geçici ASP.NET dosyalarının fiziksel dosya yolu.

• Oluşturulan istisna ve muhtemelen kaynak kodu, SQL sorguları vb. hakkında bilgiler.

Bu bilgi, bir saldırganın daha fazla bilgi edinmesine ve potansiyel olarak hedef sistem için daha fazla saldırı geliştirmeye odaklanmasına yardımcı olabilir.

## Düzeltme

web.config Özel hata sayfaları uygulayarak bilgi sızıntısını önlemek için dosyanızda aşağıdaki değişiklikleri uygulayın.

**<sistem.web>**

**<customerrors mode="açık" defaultredirect="~/error/generalerror.aspx">**

**<error statuscode="403" yönlendirme="~/error/forbidden.aspx" />**

**<error statuscode="404" yönlendirme="~/error/pagenotfound.aspx" />**

**<error statuscode="500" yönlendirme="~/error/internalerror.aspx" />**

**</customerrors>**

**</sistem.web>**

**Seydi Sarıtaş - 190541087**

# Missing X-Frame-Options Header (Eksik X-Frame-Options Başlığı)



**LOW**

## Nedir

Headerlar, HTTP'nin hem istek hem de yanıt mesajlarında yer alan ve mesajın meta bilgilerini taşıyan bölümlerdir.

Bu meta bilgileri, istekler için, talep edilen mesajın dili, tipi, browserın sakladığı yetki bilgileri, önbellekte tutulan içeriğe ait bilgiler olabileceği gibi; yanıt yani response mesajlarında ise, içeriğin boyutu ve tipi, bellekte saklanma tercihleri, sunucu bilgileri, tarih ve saati, istemci tarafında set edilecek credentials gibi, bir takım bilgilerdir.

Eksik X-Frame-Options başlığı, bu web sitesinin bir clickjacking saldırısı riski altında olabileceği anlamına gelir. X-Frame-Options HTTP başlık alanı, tarayıcının iletilen kaynağı bir çerçeve içinde mi yoksa bir iframe içinde mi oluşturması gerektiğini belirten bir ilkeyi belirtir.

## Clickjacking Nedir

UI Redress Attack olarak da bilinen Clickjacking saldırısı sızma testlerinde sık karşılaşılan bir zafiyettir. Siber saldırgan, güvenli olarak görünen web sayfası kullanıcılarını güvensiz bir sayfaya yönlendirerek manipüle ettiği bu kullanıcıların işlem yapmasını hedef alır.

## Sitede hangi istek’te header’in kısmının eksik olduğu keşfedildi?



## Düzeltme

1. Tarayıcıya diğer alanlardan gelen çerçevelemeye izin vermemesi talimatını veren HTTP yanıt başlıklarında uygun X-Frame-Options'ın gönderilmesi.
2. ALLOW-FROM URLIBelirli bir URL'ye kendisini bir iframe içinde yüklemesi için izin verir. Ancak lütfen buna dikkat edin, tüm tarayıcılar bunu desteklemez.
3. Geçerli çerçevenin en üst düzey pencere olmasını sağlamak için kullanıcı arayüzünde savunma kodu kullanmak.

**Anas Aljerf - 195541606**

# Cookie Not Marked as Secure (Çerez Güvenli Olarak İşaretlenmedi)



**LOW**

## Nedir

bir kimlik doğrulama çerezi düşünelim. Saldırgan bu çerezi yakalayabildiğinde, kullanıcının kimliğine bürünebilir.

HTTP protokolü kullanıldığında, trafik düz metin olarak gönderilir. Saldırganın trafiği görmesini/değiştirmesini sağlar (man-in-the-middle attack). HTTPS, HTTP'nin güvenli bir sürümüdür — uygulama katmanının verilerini korumak için SSL/TLS kullanır.

HTTPS kullanıldığında şu özellikler elde edilir: kimlik doğrulama, veri bütünlüğü ve gizlilik.

Bir kimlik doğrulama çerezi durumunu ele alalım. Bu çerezi çalmak, kullanıcının kimliğine bürünmekle eşdeğerdir. HTTP kullanıldığında, çerez düz metin olarak gönderilir. Bu, tarayıcı ve sunucu arasındaki iletişim kanalını gizlice dinleyen saldırgan için iyidir - çerezi alıp kullanıcının kimliğine bürünebilir.

Şimdi HTTP yerine HTTPS kullanıldığını varsayalım. HTTPS gizlilik sağlar. Bu nedenle saldırgan çerezi göremez. Sonuç, kimlik doğrulama çerezini gizlice dinlenmemesi için güvenli bir kanal üzerinden göndermektir.

Şu anda ortaya çıkabilecek soru şudur: HTTPS kullanabiliyorsak neden güvenli bir bayrağa ihtiyacımız var?

Bu soruyu cevaplamak için aşağıdaki senaryoyu ele alalım. Site, HTTP ve HTTPS üzerinden kullanılabilir. Ayrıca tarayıcı ile sunucu arasındaki iletişim kanalının ortasında bir saldırganın olduğunu varsayalım. HTTPS üzerinden gönderilen çerez dinlenemez.

Ancak saldırgan, sitenin HTTP üzerinden de erişilebilir olmasından faydalanabilir. Saldırgan, sitenin HTTP sürümünün bağlantısını kullanıcıya gönderebilir. Kullanıcı bağlantıyı tıklar ve HTTP isteği oluşturulur. HTTP trafiği düz metin olarak gönderildiğinden, saldırgan iletişim kanalını gizlice dinler ve kullanıcının kimlik doğrulama çerezini okur. Bu çerezin yalnızca HTTPS üzerinden gönderilmesine izin verebilir miyiz? Bu mümkün olsaydı, saldırganın hikayemizdeki kimlik doğrulama çerezini okumasını engellerdik. Bunun mümkün olduğu ve tam olarak bu amaç için güvenli bir bayrak kullanıldığı ortaya çıktı - güvenli bir bayrağa sahip çerez yalnızca bir HTTPS bağlantısı üzerinden gönderilecek.

Biraz önce tarayıcı ile sunucu arasındaki iletişim kanalını gizlice dinleyen bir saldırgandan çerezin nasıl korunacağı anlatılmıştı. Ancak, gizlice dinleme, çerezi kapmak için tek saldırı vektörü değildir.

Hikayeye kimlik doğrulama çerezi ile devam edelim ve uygulamada XSS (siteler arası komut dosyası çalıştırma) güvenlik açığı olduğunu varsayalım. Ardından saldırgan, kimlik doğrulama tanımlama bilgisini çalmak için XSS güvenlik açığından yararlanabilir. Bunun olmasını bir şekilde önleyebilir miyiz?

Bu sorunu çözmek için bir HttpOnly bayrağının kullanılabileceği ortaya çıktı. HttpOnly bayrağı kullanıldığında, XSS istismarı durumunda JavaScript bu kimlik doğrulama tanımlama bilgisini okuyamaz. Hedefe ulaştık gibi.

## Özet

Invicti, güvenli olarak işaretlenmeyen ve HTTPS üzerinden aktarılan bir tanımlama bilgisi belirledi. Bu, çerezin trafiği başarıyla kesip şifresini çözebilen bir saldırgan tarafından veya başarılı bir ortadaki adam saldırısını takiben potansiyel olarak çalınabileceği anlamına gelir.

## Başarılı Sömürü için Gerekli Beceriler

Bu sorundan yararlanmak için saldırganın trafiği kesebilmesi gerekir. Bu genellikle web sunucusuna veya kurbanın ağına yerel erişim gerektirir. Saldırganların 2. katmanı anlaması, trafik için geçiş noktaları olarak sistemlere fiziksel erişime sahip olması veya kurban ile web sunucusu arasındaki bir sisteme yerel olarak erişim kazanmış olması gerekir.

# Robot.txt



**INFORMATION**

## Giriş

Arama motorları kendi botları aracılığıyla web sitelerini düzenli olarak tarar ve tarama sonuçlarını arşivler. Ancak bazı durumlarda web sitenizin her kısmını botlar için görünür kılmak istemiyor olabilirsiniz. Botların görmesinin önüne geçmek istediğiniz bölümler içinse en iyi yöntem ise robots.txt dosyasını kullanmak olacaktır.

## Robots.txt Nedir?

Web sitenizdeki hangi dosya ya da sekmelerin tarayıcı botlarına açık olduğunu seçmenize imkan tanıyan robots.txt dosyası, sitenizin kök dizininde yer alır. 'Robot hariç tutma standartları' çerçevesinde oluşturulan bu dosya, aslında düz bir metinden oluşur. Ancak dosya içeriğinde bir veya daha fazla komut bulunur. Her komutla birlikte farklı bir tarayıcının, ayrı bir dosyaya erişmesi engellenir.

Robots.txt içerisinde kullanılan allow ve disallow direktifleri site içerisindeki hangi sayfaların taramaya açılıp kapanacağının belirtildiği protokollerdir.

Allow’un türkçe karşılığı izin vermek ve Disallow’un karşılığı izin vermemektir.

Bu bilgiden hareketle Allow: protokol kulllanımı ile site içerisindeki herhangi bir dizini, URL yolunu tümüyle yada kısmen taramalara açabilirsiniz.

Aynı şekilde Disallow: kullanımı ile herhangi bir dizini, URL yolunu tümüyle yada kısmi olarak taramalara kapatabilirsiniz.

## Peki riski nedir?

Robots.txt dosyasının varlığı kendi başına herhangi bir güvenlik açığı oluşturmaz. Ancak, genellikle bir sitenin içeriğinin kısıtlı veya özel alanlarını belirlemek için kullanılır. Bu nedenle, dosyadaki bilgiler, özellikle tanımlanan konumlardan bazıları sitenin başka bir yerinden bağlantılı değilse, bir saldırganın sitenin içeriğinin haritasını çıkarmasına yardımcı olabilir. Uygulama, bu alanlara erişimi korumak için robots.txt'ye güveniyorsa ve bunlar üzerinde uygun erişim denetimi uygulamıyorsa, bu ciddi bir güvenlik açığı oluşturur.

## Kaynaklar

https://silesiasecuritylab.com/free-resources/

Author: Dawid Czagan

Posted: August/10/2020

https://www.invicti.com/web-vulnerability-scanner/vulnerabilities/cookie-not-marked-as-secure/

https://boosmart.com/robots-txt-nedir-nasil-kullanilir/#Robotstxt\_Dosyasi\_Nerede\_Barindirilir\_Bulunur

**Aslan Koyuncu - 210541101**

# Version Disclosure(ASP.NET)



**LOW**

## Özet

Invicti, hedef web sunucusunun HTTP yanıtında bir sürüm ifşası (ASP.NET) belirledi.

Bu bilgiler, bir saldırganın kullanılan sistemleri daha iyi anlamasına ve potansiyel olarak ASP.NET'in belirli sürümünü hedefleyen başka saldırılar geliştirmesine yardımcı olabilir.

## Etki

Saldırgan, açıklanan bilgileri, tanımlanan sürüm için belirli güvenlik açıklarını toplamak için kullanabilir.

## Düzeltme

Özel hata sayfaları kullanarak ve HTTP yanıtlarından web.configkaldırarak bilgi sızıntısını önlemek için aşağıdaki değişiklikleri dosyanıza uygulayın .

X-AspNet-Version

<Sistem.Web>

<httpRuntime enableVersionHeader="false" />

<customErrors mode="Açık" defaultRedirect="~/error/GeneralError.aspx">

<error statusCode="403" yönlendirme="~/error/Forbidden.aspx" />

<error statusCode="404" yönlendirme="~/error/PageNotFound.aspx" />

<error statusCode="500" yönlendirme="~/error/InternalError.aspx" />

</customErrors>

</Sistem.Web>

# Insecure Transportation Security Protocol Supported (TLS 1.0)



**LOW**

## Özet

Invicti, güvenli olmayan ulaşım güvenlik protokolünün (TLS 1.0) web sunucunuz tarafından desteklendiğini tespit etti.

TLS 1.0'ın birkaç kusuru vardır. Saldırgan bağlantı hatalarına neden olabilir ve BEAST (SSL/TLS'ye Karşı Tarayıcı İstismarı) gibi güvenlik açıklarından yararlanmak için TLS 1.0 kullanımını tetikleyebilir.

TLS 1.0 kullanan web siteleri, 30 Haziran 2018'den beri PCI tarafından uyumlu değildir.

## Etki

Saldırganlar, ortadaki adam saldırıları gerçekleştirebilir ve web siteniz ile ziyaretçileri arasındaki şifreleme trafiğini gözlemleyebilir.

## Yapılacak İşlemler

TLS 1.0'ı devre dışı bırakmanızı ve bunu TLS 1.2 veya üstü ile değiştirmenizi öneririz. Daha fazla ayrıntı için Çözüm bölümüne bakın.

## Düzeltme

Web sunucunuzu zayıf şifrelerin kullanılmasına izin vermeyecek şekilde yapılandırın. Değişiklikleri etkinleştirmek için web sunucusunu yeniden başlatmanız gerekir.

Apache için mod\_ssl modülü tarafından sağlanan SSLProtocol yönergesini ayarlayın. Bu yönerge, sunucu düzeyinde veya sanal ana bilgisayar yapılandırmasında ayarlanabilir.

SSLProtokol +TLSv1.2

nginx.confNginx için, dosyada ssl\_protocols yönergesinin herhangi bir kullanımını bulun ve TLSv1.

ssl\_protocols TLSv1.2;

Microsoft IIS için sistem kayıt defterinde bazı değişiklikler yapmalısınız. Kayıt defterini yanlış düzenlemek, sisteminize ciddi şekilde zarar verebilir. Kayıt defterinde değişiklik yapmadan önce, bilgisayarınızdaki tüm değerli verileri yedeklemelisiniz.

Başlat'a ve ardından Çalıştır'a tıklayın, regedt32 veya yazın regeditve ardından Tamam'a tıklayın.

Kayıt Defteri Düzenleyicisi'nde aşağıdaki kayıt defteri anahtarını bulun veya yoksa oluşturun:

HKEY\_LOCAL\_MACHINESYSTEMCurrentControlSetControlSecurityProvidersSCHANNELProtokollerTLS 1.0

Adlandırılmış bir anahtar bulun Serverveya yoksa oluşturun.

Anahtarın altında, mevcut değilse veya oluştur Serveradlı bir DWORD değeri bulun ve değerini "0" olarak ayarlayın.Enabled

Lighttpd için yapılandırma dosyanıza aşağıdaki satırları koyun:

ssl.use-sslv2 = "devre dışı bırak"

ssl.use-sslv3 = "devre dışı bırak"

ssl.openssl.ssl-conf-cmd = ("Protokol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 veya üzeri

ssl.ec-eğrisi = "secp384r1"

**Malik Sarrafoğlu - 195541608**

# Missing 'Content-Type' Header(Eksik İçerik Türü Başlığı)



**LOW**

## Tanım

Content-Type başlığı, istemcilerin verileri işlemek için uygun bir yol bulmasına olanak tanır, bu başlığın atlanması MIME koklama saldırılarını kolaylaştırabilir.

## İçerik Türü Nedir?

İçerik Türü alanının amacı, gövdede bulunan verileri, alıcı kullanıcı aracısının verileri kullanıcıya sunmak için uygun bir aracı veya mekanizma seçebileceği veya başka bir şekilde verilerle uygun bir şekilde ilgilenebileceği kadar tam olarak tanımlamaktır.

## MIME Koklama Saldırısı Nedir?

MIME türü koklama, sunucu tarafından gönderilen HTTP üstbilgilerinin sonuçsuz veya eksik olduğu durumlarda verileri işlemenin uygun bir yolunu bulmak için tarayıcılarda standart bir işlevdir.

Bu, Internet Explorer ve Chrome'un eski sürümlerinin yanıt gövdesinde MIME koklaması gerçekleştirmesine olanak tanır ve bu da yanıt gövdesinin amaçlanan içerik türünden farklı bir içerik türü olarak yorumlanmasına ve görüntülenmesine neden olabilir.

Bir web sitesi, kullanıcıların web sunucusunda yayınlanan içeriği yüklemesine izin verdiğinde ortaya sorun çıkar. Bir saldırgan, içeriği web uygulaması tarafından kabul edilecek ve tarayıcı tarafından HTML olarak işlenecek şekilde manipüle ederek XSS (Cross-site Scripting) saldırısı gerçekleştirebiliyorsa, örneğin bir görüntü dosyasına kod enjekte etmek ve bunu yapmak mümkündür. kurban görüntüyü izleyerek yürütür.

## Düzeltme

Kaynakları sunarken, içerik türü başlığını, sunulan kaynağın türüyle uygun şekilde eşleşecek şekilde gönderdiğinizden emin olun. Örneğin, bir HTML sayfası sunuyorsanız, HTTP başlığını göndermelisiniz.

## Örnek

Tarayıcıyı sitenin gönderdiği şeyin uygun içerik türü olduğuna güvenmesi ve gerçek içerik türünü "koklamaya" çalışmaması konusunda bilgilendirmek için X-Content-Type-Options başlığını "nosniff" değeriyle ekleyin.

X-Content-Type-Options: nosniff.

**Zafer Koç - 195541305**

# [Possible] Backup File Disclosure(Yedek Dosya İfşası)



**LOW**

## Nedir

Sistemde kullanılmayan ama aynı zamanda da sistemde var olan dosyalar olduğunu belirtir. Genellikle geliştiriciler tarafından çalışmalarını yedeklemek için oluşturulan bu dosyalar birçok açıdan güvenlik açığı barındırabilir.

## Özet

Web sunucusunda olası bir yedekleme dosyası ifşası tespit etti.

## Etki

Yedekleme dosyaları, web sunucusundaki bir dosyanın eski veya güncel sürümlerini içerebilir. Bu, parola dosyaları ve hatta uygulamanın kaynak kodu gibi hassas verileri içerebilir. Bu tür bir sorun normalde daha fazla güvenlik açığına veya en kötü ihtimalle hassas bilgilerin ifşa edilmesine yol açar.

## Düzeltme

Yedekleme dosyalarını üretim sunucularında saklamayın.

# Internal Server Error (İç Sunucu Hatası)



**LOW**

## Nedir

500 Internal Server Error Hatası, genellikle sitenizin. htaccess dosyasının yanlış yapılandırılması veya web sitenizdeki aşırı yoğunluk sonucunda sunucunun yanıt verememesine neden olan bir hatadır.

## Özet

Dahili bir sunucu hatası tespit etti.

Sunucu, sunucu tarafında bir hata olduğunu belirten bir HTTP durumu 500 ile yanıt verdi. Sebepler değişebilir ve davranış dikkatlice analiz edilmelidir. Aynı kaynakta bir güvenlik sorunu bulabilirse, bunu ayrı bir güvenlik açığı olarak bildirir.

## Etki

Etki, duruma bağlı olarak değişebilir. Genellikle bu, yetersiz kodlama uygulamalarını, yeterli hata denetimi, temizleme ve beyaz listeye alma olmadığını gösterir. Ancak, SQL enjeksiyonu gibi daha büyük bir sorun olabilir. Bu durumda, Netsparker diğer olası sorunları kontrol edecek ve bunları ayrı olarak bildirecektir.

## Düzeltme

Beklenmeyen hataları işlemek için bu sorunu analiz edin ve uygulama kodunu inceleyin; bu, bir hata hakkında daha fazla bilgi ifşa etmeyen genel bir uygulama olmalıdır. Tüm hatalar yalnızca sunucu tarafında ele alınmalıdır.

**HASSAN SANUSI BAYERO - 205541608**

# Insecure Frame (External)



**LOW**

## Summary

Impact

IFrame sandboxing enables a set of additional restrictions for the content within a frame in order to restrict its potentially malicious code from causing harm to the web page that embeds it.

The Same Origin Policy (SOP) will prevent JavaScript code from one origin from accessing properties and functions - as well as HTTP responses - of different origins. The access is only allowed if the protocol, port and also the domain match exactly.

Here is an example, the URLs below all belong to the same origin as http://site.com :

http://site.com

http://site.com/

http://site.com/my/page.html

Whereas the URLs mentioned below aren't from the same origin as http://site.com :

http://www.site.com (a sub domain)

http://site.org (different top level domain)

https://site.com (different protocol)

http://site.com:8080 (different port)

When the sandbox attribute is set, the iframe content is treated as being from a unique origin, even if its hostname, port and protocol match exactly. Additionally, sandboxed content is re-hosted in the browser with the following restrictions:

• Any kind of plugin, such as ActiveX, Flash, or Silverlight will be disabled for the iframe.

• Forms are disabled. The hosted content is not allowed to make forms post back to any target.

• Scripts are disabled. JavaScript is disabled and will not execute.

• Links to other browsing contexts are disabled. An anchor tag targeting different browser levels will not execute.

• Unique origin treatment. All content is treated under a unique origin. The content is not able to traverse the DOM or read cookie information.

When the sandbox attribute is not set or not configured correctly, your application might be at risk.

A compromised website that is loaded in such an insecure iframe might affect the parent web application. These are just a few examples of how such an insecure frame might affect its parent:

• It might trick the user into supplying a username and password to the site loaded inside the iframe.

• It might navigate the parent window to a phishing page.

• It might execute untrusted code.

• It could show a popup, appearing to come from the parent site.

Sandbox containing a value of :

• allow-same-origin will not treat it as a unique origin.

• allow-top-navigation will allow code in the iframe to navigate the parent somewhere else, e.g. by changing parent.location.

• allow-forms will allow form submissions from inside the iframe.

• allow-popups will allow popups.

• allow-scripts will allow malicious script execution however it won't allow to create popups.

Remediation

• Apply sandboxing in inline frame <iframe sandbox src="framed-page-url"></iframe>.