

XSS (Cross Site Scripting) Güvenlik ihlali

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

XSS Güvenlik İhlali

- İlgili web sayfasına kod enjeksiyonu yapılması ile gerçekleştirilmektedir
- Üç tipi mevcuttur
 - Enjekte edilen kodun uygulamada kaydedildiği durum – tüm kullanıcıları etkilemesi olasıdır
 - Enjekte edilen kodun kaydedilmeden tek kullanıcıyı etkilediği durum
 - DOM temelli enjeksiyon

HTML Özel Karakterler

Özel Karakter	İşlevi	Dönüştürülmesi Gereken
<	HTML Tag başlangıç	<
>	HTML Tag bitiş	>
"	Bir özelliğin belirtilmesinde kullanılır	" " "
'	Bir özelliğin belirtilmesinde kullanılır	' ' '
&	Öğelerin başlangıcında kullanılır	&

@System.Web.HttpUtility.HtmlEncode()

@System.Text.Encodings.Web.HtmlEncoder.Default.Encode()

JSON Döndüren API

- JSON verisi döndüren API'larda döndürülen içeriğin MIME türü application/json olmalıdır. Text/html olursa web sayfası gibi yorumlanır ve kod çalıştırılır.

JavaScript kodu içerisinde özel karakterler

Özel Karakter	JavaScript Muadili
<	\x3c
>	\x3e
'	\x27

@System.Web.HttpUtility.JavaScriptStringEncode()

@System.Text.Encodings.Web.JavaScriptEncoder.Default.Encode()

Content Security Policy

- Tarayıcıya hangi kaynaklardan veri yükleyebileceği, satır içi kod çalıştırıp çalıştırılamayacağı gibi konularda talimat iletebiliriz.
- Bilhassa renderlamanın browserda gerçekleştiği uygulamalarda (tek sayfadan oluşan uygulamalar) güvenlik açıklarını önleyen en etkili yöntemdir.
- 2. versiyonu şu anda geçerlidir. 3. versiyonu taslak halindedir.
- Response headerlar vasıtası ile direktifler tarayıcıya iletilirler

Default-src ‘self’ ve diğer direktifler

Tarayıcıya default-src ‘self’ iletiliği anda dış kaynaklardan yüklenen tüm dokuman ve içerikler ile satır içi komutlar durdurulur. Direktifler arasında en genel olanıdır.

Direktif adı	İlgili olduğu içerik türleri
child-src	iframes ve web workers
connect-src	HTTP ve WebSocket talepleri (XMLHttpRequest, fetch())
font-src	Web fontları
img-src	Resimler
media-src	Ses ve video dosyaları
object-src	Eklentiler tarafından yürütülen veriler
script-src	JavaScript kodu
style-src	CSS stilleri

META Tag'ı ile CSP Tanımlama

- Aşağıdaki örnekte görüldüğü gibi html meta tag'ı ile de aktarılabilir. Meta tag'ı daha kısıtlı imkanlar sunmaktadır. CDN'lere yüklenecek içeriklerde tercih edilmektedir.

```
<meta http-equiv='Content-Security-Policy' content='default-src 'self'>
```

NONCE

- Dokuman içerisine gömülü kod satırlarının header-satır içi nonce özelliği şifre eşleştirme ile çalıştırılabilmesini sağlar.

SRI – Subresource Integrity

- Harici kütüphanelerde tanımlı integrity attribute’ı (özellik) ile NONCE’dekine benzer şekilde şifre eşleştirme ile harici kütüphanenin orijinalliginden emin olunur.

Diğer Bazi Komutlar

- Unsafe-eval – dinamik yöntemlere (eval, setInterval, setTimeout, sort vb) izin verilir
- Unsafe-inline – satır içi kod kullanımına izin verilir (javascript – style).
- Form-action – hangi domainlere form submit edilebilir kısıtlanabilir
- Report-uri – bir sorun ile karşılaşıldığında tanımlanan linke raporlar
- Report-sample – hatanın gerçekleştiği kısımdan kod örneğini de ekler
- Strict-dynamic – javascript kodunun dinamik şekilde siteye yeni javascript kodu eklemesini sağlar
- Navigate-to – mevcut sayfadan hangi sayfaya yönlendirileceğini kısıtlar
- Base-uri – base taginde kullanılabilecek domainleri sınırlar

Content-Security-Policy-Report-Only

- Herhangi bir kısıtlama getirmeden sadece raporlar

Hata raporu örneği

```
• {  
•   "csp-report": {  
•     "document-uri": "https://localhost:44373/CSP",  
•     "referrer": "https://localhost:44373/CSP",  
•     "violated-directive": "connect-src",  
•     "effective-directive": "connect-src",  
•     "original-policy": "default-src 'self'; img-src 'self';  
•     "style-src 'self' 'unsafe-hashes' 'sha256-yckz1zrIL2HgQwm7x1ins99s5jndZE3XnmgOAkJvDOg=';  
•     "script-src 'self' https://cdn.jsdelivr.net 'sha256-nzwB34aHhENPZBCXYw2IW6Io6IJ79  
•     "hTi15ZZbN/nP9I=' 'unsafe-eval'; report-uri /collect",  
•     "disposition": "enforce",  
•     "blocked-uri": "ws://localhost:49326",  
•     "line-number": 5,  
•     "column-number": 18,  
•     "source-file": "https://localhost:44373/\_framework/aspnetcore-browser-refresh.js",  
•     "status-code": 0,  
•     "script-sample": ""  
•   }  
• }
```

Kullanıcı Oturum Yönetimine Karşı Güvenlik İhlalleri

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

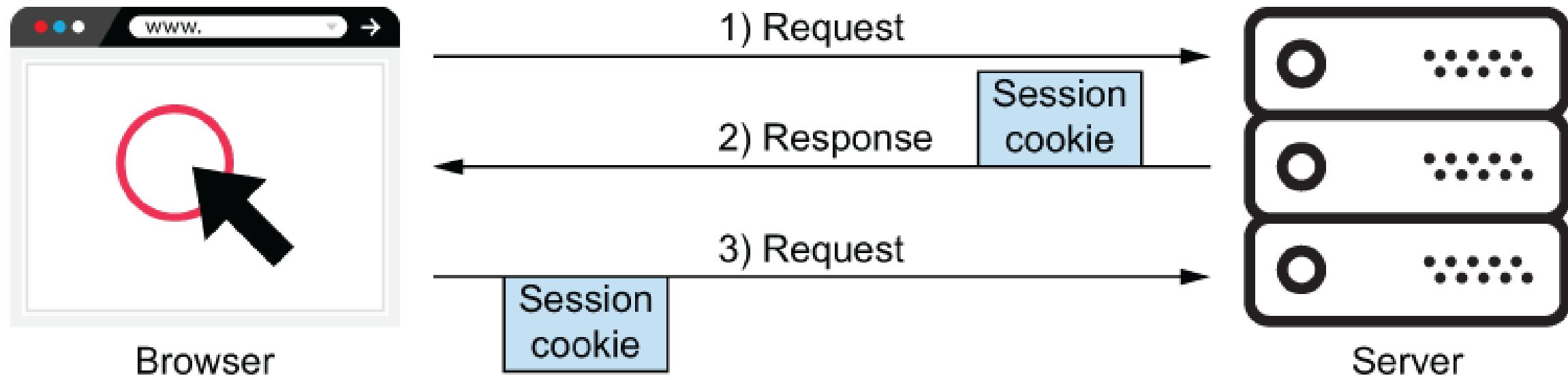
Dr. Aydın Erden

HTTP bağlantı üzerinden bilgi çalınması

- 2010 yılında Firesheep isimli bir Firefox eklentisi aynı WiFi bağlantısını kullanan kişilerin HTTP bağlantı üzerinden hesaplarına erişirken ettiği oturum cookielerini çalarak o kişinin hesabına giriş yapılmasına imkan sağlamıştır.
- Bunun gerçekleşebilmesinin nedenleri:
 - HTTP bağlantı veriyi şifresiz şekilde iletmektedir
 - Wi-Fi ağları ve HUB kullanan kablolu ağlarda bir paket tüm kullanıcılar iletilmektedir. Yani başka kullanıcıya gönderilmiş paketleri okumak mümkündür.

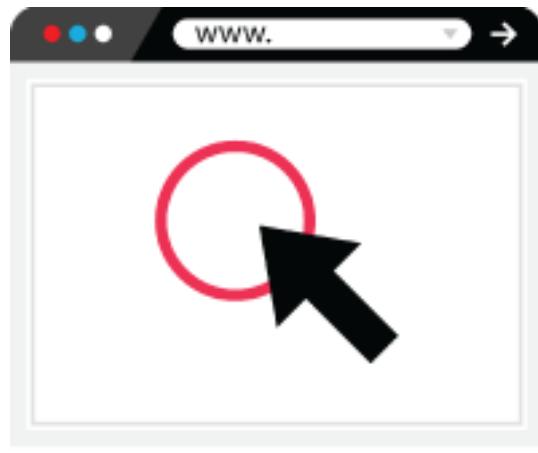
Oturum Çalınması Örnek Uygulama

- Asp.Net oturum için AspNetCore.Session isimli bir cookie tanımlamaktadır. Bu cookie'de bulunan benzersiz bir değer her bir talepte uygulamaya iletilemektedir.
- İletilen bu değeri alıp bir başka browserdaki aynı isimli cookie'ye aktarırsanız bir başkasının oturumuna bağlanırsınız.
- HTTP ve HTTPS stateless'tir. Yani kullanıcıya ait herhangi bir veri saklamaz. O nedenle oturum yönetimi için cookieler kullanılır.

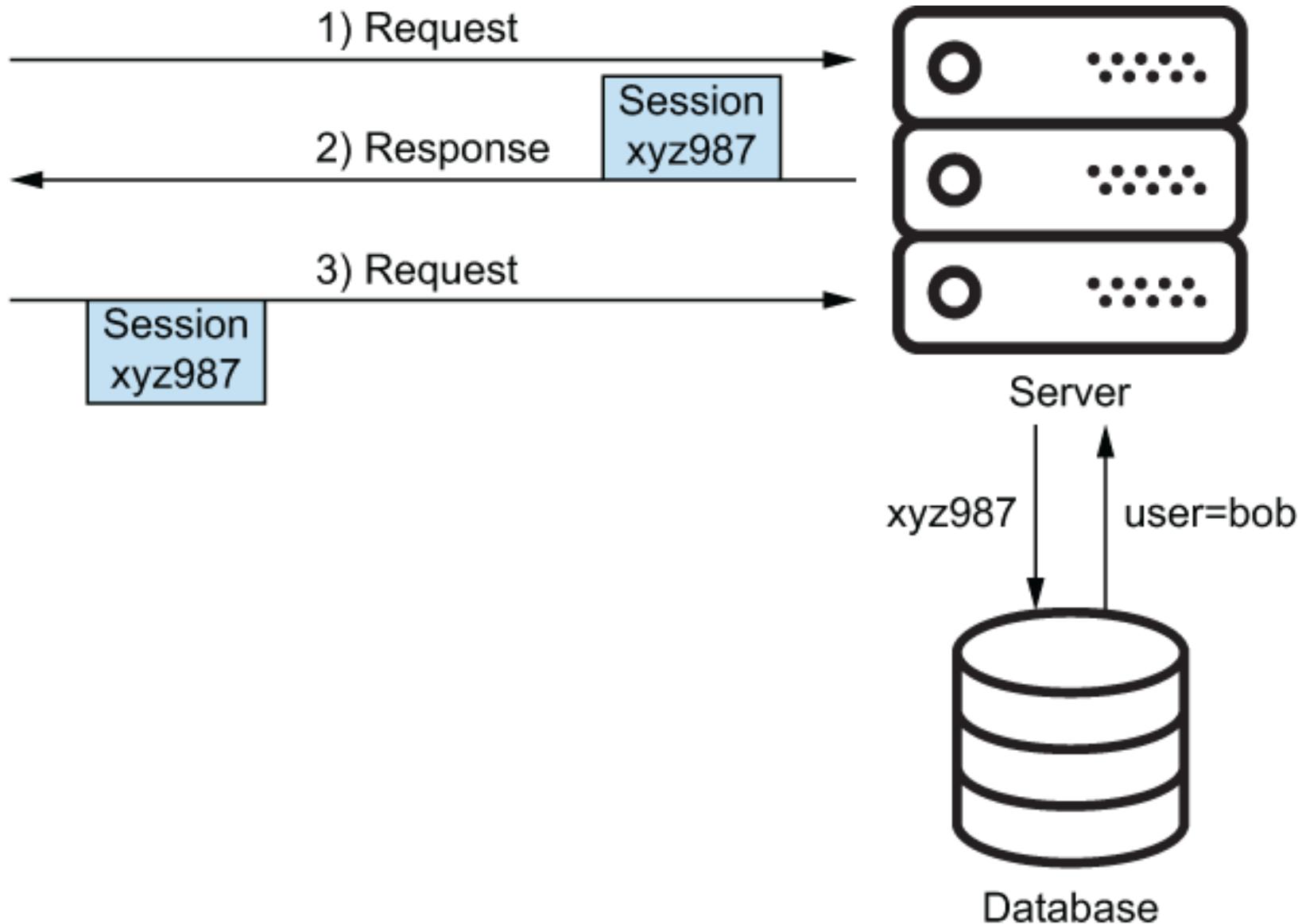


Cookie

- Set-Cookie HTTP Header – server'dan gönderilir. Cookie'nin browser'a kayıt edilmesini sağlar.
- Cookie HTTP Header – browser tarafından gönderilir. Kayıtlı cookie'nin server'a gönderimini sağlar.



Browser



Browser'da Veri Saklamak

- Single Page Application'larda kullanılır. İki yerde veri saklanabilir:
 - sessionStorage – ilgili browser tab'ı kapatıldığı anda silinir
 - localStorage
- Bundaki risk bir önceki bölümde gördüğümüz XSS saldırısı – kod enjeksiyonu riskidir.

Session ID'sini Çalmanın Üç Yolu

- XSS
- SQL enjeksiyonu
- HTTP bağlantı ile gönderilen şifresiz veriyi okuma

XSS ile oturum ID'si Çalma

- Bunu önlemek için önceki bölümde gördüğümüz XSS önlemlerine ilaveten cookie'de HttpOnly aktif hale getirilir. Böylece browser'da saklanan cookie Javascript kullanılarak okunamaz.
- document.cookie komutu ile HttpOnly olmayan cookieler okunabilir.
- Öte yandan eğer server ayarlarında (uygulama değil serverin kendisi) TRACE metodu aktif ise HttpOnly olmasına rağmen cookieler okunabilir

Cookilerde secure ayarı

- Cookilerde secure özelliği aktif hale getirilir ise o cookie'nin sadece HTTPS bağlantısı ile iletilmesi garanti edilir.
- Üç seçenek mevcuttur
 - None – hiçbir şekilde HTTPS ile göndermez
 - Always – Daima HTTPS ile gönderilir
 - SameAsRequest – Kullanıcı hangi protokolü kullandı ise (HTTP veya HTTPS) aynı yöntem ile gönderilir

Olası ID Çalınmasına Karşı Oturum Süresi Sınırlanabilir

- Varsayılan olarak ASP.NET'te oturum süresi 20dk'dır.

HTTPS Kullanımına Zorlama

- HTTP bağlantı ile veriler şifresiz olarak iletilirler
- HTTP ile gelen talepler uygulama içerisinde otomatik olarak HTTPS'ye yönlendirilecek şekilde ayarlanmalıdır
- Yönlendirme varsayılan olarak 307 HTTP durum kodu ile yapılır
- 302 ile de yönlendirme yapmak mümkün olsa da 302 sadece GET talebi ilettiğinden POST talepleri de GET olarak iletilir ve çalışmaz.
- Bu ayarlansa bile browserdan gerçekleşecek ilk bağlantı HTTP ile olacaktır. Bu bile bir güvenlik riskidir. Bu riskin yinelenmemesi için browser HSTS - HTTP Strict Transport Security ile bilgilendirilebilir. Böylece ilk talep, hatta erişilmek istenen domain bilgisi bile servera şifreli olarak iletilebilir.
- Bazı sitelerde HTTP ile bağlantı kurmak isteseniz bile browser otomatik olarak HTTPS bağlantıya yönlendirecektir.

HSTS - HTTP Strict Transport Security

- Browser'a bundan sonra bu site için sadece HTTPS kullan diyen yöntemdi. Bu komut her bir site için sitenin ilettiği komutun süresi boyunca geçerlidir.
- Response Header'da şu şekilde görünür: Strict-Transport-Security: max-age=63072000; includeSubDomains;
- Max-age talimat geçerlilik süresidir. Birimi saniyedir
- includeSubDomains alt domainler için de bu talimat geçerli demektir
- Kimi durumlarda ilk bağlantıda bile otomatik HTTPS bağlantısı gerçekleşir. Bu browser içerisinde HSTS başlığı altında kayıtlı site bilgileri ile mümkün olabilmektedir.

HSTS Preload

- Browserda HSTS başlığı altında önkayıtlı siteler
- Bu şekilde kaydetmek için yapılması gerekenler:
 - Sitedeki tüm sayfaların HTTPS'i desteklemesi gerekmektedir. Bu şekilde kayıtlandıktan sonra desteklemeyen sayfalara bağlanmak mümkün olmayacağı.
 - Web uygulaması HSTS desteklemelidir.
 - Web uygulaması HSTS ayarlarında ASP.NET'te options.Preload=true yapılmalıdır

Enter a domain:

example.com

Check HSTS preload status and eligibility

Information

This form is used to submit domains for inclusion in Chrome's [HTTP Strict Transport Security \(HSTS\)](#) preload list. This is a list of sites that are hardcoded into Chrome as being HTTPS only.

Most major browsers (Chrome, [Firefox](#), Opera, Safari, [IE 11 and Edge](#)) also have HSTS preload lists based on the Chrome list. (See the [HSTS compatibility matrix](#).)

Browser'a manuel olarak HSTS özellikli site ekleme

- chrome://net-internals/#hsts
- edge://net-internals/#hsts

Oturuma Sızıldığını Nasıl Anlayabilirsiniz?

- Oturuma sizildığını anlamanın kesin bir yolu yoktur. Fakat aşağıdaki iki yöntem ipucu verebilir.
 - IP adres değişikliği – fakat bunun başka nedenleri de olabilir mesela kullanıcı wifi'dan mobil bağlantıya geçmiş olabilir.
 - HTTP request headerlardaki şüpheli değişimler – mesela kullanıcı Chrome tarayıcı kullanırken aniden Firefox kullanmaya başladı ise şüphe duymak gereklidir

Siteler Arası İstek Sahteciliği

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

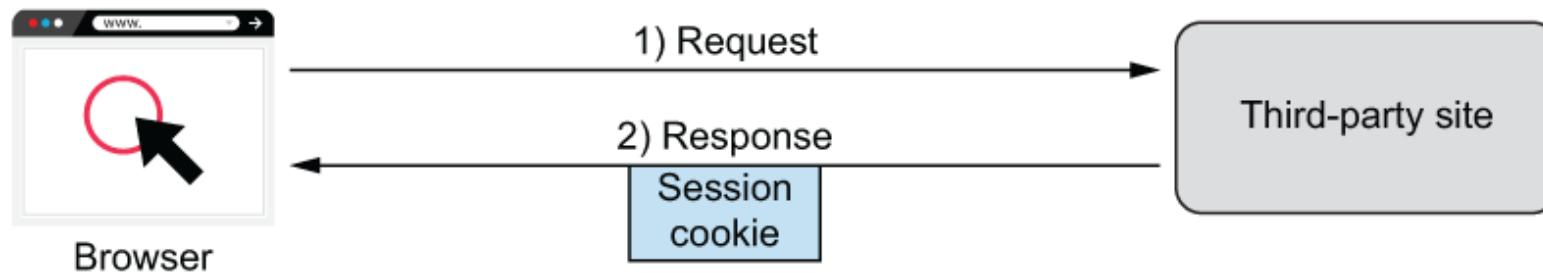
Değineceğimiz Konular

- Siteler Arası İstek Sahteciliği, yarattığı güvenlik riskleri ve nasıl önlenebileceği
- Cookielerin korunması yolu ile siteler arası istek sahtecığının önlenmesi
- Tıklama hırsızlığı ve siteler arası istek sahteciliği ile bağlantısı

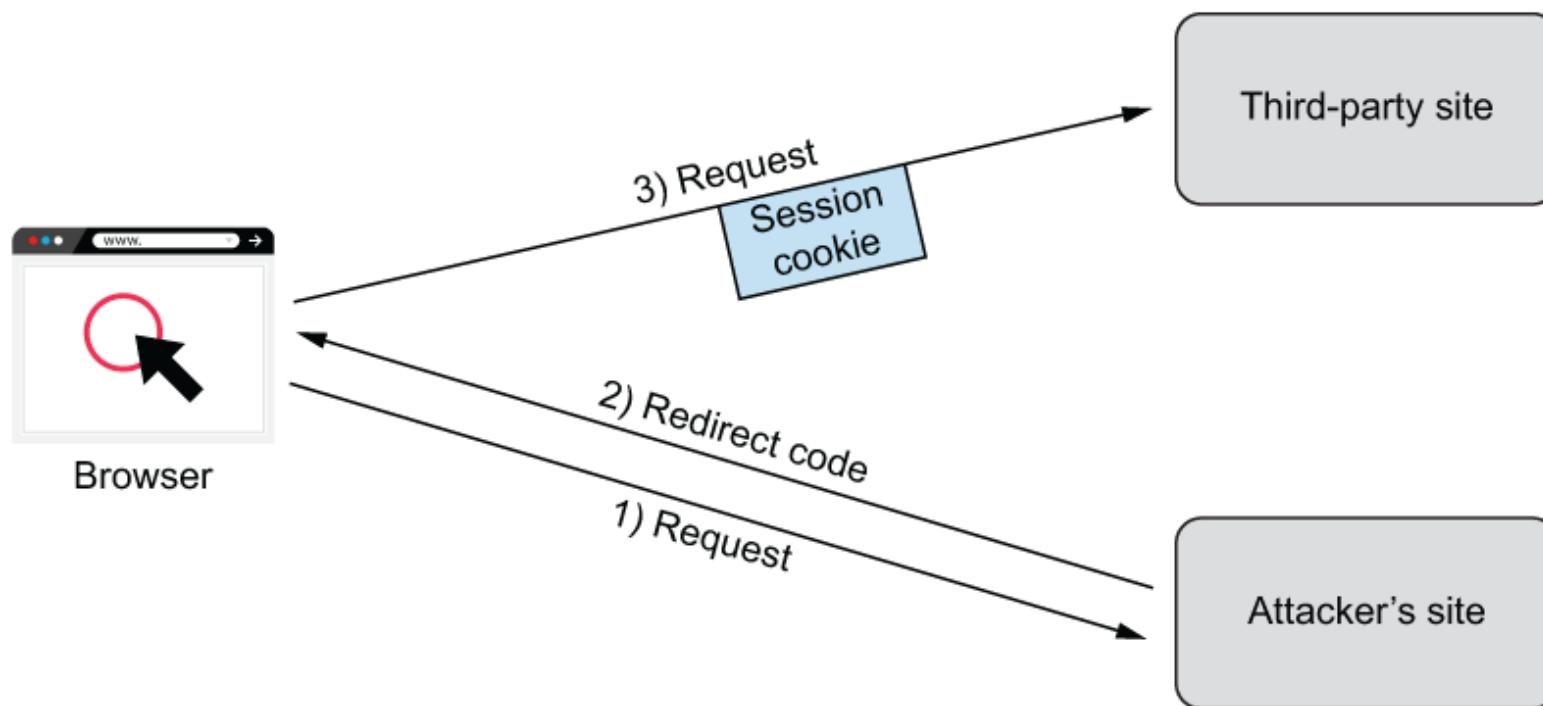
Yaşanmış Örnekler

- 2005 yılında Samy Kamkar MySpace sitesinde bulduğu açığı kod enjeksiyonu yöntemi (XSS) ile kullanarak sahteciliğe maruz kalan kullanıcı adına kendisine arkadaşlık isteği gönderilmesini sağlamıştır.
- 2018 yılında DrayTek firması kendilerine ait routlerlarda bir açık olduğunu bunun siteler arası istek sahteciliği yöntemi ile router dns ayarlarının değiştirilebildiğini duyurmuştur.
<https://www.draytek.co.uk/support/security-advisories/kb-advisory-csrf-and-dns-dhcp-web-attacks>

Step 1: User interacts with site



Step 2: CSRF attack



GET ile veri talebinde bulunurken

- Location.href kullanılır. Ya bir <iframe> veya taginde href özelliği kullanılarak oturumun açık olduğu siteye yönlendirme yapılır

POST ile veri talebinde bulunurken

- Saldırgana ait sitede <form> mevcuttur ve bu form diğer siteye POST yöntemi ile veri iletir. Elbette bu formun POST ile iletilmesi için başka birtakım JavaScript kodları da bulunur.

POST Yöntemi ile Sahtecilikte Asgari Şartlar

- Sistemin işleyışı hakkında bilgi – HTTP talebinin detayları
- Kullanıcının oturum ID'si

HTTP Talebini Tahmin Edilemez Hale Getirme

- Bunun için token kullanılır. Forma bir veri daha eklenir ve token bilgisi bu form ögesi ile iletilir. Aynı anda AntiForgery isimli cookie'de de aynı token kayıtlıdır ve bu cookie de browser aracılığı ile iletilir. İki token aynı ise form kaydedilir değil ise kaydedilmez.

Form Öğesi:

```
<input name='__RequestVerificationToken' type='hidden'  
value='CfDJ8FflGUpI_sxPt3SJyGw1tBgM8keVYHfkxE19T2rR10nRpI0j4d  
Ayr-  
yplT4caP6xPy807LVMrTzSRC0vRPOaUfJcqm6U8z4y3LePaCEICKCw0FVA  
z3z-3V694vTIYpGckc97w08oOQSCfxBrucwU90c' />
```

Cookie:

```
.AspNetCore.Antiforgery.Za7zYHoQn5w=CfDJ8FflGUpI  
_sxPt3SJyGw1tBhPF5OXlg6aFVm0YmyUzF29aTY-  
OBqu9g9jGqOdsqSDWGCDrO1LadtSyRd  
BiKBZemMxW6znfiVY5IZ5F4JqGcBCwDI8UIQgB9iqQaoXz8HWbWsB8M  
a7JFK3j8RRPVFOuel
```

- Aşağıdaki iki yöntem ile program içerisinde form oluşturmadığınız sürece Asp.Net'te bu koruma aktif haldedir ve ek işlem yapmanız gereklidir.
 - <form method='post'>...</form>
 - @Html.BeginForm(...)

Bazı Ek Ayarlar

- `options.FormFieldName = "AntiForgeryToken"; // FORMDA TOKEN İLE İLGİLİ INPUT ALANININ NAME'İ BU ŞEKİLDE DEĞİŞTİRİLYOR`
- `options.HeaderName = 'X-Anti-Xsrf-Token'; //TOKEN'İN HTTP HEADER VASITASI İLE İLETİLMESİSİ SAĞLAR`

Cookieleri Korumak

- Same Site Politikası
- options.Cookie.SameSite = SameSiteMode.Strict;
- Seçenekler
 - Strict – başka site sizi yönlendirecek olur ise cookie'yi göndermez
 - Lax – sadece GET, HEAD, OPTIONS, veya TRACE taleplerinde iletilir, POST, PUT, DELETE ve PATCH taleplerinde iletilmez
 - None – herhangi bir koruma yoktur, her koşulda gönderir

Chrome Tabanlı Browserlarda Varsayılan SameSite cookie ayarı Lax'tır.

Chrome Yaptırımı Şubat 2020'de Başlıyor

Şubat ayında kullanıma sunulacak Chrome 80 sürümünde, SameSite değeri belirtilmemiş çerezler `SameSite=Lax` çerezleri olarak değerlendirilecektir. Yalnızca `SameSite=None ; Secure` ayarı yapılmış çerezlere, bağlantıların güvenli

<https://developers.google.com/search/blog/2020/01/get-ready-for-new-samesitenone-secure?hl=tr>

Cookieleri Güvenli Hale Getirme

```
builder.Services.AddAntiforgery(options =>
{
    options.Cookie.SameSite = SameSiteMode.Strict;
    options.Cookie.SecurePolicy = CookieSecurePolicy.Always;
    options.Cookie.HttpOnly = true;
});
```

Origin – Kaynak

https://www.example.com:443

scheme host name port

- Origin (kaynak), scheme diğer ifade ile protocol, host name ve port'un birleşimidir.
- https://www.example.com:443/foo örnek sitesinde origin https://www.example.com:443 'dir.

Same Origin – Cross Origin Örnekler

Kaynak A	Kaynak B	A ve B Kaynaklarının "aynı kaynak" mı yoksa "çapraz kaynak" mı olduğuna dair açıklama
https://www.example.com:443	https://www.evil.com:443	Cross-origin: farklı alanlar
	https://example.com:443	cross-origin: farklı alt alan adları
	https://login.example.com:443	cross-origin: farklı alt alan adları
	http://www.example.com:443	cross-origin: farklı şemalar
	https://www.example.com:80	Cross-origin: farklı bağlantı noktaları
	https://www.example.com:443	same-origin: tam eşleme
	https://www.example.com	same-origin: implicit bağlantı noktası numarası (443) ile eşleşir

Site

https://www.example.com:443

scheme TLD
TLD+1

- .com ve .org gibi üst düzey alanlar (TLD'ler), Kök Bölge Veritabanı'nda listelenir. Yukarıdaki örnekte "site"; scheme (protocol), TLD ve alanın kendisinden hemen önceki bölümünün (TLD+1 olarak adlandırılır) kombinasyonudur.
- `https://www.example.com:443/fooooo` URL'sinde "site" `https://example.com` olur.

eTLD – effective Top Level Domain



- Tüm dünyadaki eTLD listesi için:
[publicsuffix.org/list/public suffix list.dat](https://publicsuffix.org/list/public_suffix_list.dat)
- Örnek olarak `https://www.project.github.io:443/foo` şeması `https`, eTLD `.github.io`, eTLD+1 ise `project.github.io` şeklindedir. Bu nedenle, `https://project.github.io` bu URL için "site" olarak kabul edilir.

Same Site – Cross Site

Kaynak A	Kaynak B	A ve B Kaynağının "aynı site" mi yoksa "çapraz site" mi olduğuna ilişkin açıklama
<code>https://www.example.com:443</code>	<code>https://www.evil.com:443</code>	Cross-site: farklı alan adları
	<code>https://login.example.com:443</code>	same-site: farklı alt alan adları önemli değildir
	<code>http://www.example.com:443</code>	çapraz-site: farklı şemalar
	<code>https://www.example.com:80</code>	same-site: farklı bağlantı noktaları önemli değildir
	<code>https://www.example.com:443</code>	same-site: tam eşleme
	<code>https://www.example.com</code>	same-site: bağlantı noktaları önemli değildir

Tıklama Hırsızlığı

- Kullanıcıyı aslında neye tıkladığı konusunda yanıltma durumu

X-FRAME-OPTIONS

- DENY – sayfa hiçbir şekilde <iframe></iframe> içerisinde yüklenmez
- SAMEORIGIN – sayfa sadece <iframe></iframe> In yer aldığı site ile iframe'de açılmasına çalışan site Same Origin ise yüklenir
- Asp.Net'te Antiforgery active edildiğinde X-FRAME-OPTIONS: SAMEORIGIN headerini otomatik olarak gönderir. Derste örneğimiz çalışın diye biz bunu aşağıdaki komut ile kapattık.

```
builder.Services.AddAntiforgery(options =>
{
    options.SuppressXFrameOptionsHeader = true;
});
```

Frame-ancestors Kullanımı

- Content-Security-Policy: frame-ancestors 'none';
- Content-Security-Policy: frame-ancestors 'self'
<https://www.example.org>;
- Content-Security-Policy: frame-ancestors 'self' <https://example.org>
<https://example.com> <https://store.example.com>;

Cross-Origin Resource Sharing - CORS

- Farklı orijinli siteler ile API aracılığı ile veri paylaşımı için kullanılır
- Erişime izin vermek için response header'da Access-Control-Allow-Origin: <https://localhost:7201>

CORS Origin Ayarı

```
builder.Services.AddCors(options =>
{
    options.AddPolicy(
        "CORS API",
        builder =>
    {
        builder.WithOrigins("https://localhost:7201");
    });
});
```

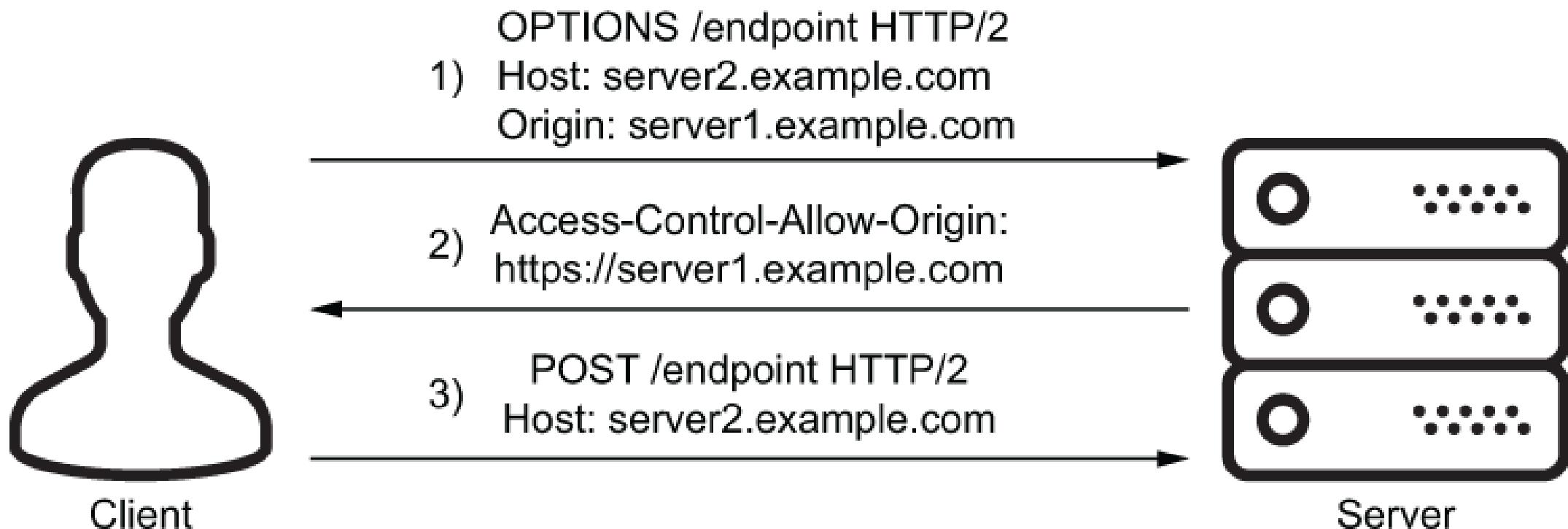
CORS Seçenekler

- Tüm sisteme erişim açılabilir
- Sadece bazı kontrolrlere veya metotlara erişim açılabilir
- Sadece bazı domain pathleri için erişim açılabilir
- Kullanılabilecek metodların bilgisinin kullanıcıya headerlar vasıtası ile iletimi mümkündür

GET Yöntemi CORS Süreci

- Talep browser tarafından servera direct olarak iletilir ama erişim hakkımız yok ise browser tarafından gösterilmez. GET yöntemi server tarafından bir değişikliğe yol açmadığından bu yöntem izlenir.

POST Yöntemi CORS Süreçler



Veri Doğrulama

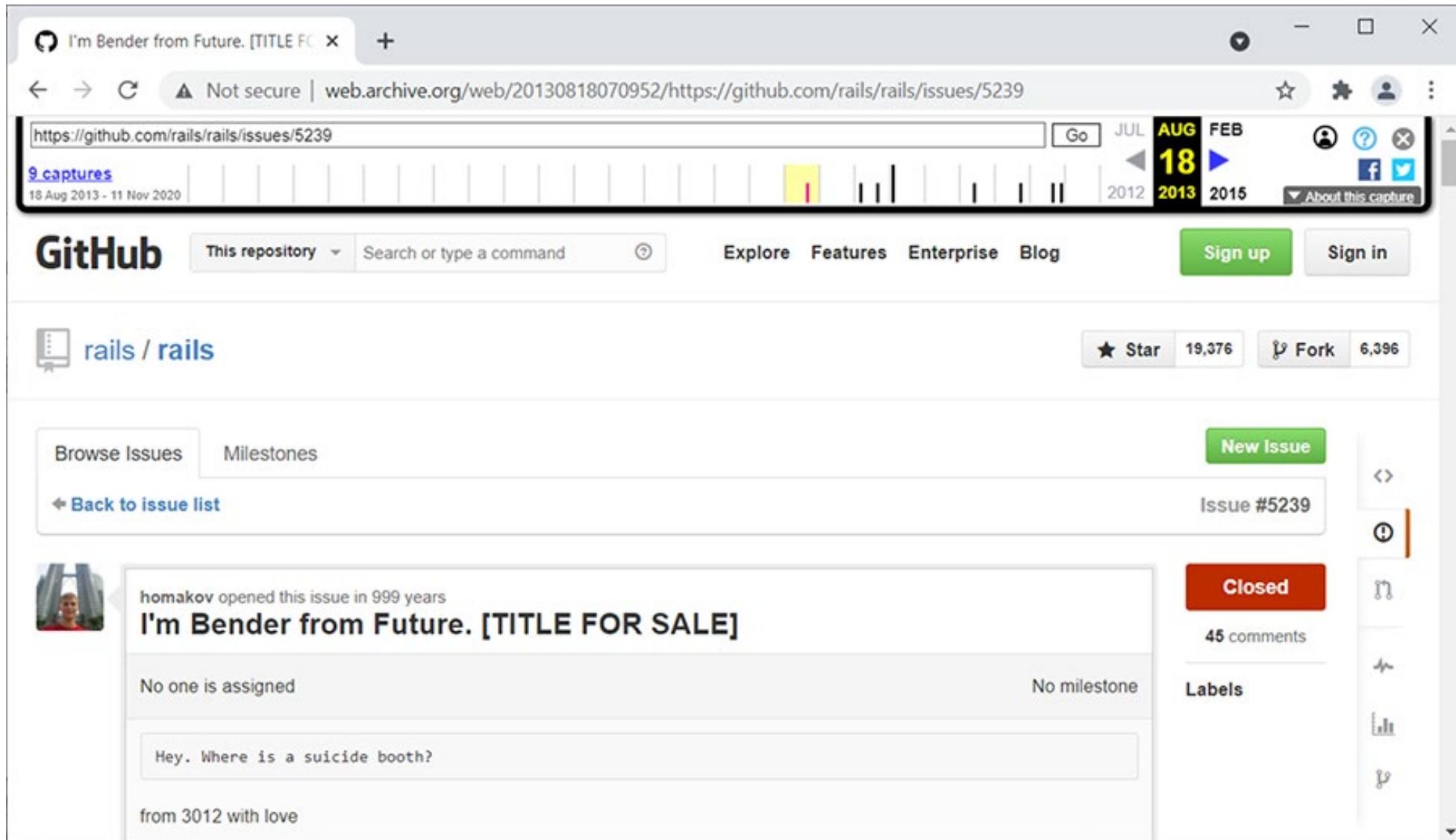
SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Konular

- HTTP'nin hangi kısımlarının manipüle edilebileceği
- ASP.NET'te veri doğrulaması
- Mass Assignment

- 2012 yılında Rus geliştirici Egor Homakov GitHub üzerinde Ruby on Rails platformunda Mass Assignment saldırısının gerçekleşebileceği üzerine bir tartışma başlattı. Gelen yanıtlardan memnun olmayınca 2012 yılında Ruby on Rails kullanılan GitHub platformunun kendisini hackledi.



Temel sorun verinin kontrol edilmeden kaydedilmesidir.

HTTP GET

GET / HTTP/1.1

Host: www.example.com

Connection: keep-alive

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Cache-Control: max-age=0

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110

Safari/537.36

HTTP POST

POST /login HTTP/1.1

Host: login.example.com

Connection: keep-alive

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Cache-Control: max-age=0

Content-Length: 5650

Content-Type: application/x-www-form-urlencoded

Cookie: c_i=8b7d22e5-6297-4a72-a8c4-da4b349f7ea8;

cf_bm=796aaf6f8505ce143732cb0bf5fe5570704f9b70-1624959363-1800-
AXD+HXmAXJF166DRun4wgMAN0pH32mzn/VVgFkyoNQNN3AlioVtMdTNbrx419oEGvHIGH+TdpC1NW9Px6vtCqDsKKVl3zrISkXNEXKHp

0ExULqnil0Y1ZfH+k7/G4KW4GI1rxCcaksfxIVxA+cJ3KWtus8nZ/qxiKh1WK3P/iBNxb+lWbu7ggb03dWPj8tJPQ==

Origin: https://login.example.com

Referer: https://login.example.com/login?service=https%3A%2F%2Fwww.example.com%2Flogin%2Fcas

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36

Browser bu sayfaya bir talep ilettiğinde:
<https://example.com/page?param=value>

Server ve browser şifreleme konusunda teyitleşirler. Şifreleme konusunda mutabık kalırlarsa protocol HTTP'den HTTPS'e çevrilir.

Sonrasında GET yönetemi için aşağıdaki header servera ilettilir.

GET /page?param=value HTTP/1.1

Host: example.com

İletilen bu talep serverin talebi işleyebilmesi için gerekli asgari bilgileri içermektedir. Diğer headerlar ise muhtelif metadata verilerini içerirler. Bu veriler kullanıcı tarafından iletildiğinden hertürlü manipüle edilebilirler ve uygulama kullanıcısından gelen bu bilgilere güvenmemelidir.

Accept Headerı

Accept—Bu header browserin kabul edebildiği MIME (Multipurpose Internet Mail Extensions) türlerinin listesini içerir. Genel olarak HTML, bazı resim formatları belirtilir. Bazı API'lar bu headera bakarak veriyi hangi formatta döndürmesi gerektiğine karar verirler. Kod enjeksiyonu dersinde gördüğümüz gibi JSON yerine text/html döndürülp bir de javascript kodu enjekte edilecek olur ise kullanıcının tarayıcısında istenmeyen bir kod çalıştırılabilir.

Cookie Headerı

Cookie — Kullanıcı oturumuna yönelik saldırılarla gördüğümüz gibi cookieler kullanıcının tarayıcısında saklanır. Yeterli güvenlik önlemi alınmaz ise manipüle edilebilir. Kullanıcının kendisi de bu cookieleri manipüle edebilir.

Origin Headerı

Origin — Bilhassa CORS taleplerinde (cross-domain) tarayıcı origin headerini otomatik olarak ekler. Origin headerı diğer bir domainı koruma amacı ile kullanılır. Öte yandan tamamen manipüle edilemez değildir. Farklı yöntemler ile origin headerı değiştirilebilir. Bilhassa yetkilendirme için origin headerı kullanılmamalıdır.

Referer Headerı

Referer — Bu header kullanıcının yönlendirildiği site tarafından eklenir. Ayrıca siteye ait resimlerin serverdan yüklenmesi talep edilirken de request headerları arasında referrer headerı olur. Bu header da manipüle edilebilir. Yetkilendirme işlemleri için kesinlikle kullanılmamalıdır.

User-Agent Headerı

User-Agent — Kullanıcının kullandığı tarayıcının bilgisinin iletiliği headerdir. Mevcut hali ile verimli değildir. Vakti ile muhtelif scriptlerin kullanıcı hangi tarayıcıyı kullanıyor ayırt etmesinde kullanılmıştır. Hiçbir konuda bu headera güvenilmelidir. Manipüle edilebilir. Hatta kod enjeksiyonu bile yapılabilir.

Headerlar vasıtası ile gönderilen veriler manipüle edilebilse de hatalı veri iletimi kaynaklı sorunların çoğu form verilerinden kaynaklanmaktadır.

Headerların hemen ardından, bir boş satır onun da hemen ardından POST verisi gelir.

POST verileri büyük oranda aşağıdaki formatta iletılır.

application/x-www-form-urlencoded

Bu formatta her bir veri, anahtar-veri şeklinde eşlidir ve her bir anahtar-veri birbirinden & sembolü ile ayrılır.

Eğer dosya upload edilecekse çoğunlukla multipart/form-data kullanılır.

Güvenlik Kuralları

- Kullanıcıdan gelen veriye güvenilmemelidir. Her zaman doğrulanmalıdır. Örneğin gelen veri türü doğru mu kontrol edilmelidir.
- Üretilen çıktıdaki özel karakterler için gerekli dönüşümler mutlaka yapılmalıdır. Bu dönüşümler daha önce gördüğümüz HTML özel karakterlerin dönüştürülmesi, JavaScript özel karakterlerin dönüştürülmesi ve son olarak ileride göreceğimiz SQL kod enjeksiyonu ile ilgili dönüşümlerdir.

ASP.NET Veri Doğrulaması

- Model binding kullanılır. Böylece programımız kullanıcıdan gelen verinin nasıl olması gerektiğini bilir.

ASP.NET Doğrulama Özellikleri

Attribute	Tanım
[Required]	Mutlaka veri girilmelidir.
[StringLength(42)]	Girilen string azami 42 karakter uzunlukta olabilir
[Range(10, 99)]	Girilen rakam 10 ile 99 arasında olabilir.
[Compare(nameof(OncekiGirdi))]	Girilen veri OncekiGirdi ile aynı olmalı.
[RegularExpression(@'^[a-zA-Z]{1,42}\$')]	Girilen veri 42 adet küçük veya büyük harften oluşmalı.
[EmailAddress]	Eposta adresi olmalı.
[Url]	Bir web adresi olmalı.
[Remote(action: 'Dogrulama', controller: 'Dogrulamalar')]	Girilen veri Dogrulamalar adlı contollerdaki Dogrulama isimli metot tarafından doğrulanmalıdır
[ValidateNever]	Doğrulama yapma.

Kullanıcı Tarafında Doğrulama

- Javascript aracılığı ile henüz daha talep servera ulaşmadan doğrulanabilir. Bu yöntem çok daha hızlı çalışmakla birlikte kullanıcı tarafından aşılabilir bir yöntemdir. O nedenle server tarafında doğrulama daha güvenilirdir.
- Daha da kötüsü bu doğrulama scripti analiz edilerek server tarafında kullanılan yazılımın iç mimarisi hakkında da bilgi edinilebilir.

Arka Planda Gerçekleşen İşlemler

```
var sorun = new Sorun();
sorun.Baslik = HttpContext.Request.Form['Baslik'];
sorun.Tanim = HttpContext.Request.Form['Tanim'];
```

Id değeri ataması ve tarih ataması program tarafından otomatik gerçekleştirilmekte.

Kullanıcı tarafından POST talebi ile tarih bilgisi de ilettilirse ne olur?

```
var sorun = new Sorun();
sorun.Baslik = HttpContext.Request.Form['Baslik'];
sorun.Tanim = HttpContext.Request.Form['Tanim'];
sorun.OlusturulmaTarihi = Convert.ToDateTime(HttpContext.Request.Form['OlusturulmaTarihi']);
```

Bu sayede kullanıcı istediği tarih için giriş yapabildi – Bu Saldırı türüne “mass assignment” veya Türkçe yetki sahibi olunmayan verilerin girişi denilebilir

VeriDogrulama Listele Olustur

Sorun Liste

[Yeni Oluştur](#)

Başlık	Sorun Tanımı	Oluşturulma Tarihi	
Sisteme bağlanamıyorum	Sistem yanıt vermiyor	11/5/2023 5:00:53 PM	Duzenle Detaylar Sil
Şifre hatası	Şifre hatalı uyarısı alıyorum	11/5/2023 5:00:40 PM	Duzenle Detaylar Sil
Eposta doğrulaması	Eposta doğrulaması gelmiyor	11/5/2023 5:04:02 PM	Duzenle Detaylar Sil
x	x	1/1/2300 12:00:00 AM	Duzenle Detaylar Sil

Sorunun Çözümü

- Sadece ilgili verileri içeren bir model kurup bu model üzerinden kontrollü şekilde veri kaydetmek.
- Bind('Baslik,Tanim')] attribute kullanmak
- [Bind(Exclude='Id,CreationDate')] ters yaklaşım. Neyin olacağını değil olmayacağını belirtiyoruz

Veri Deserialization Güvenliği

- Veri deserialization sıkılıkla yapılan bir işlemidir. Örneğin:
 - HTTP POST verisi application/x-www-form-urlencoded MIME türünde veri ilgili veri yapısına çevrilir
 - JSON türünde veri ilgili veri yapılarına çevrilir

.NET Framework Eski Versiyonlarında Server Tarafında Komut Satırı Komutlarını Çalıştırmak Mümkün Olabilmekteydi

```
var payload = @"""{$type:'System.Windows.Data.ObjectDataProvider,  
PresentationFramework, Version=4.0.0.0, Culture=neutral,  
PublicKeyToken=31bf3856ad364e35','MethodName':'Start','MethodParameters':{$  
type:'System.Collections.ArrayList, mscorelib, Version=4.0.0.0, Culture=neutral,  
PublicKeyToken=b77a5c561934e089',  
'$values':['cmd', '/c notepad']},  
'ObjectInstance':{'$type:'System.Diagnostics.Process, System, Version=4.0.0.0,  
Culture=neutral, PublicKeyToken=b77a5c561934e089'}}};  
var data = JsonConvert.DeserializeObject(payload, new JsonSerializerSettings() {  
    TypeNameHandling = TypeNameHandling.All});
```

Komut Satırı Komutlarının Çalıştırılmasını Önlemek İçin:

Newtonsoft'ta:

```
TypeNameHandling = TypeNameHandling.None
```

ASP.NET'in kendi kütüphanelerinde:

```
Sınıf data = JsonSerializer.Deserialize<Sınıf>(payload);
```

ASP.NET kendi kütüphanelerinde dönüştürmek istediğimiz türü kendimiz belirttiğimizden saldırıya açık herhangi bir açık bırakmamış oluyoruz.

SQL Kod Enjeksiyonu, XML External Entities, Server Side Request Forgery

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Gab.com Vakası

- 2021 yılı Mart ayında gab.com sosyal media platformuna ait 70gb veri çalındı.
- Çalınan veriler arasında kullanıcı şifreleri ve özel mesajlaşmalar da mevcuttu.
- Eski ABD başkanı Donald Trump'ın hesabına ait bilgiler de çalınan veriler arasındaydı.

Giriş

SELECT * FROM users WHERE email='a@a.com' AND password='123456'

Kullanıcı Adı

user@example.com

Şifre

Login

```
SELECT * FROM users WHERE email='a@a.com' AND password='birfikrimyok' OR ''=='
```

birfikrimyok' OR ''=='

- Aslında kullanıcıdan beklenen string formatta kullanıcı adı ve şifresini girmesi. Fakat tek tırnak kullanıp sondaki ve baştaki tırnak işaretlerini girdide yazmayarak SQL sorgusuna bir kıyaslama operatörü daha eklemeyi başardık.

Çözüm 1 – Karakterlerin Değiştirilmesi

- Mesela ' yerine \' kullanımı
- XSS'te gördüğümüz gibi karakterleri değiştirmek bir çözüm gibi görünse de farklı veritabanlarında farklı karakterlerin kullanılması sözkonusudur ve veritabanları arasında bir standart yoktur.

Özel karakterlere örnekler:

- | | |
|---------|---------------------|
| () [] | - gruplama |
| _ % | - joker karakterler |
| ; | - sorgu sonlandırma |
| -- # /* | - yorum başlangıcı |

Çözüm 2 - İki Aşamalı Sorgu

- SQL sorugalarında yer tutucular kullanılır.
- Kullancıdan gelen veriler öncelikle bu yer tutuculara atanır akabinde SQL sorusu olarak gönderilir.
- Böylece neyin SQL komutu ve neyin veri olduğu daha net hale gelir

```
SELECT * FROM users WHERE email=@email AND password=@password
```

Oracle Veritabani

Parametreler : ile ayrılır

```
SELECT * FROM users WHERE email=:email AND password=:password
```

```
public void OnPost(string email, string password)
{
    var connection = new SqlConnection("**connection string**");
    this.SqlQuery = "SELECT * FROM users WHERE email=@email AND password = @password";
    var command = new SqlCommand(this.SqlQuery, connection);
    command.Parameters.AddWithValue("@email", email);
    command.Parameters.AddWithValue("@password", password);
    var reader = command.ExecuteReader();

    ...
    reader.Close();
    connection.Close();
}
```

Çözüm 3 – SQL Sorugusunu Gerçekleştiren Bir Ara Katman Kullanmak

- .NET'te Entity Framework Core paketi kullanılmaktadır
- Uygulama içerisinde veri obje ile temsil edilir
- EF Core bu objeler ile ilişkisel veritabanının eşleştirme yapar
- Önceki ornekteki soruyu EF Core ile yapacak olsaydık şuna benzer bir kod kullanacaktık:

```
db.Users.Where(user => user.email == email && user.password == password)
```

EF Core'da da FromSqlRaw Kullanılırsa Kod Enjeksiyonu Mümkün Hale Gelebilir

```
var users = db.Users.FromSqlRaw('SELECT * FROM users WHERE  
email='a@a.com' AND password='123456').ToList();
```

Bu durumda da parameter kullanılarak daha güvenli hale getirilebilir:

```
var users = db.Users.FromSqlRaw(  
    'SELECT * FROM users WHERE email={0} AND password={1}',  
    email,  
    password)  
.ToList();
```

String Interpolation Kullanımı

```
var users = db.Users.FromSqlInterpolated(  
    $"SELECT * FROM users WHERE email={email} AND password={password}")  
.ToList();
```

EF Core yerine neden direkt SQL sorgusu kullanılır?

- Bazı durumlarda yaşanan performans kayıpları nedeni ile direkt SQL sorgusu kullanımı tercih edilir.
- EF Core'un daha hafif bir alternatifi olan Dapper'da kimi firmalar tarafından tercih edilmektedir.

XML External Entities - XXE

XPathDocument XML verileri yüklememede kullanılan bir sınıfıdır.

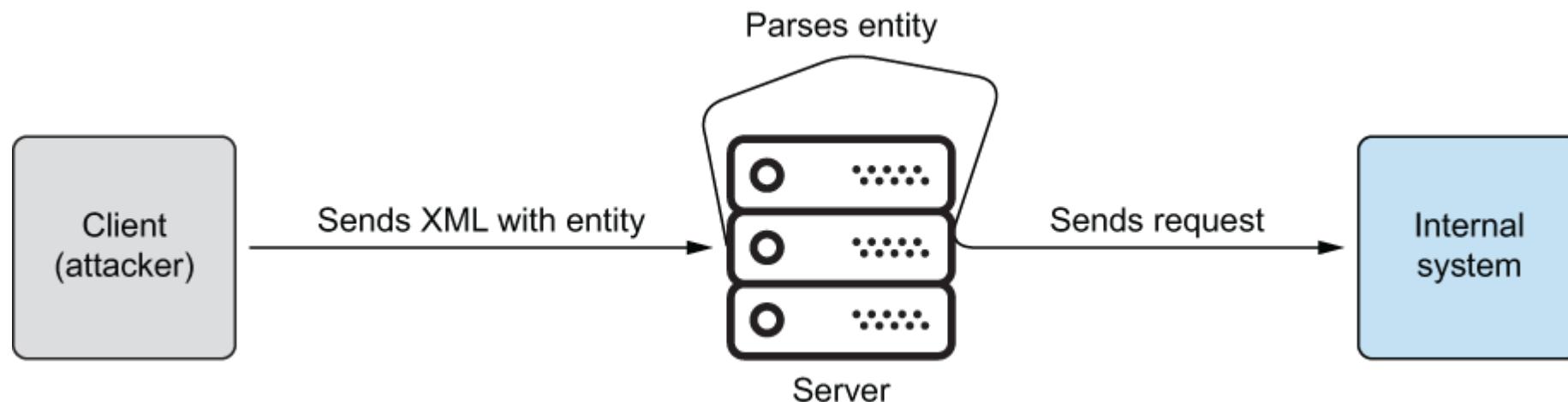
```
[HttpPost]
public ActionResult XmlEndpoint(string xml)
{
    var document = new System.Xml.XPath.XPathDocument(xml);
    var navigator = document.CreateNavigator();
    var output = navigator.InnerXml.ToString();
    return View(output);
}
```

etc/passwd klasöründeki tüm içeriği ekrana raporlayan kod:

```
<?xml version='1.0' ?>
<!DOCTYPE attack [
  <!ELEMENT attack ANY >
  <!ENTITY xxe SYSTEM 'file:///etc/passwd' >]
>
<attack>&xxe;</attack>
```

Server Side Request Forgery

```
<!ENTITY xxe SYSTEM  
'https://mainserver/actions/shutdown'>
```



XXE Çözüm

- .NET yardımcı sınıflarını kullanmak
 - XmlDocument
 - XmlNodeReader
 - XmlReader
 - XmlTextReader
 - XPathNavigator

Uygulamada Yer Alan Hassas Bilgilerin Korunması

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Örnek Vaka

- 2020 yılında SolarWinds isimli bir şirketin bir uygulamasında yer alan açık saldırganlar tarafından kullanılmıştır. Bu açığın kullanılması esnasında aşamalardan birtanesi de malicious bir yazılım paketinin indirilmesi idi. Güncellemelerin indirildiği FTP serverin şifresinin “solarwinds123” olduğu tespit edildiğinden bu saldırısı gerçekleştirilebildi.
- Yapılan araştırmada bu şifrenin programın kaynak kodları ile birlikte GitHub'a yüklediği tespit edildi.
- Şirket CEO'su hatanın sorumlusunun stajyer olduğunu beyan etti.
- We're not saying this is how SolarWinds was backdoored, but its FTP password 'leaked on GitHub in plaintext' • The Register

Örnek Vaka

- 2014 AWS müşterilerini uygulama anahtarlarının GitHub'a yüklenen kaynak kodlar ile birlikte yüklenmemesi konusunda dikkatli olması konusunda uyardı. [AWS urges developers to scrub GitHub of secret keys - Security - Software – iTnews](#)
- Basit bir araştırma ile bile binlerce anahtarın GitHub'a yüklediği tespit edildi.

Örnek Vaka

- 2018 yılında Giovanni Collazo isimli araştırmacı open etcd serverlarında 750mb'luk hassas bilgi elde edebildi. Bu bilgilerin içerisinde şifreler ve başka hassas bilgileri de mevcuttu. [The Security Footgun in etcd – Giovanni Collazo \(gcollazo.com\)](#)

Örnek Vaka

- 2019 yılında yürütülen bir araştırmada 100.000 adedin üzerinde repository'nin API token ve kriptografik anahtar içeriği tespit edildi.
[Over 100,000 GitHub repos have leaked API or cryptographic keys | ZDNET](#)
- 2020 yılında yürütülen bir deneyde GitHub'a yüklenen bir repositoryde AWS erişim bilgilerinin gönderilmesi halinde yükleme sonrası 1dk içerisinde saldırının başladığı ve 4. dakikada ise saldırganın amacına ulaştığı tespit edildi.
[It takes hackers 1 minute to find and abuse credentials exposed on GitHub - Comparitech](#)

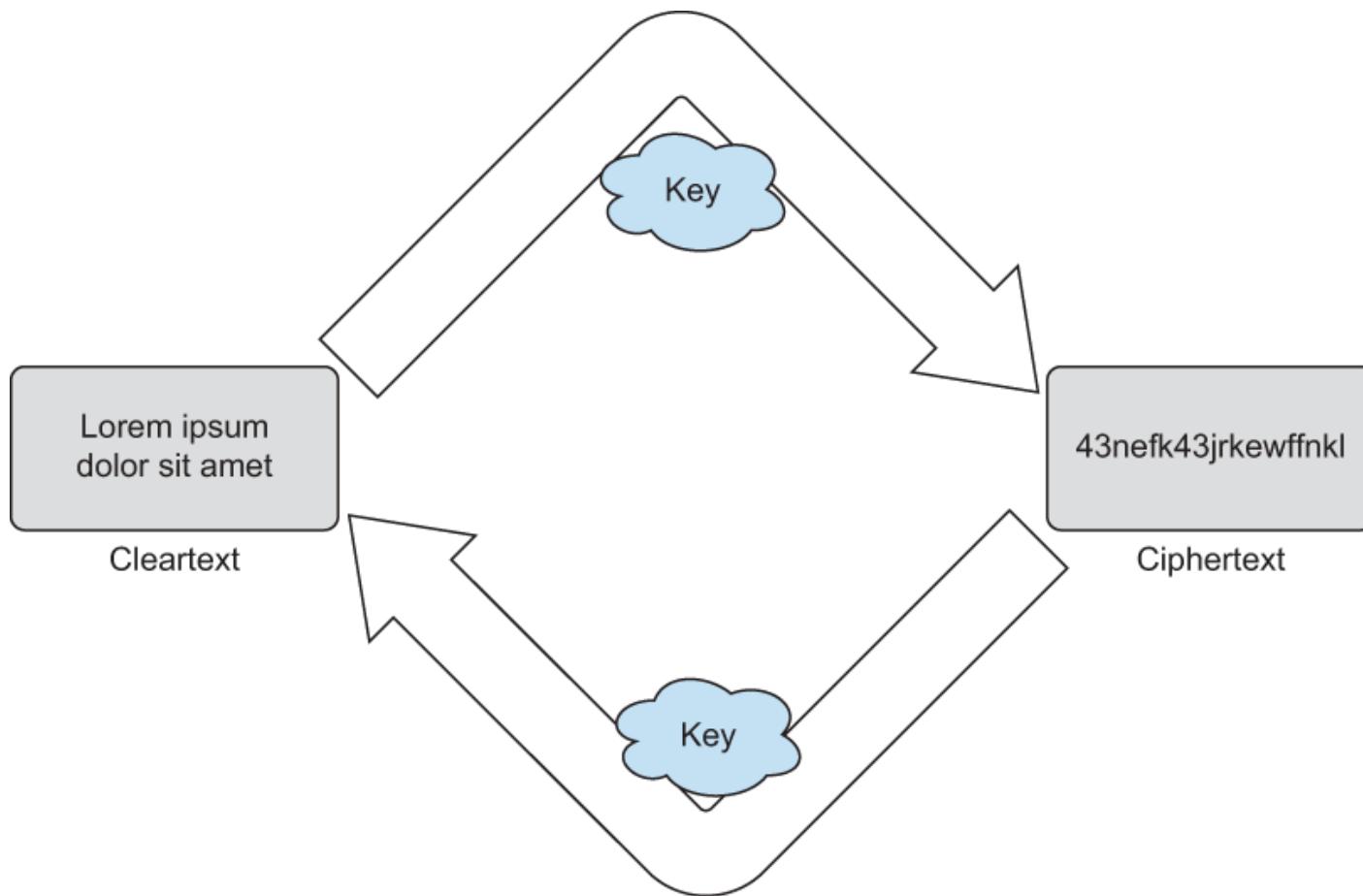
Çözümler

- Kaynak kodların hassas bilgileri içerip içermediği konusunda tarayan muhtelif uygulamalar mevcuttur.
 - [About secret scanning - GitHub Docs](#)
 - [GitHub - trufflesecurity/trufflehog: Find and verify credentials](#)
 - [GitHub - awslabs/git-secrets: Prevents you from committing secrets and credentials into git repositories](#)
 - [GitHub - auth0/repo-supervisor: Scan your code for security misconfiguration, search for passwords and secrets. :mag:](#)
- Genel Kural: Gizli olması gereken bilgileri repositorylere yükleme.

Şifreleme

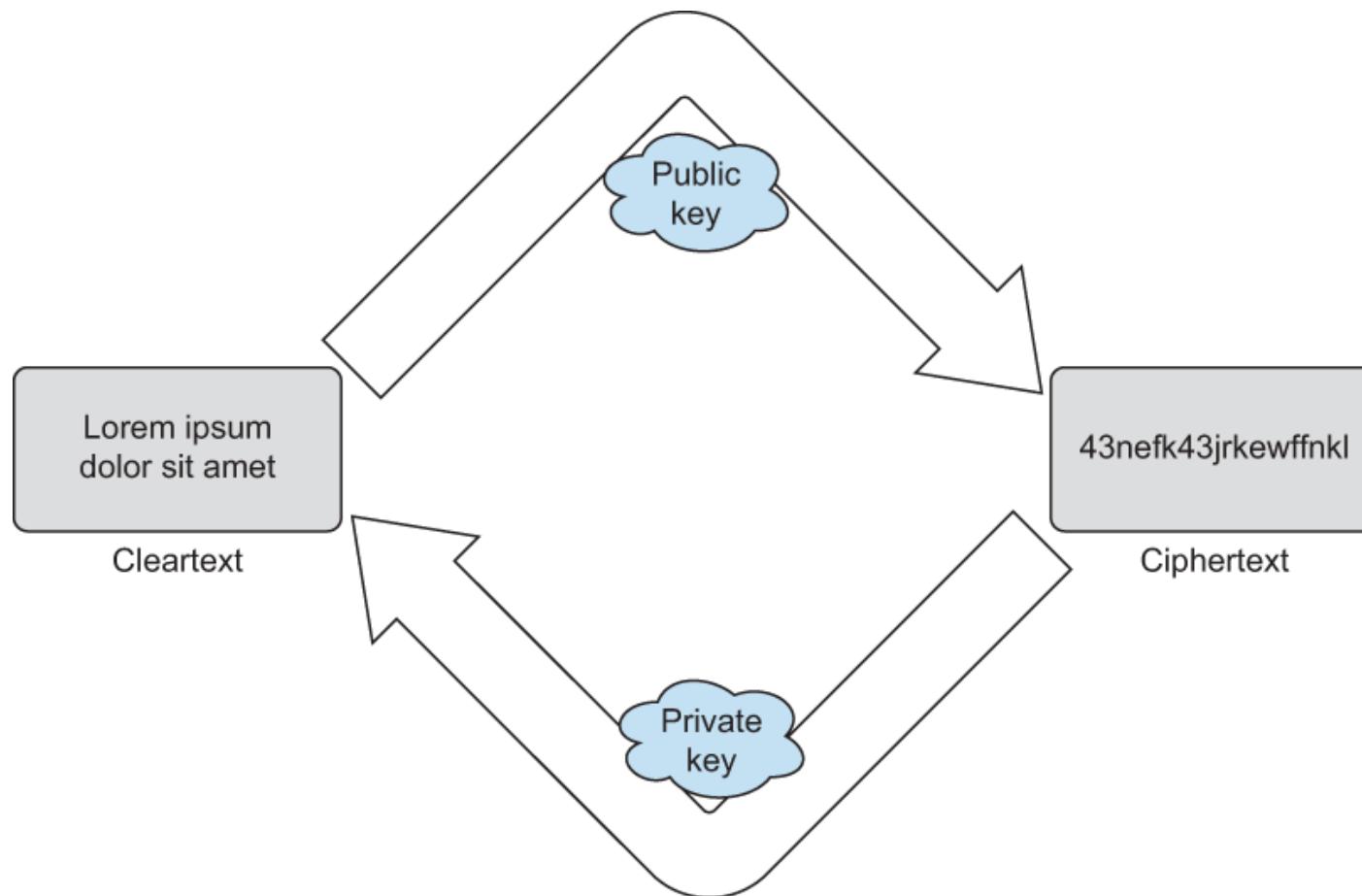
- Asimetrik şifreleme (public key)
- Simetrik şifreleme

Simetrik Şifreleme



- AES
- Blowfish – Twofish
- DES

Asimetrik Şifreleme



Kullanıldığı alanlar

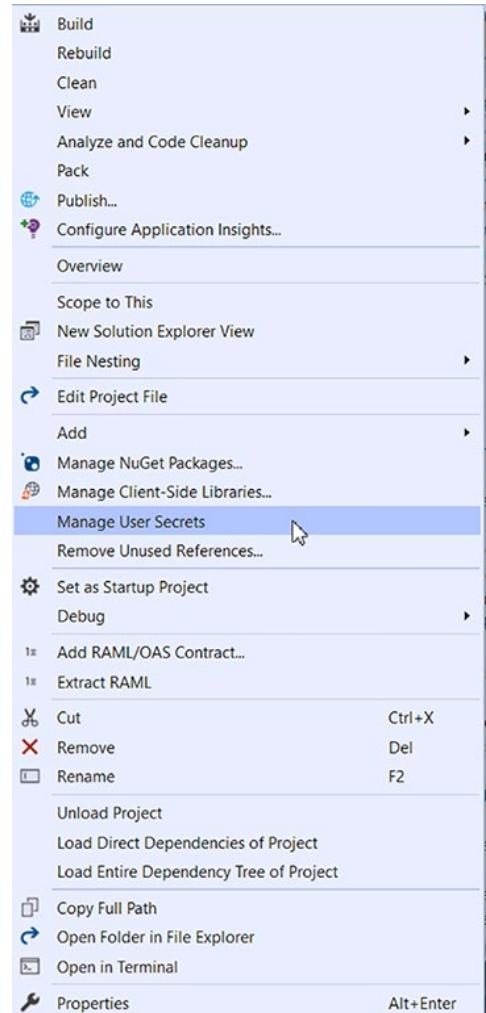
- HTTPS
- TLS – SSL
- PGP
- SSH

Hashing vs Şifreleme

- Şifrelemede şifrelenen metin original haline geri dönürtülebilir.
- Hashing işleminde geri dönürtmek mümkün değildir.

.NET Secret Manager

- Visual Studio içerisinde veya konsoldan active edilmesi gerekmektedir.



.NET Secret Manager

- Aktive edilince iki işlem yapmaktadır. Öncelikle proje dosyası olan .csproj dosyasına aşağıdakine benzer bir satır eklemektedir.

```
<Project Sdk='Microsoft.NET.Sdk.Web'>
```

```
  <PropertyGroup>
```

```
    ...
```

```
      <UserSecretsId>18b55eaa-cb11-4793-bd55-  
      e0cdd4e86063</UserSecretsId>
```

```
    ...
```

.NET Secret Manager

- Aynı zamanda bilgisayarda Microsoft/UserSecrets isimli klasörün altında aynı GUID isimli bir klasör oluşturulur. Microsoft/UserSecrets /<GUID> - globally unique identifier
- Microsoft/UserSecrets/d5d8bcfd-381f-4c93-966f-fd6245011a54
- Bu klasör secrets.json isimli bir dosya içerir. Programda kullanılan şifreler, tokenler vb. bu dosyaya kaydedilir.
- Bu dosya şifreli değildir.

.NET Secret Manager

- Tüm şifre ve tokenlar bu dosyaya kaydedilir.

```
{  
  'Shop': {  
    'ApiToken': 'abc123def456ghi789',  
    'ConnectionString': 'Server=(localdb)\\Shop;Integrated  
    Security=true'  
  }  
}
```

.NET Secret Manager

```
@page
@using Microsoft.Extensions.Configuration;
@inject IConfiguration Configuration

<div class='text-center'>
    <h1 class='display-4'>Secret Manager</h1>
    <form method='post' action="">
        <div class='mt-5 mb-5'>
            <p class='lead'>API Token: @Configuration['Shop:ApiToken']</p>
        </div>
    </form>
</div>
```

.NET Secret Manager

Harici bir klasörde saklanan bu gizli bilgileri program herhangi bir repositorye yüklenirken yüklenmez.

Fakat .NET projelerinde saklanması gereken tek gizli bilgi bu değildir.

Appsettings.json

- .NET projelerindeki ana ayarların saklandığı dosyadır.

```
{  
  'Logging': {  
    'LogLevel': {  
      'Default': 'Information',  
      'Microsoft': 'Warning',  
      'Microsoft.Hosting.Lifetime': 'Information'  
    }  
  },  
  'AllowedHosts': '*'  
}
```

Launchsettings.json

```
{  
  'iisSettings': {  
    'windowsAuthentication': false,  
    'anonymousAuthentication': true,  
    'iisExpress': {  
      'applicationUrl': 'http://localhost:40000',  
      'sslPort': 40001  
    }  
  },  
  'profiles': {  
    'IIS Express': {  
      'commandName': 'IISExpress',  
      'launchBrowser': true,  
      'environmentVariables': {  
        'ASPNETCORE_ENVIRONMENT': 'Development',  
        'ASPNETCORE_HOSTINGSTARTUPASSEMBLIES': 'Microsoft.AspNetCore.Mvc.Razor.RuntimeCompilation'  
      }  
    }  
  }  
}
```

Launchsettings.json

```
},  
    'AspNetCoreSecurity.RazorSamples': {  
        'commandName': 'Project',  
        'dotnetRunMessages': 'true',  
        'launchBrowser': true,  
        'applicationUrl': 'https://localhost:5001;http://localhost:5000',  
        'environmentVariables': {  
            'ASPNETCORE_ENVIRONMENT': 'Development',  
            'ASPNETCORE_HOSTINGSTARTUPASSEMBLIES': 'Microsoft.AspNetCore.Mvc.Razor.RuntimeCompilation'  
        }  
    }  
}
```

Gizli Bilgilerin Bulutta Saklanması

- Tüm bulut servis sağlayıcıları programlara ait gizli bilgilerin saklanması için servisler sunmaktadır.
- Genelde asimetrik şifreleme kullanılmaktadır.

Azure – Key Vault

The screenshot shows the Microsoft Azure portal interface for creating a new Key Vault. The left sidebar contains a navigation menu with various service icons and links. The main content area is titled 'Create a key vault' and includes tabs for 'Basics', 'Access configuration', 'Networking', 'Tags', and 'Review + create'. The 'Basics' tab is selected. A descriptive paragraph explains that Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. It highlights the benefits of centralizing secret storage, reducing the risk of leaks, and using Hardware Security Modules (HSMs) for secure storage. The 'Project details' section asks to select a subscription and resource group. A dropdown menu for 'Subscription' shows 'Azure for Students' is selected. Below it, a dropdown for 'Resource group' has 'Create new' highlighted. The 'Instance details' section requires entering a 'Key vault name' and selecting a 'Region'. The 'Key vault name' field is empty, and the 'Region' dropdown shows 'East US'. At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Key vaults >

Create a key vault

Basics Access configuration Networking Tags Review + create

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Azure for Students

Resource group *

Create new

Instance details

Key vault name * ⓘ

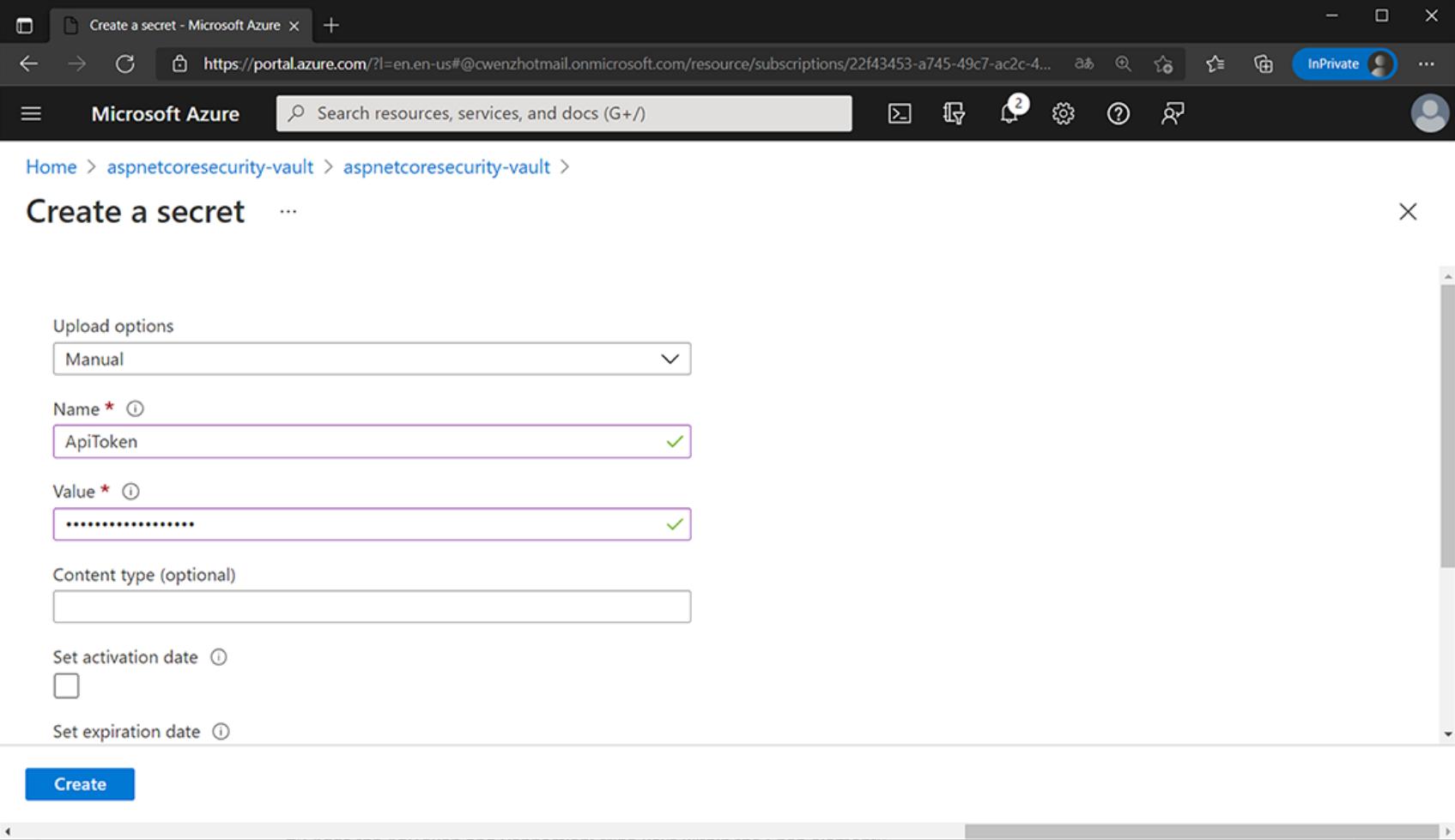
Enter the name

Region *

East US

Previous Next Review + create

Azure – Key Vault



The screenshot shows the Microsoft Azure portal interface for creating a new secret. The URL in the browser is <https://portal.azure.com/?l=en-en-us#@cwenzh@hotmail.onmicrosoft.com/resource/subscriptions/22f43453-a745-49c7-ac2c-4...>. The page title is "Create a secret - Microsoft Azure". The main content area is titled "Create a secret" and contains the following fields:

- Upload options:** A dropdown menu set to "Manual".
- Name ***: A text input field containing "ApiToken", which has a green checkmark indicating it is valid.
- Value ***: A text input field containing "*****", which also has a green checkmark.
- Content type (optional)**: An empty text input field.
- Set activation date**: A checkbox that is unchecked.
- Set expiration date**: A checkbox that is unchecked.

At the bottom left is a blue "Create" button.

Azure – Key Vault

- Nuget paket yöneticisinden aşağıdaki iki paketin programa eklenmesi gerekmektedir.
 - Azure.Identity
 - Microsoft.Azure.AppConfiguration.AspNetCore

Azure – Key Vault

```
using Azure.Identity;
var builder = WebApplication.CreateBuilder(args);
builder.Configuration.AddAzureAppConfiguration(options =>
{
    options.Connect('...')
        .ConfigureKeyVault(vault =>
    {
        vault.SetCredential(
            new DefaultAzureCredential());
    });
});
```

AWS – Secrets Manager

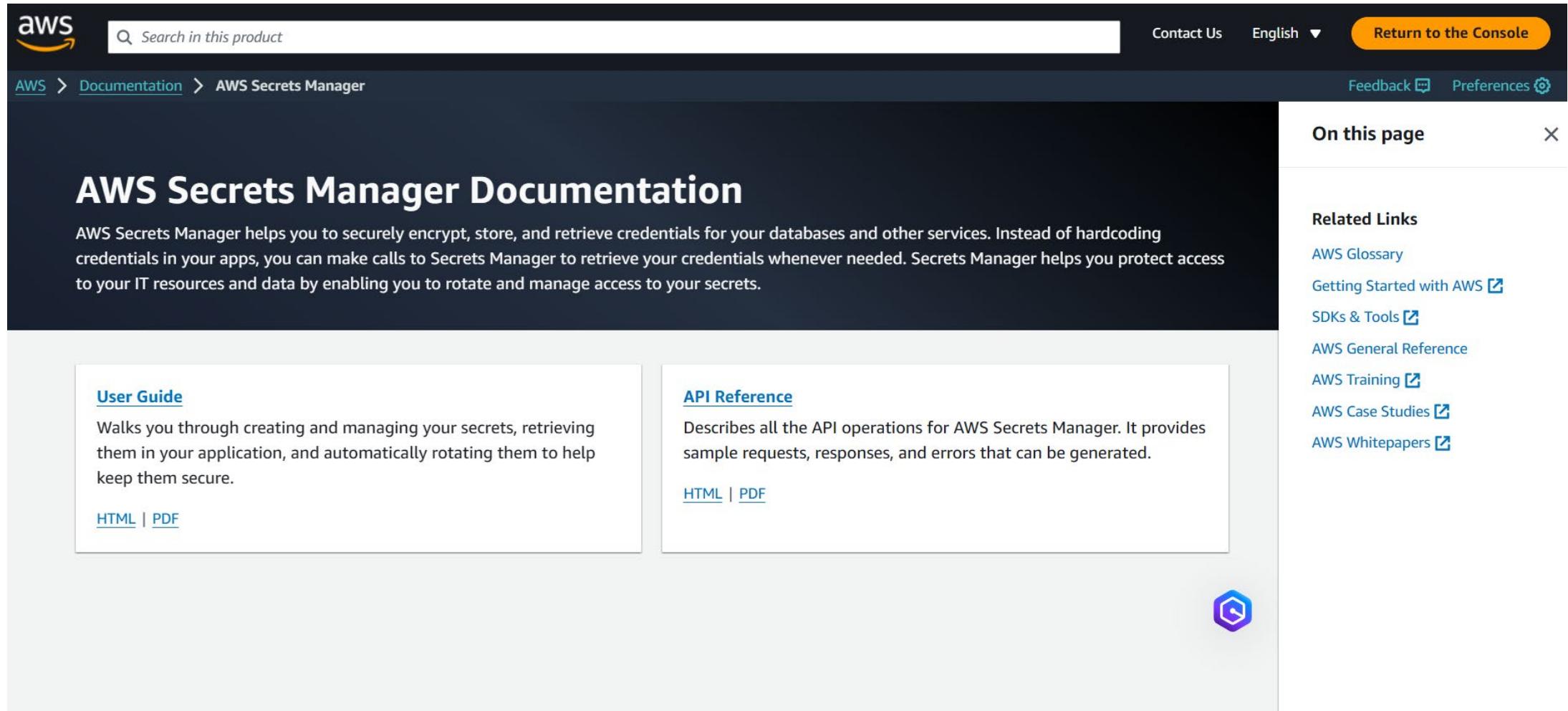
The screenshot shows the AWS Secrets Manager interface for creating a new secret. The left sidebar lists steps: Step 1 (Secret type), Step 2 (Name and description), Step 3 (Configure rotation), and Step 4 (Review). The main area is titled "Store a new secret" and "Select secret type". It displays five options for "Credentials for" various databases and an option for "Other type of secrets (e.g. API key)", which is selected and highlighted with a blue border. Below this, a table allows specifying key/value pairs. The first row shows "Shop:ApiKey" with value "abc123def456ghi789". A "+ Add row" button is visible at the bottom of the table. The browser address bar shows the URL <https://us-east-2.console.aws.amazon.com/secretsmanager/home?region=us-east-2#/newSecret?step=selectSecret>.

AWS – Secrets Manager

The screenshot shows the AWS Developer Center homepage with the following details:

- Header:** AWS logo, re:Invent link, navigation menu (Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More), Contact Us, Support, English, My Account, Sign In, and a yellow "Create an AWS Account" button.
- Sub-navigation:** Developer Center, Learning, Programming Languages, Events, Tools, Community, and More Resources.
- Section:** Tool to Build on AWS, titled "AWS SDK for .NET".
- Text:** "Develop and deploy applications with the AWS SDK for .NET. The SDK makes it easy to call AWS services using idiomatic .NET APIs."
- Buttons:** "Get started with AWS SDK for .NET" (orange) and "Install from NuGet" (blue).
- Image:** A large, abstract geometric graphic composed of white and light gray 3D-like shapes.
- Section:** "How it Works" (bolded).
- Text:** "AWS SDK for .NET simplifies use of AWS Services by providing a set of libraries that are consistent and familiar for .NET developers. All AWS SDKs provide support for API lifecycle consideration such credential management, retries, data marshaling, and serialization. AWS SDK for .NET also supports for higher level abstractions such as the S3 Transfer Utility, Cognito Identity Provider, and AWS DyanamoDB Session State provider. Visit [aws/dotnet Github](#) for .NET tools and libraries on AWS."

AWS – Secrets Manager



The screenshot shows the AWS Secrets Manager Documentation page. At the top, there's a navigation bar with the AWS logo, a search bar, and links for Contact Us, English (dropdown), and Return to the Console. Below the navigation bar, the breadcrumb trail shows AWS > Documentation > AWS Secrets Manager. On the right side, there's a sidebar titled "On this page" with an "X" button. The main content area features a large title "AWS Secrets Manager Documentation" and a brief description of the service's purpose: "AWS Secrets Manager helps you to securely encrypt, store, and retrieve credentials for your databases and other services. Instead of hardcoding credentials in your apps, you can make calls to Secrets Manager to retrieve your credentials whenever needed. Secrets Manager helps you protect access to your IT resources and data by enabling you to rotate and manage access to your secrets." Below the main title, there are two sections: "User Guide" and "API Reference". The "User Guide" section describes how it walks you through creating and managing secrets, and includes links to "HTML" and "PDF" versions. The "API Reference" section describes all API operations and includes a link to "HTML | PDF". A small purple hexagonal icon is located at the bottom right of the main content area.

AWS Secrets Manager Documentation

AWS Secrets Manager helps you to securely encrypt, store, and retrieve credentials for your databases and other services. Instead of hardcoding credentials in your apps, you can make calls to Secrets Manager to retrieve your credentials whenever needed. Secrets Manager helps you protect access to your IT resources and data by enabling you to rotate and manage access to your secrets.

User Guide

Walks you through creating and managing your secrets, retrieving them in your application, and automatically rotating them to help keep them secure.

[HTML](#) | [PDF](#)

API Reference

Describes all the API operations for AWS Secrets Manager. It provides sample requests, responses, and errors that can be generated.

[HTML](#) | [PDF](#)



AWS – Secrets Manager

The screenshot shows a GitHub repository page for 'Kralizek / AWSSecretsManagerConfigurationExtensions' (Public). The repository has 14 issues, 2 pull requests, and 50 commits. It includes sections for Code, Issues, Pull requests, Actions, Security, and Insights. The repository description states: "This repository contains a provider for Microsoft.Extensions.Configuration that retrieves secrets stored in AWS Secrets Manager." It features tags for aws, aspnet-core, dotnet-standard, and aws-secrets-manager. The repository has 204 stars, 6 watchers, 41 forks, and was last updated on Nov 9, 2022.

Kralizek / AWSSecretsManagerConfigurationExtensions Public

Code Issues 14 Pull requests 2 Actions Security Insights

master 2 branches 12 tags

Kralizek Added button to Buymeacoffee ... b3a8a7a on Feb 1 50 commits

.config CI/CD maintenance (#31) 3 years ago

.devcontainer Added devcontainer last year

build Improved CI setup (#13) 4 years ago

samples Refresh samples to .NET 6.0 (#79) last year

src/Kralizek.Extensions.Configuration.... Update AWS SDK package to 3.7.2.0 last year

tests/Tests.Extensions.Configuration.A... Replace Newtonsoft with System.Text.Json (#71) last year

tools CI/CD maintenance (#31) 3 years ago

.gitignore Improved CI setup (#13) 4 years ago

GitVersion.yml Improved CI setup (#13) 4 years ago

LICENSE Initial commit 5 years ago

README.md Added button to Buymeacoffee last year

SecretsManager.sln Allow customization of GetSecretValueRequest (#77) last year

About

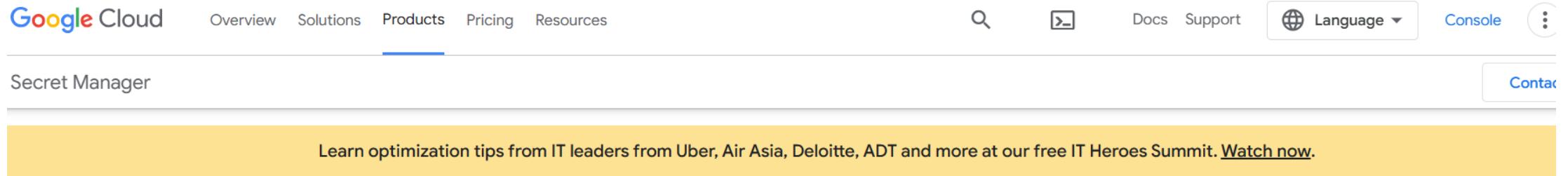
This repository contains a provider for Microsoft.Extensions.Configuration that retrieves secrets stored in AWS Secrets Manager.

aws aspnet-core dotnet-standard
aws-secrets-manager

Readme MIT license Activity 204 stars 6 watching 41 forks Report repository

Releases 12 v1.7.0 Latest on Nov 9, 2022

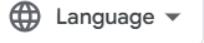
Google Cloud – Secret Manager



The screenshot shows the Google Cloud homepage with the 'Secret Manager' section highlighted. The top navigation bar includes links for Overview, Solutions, Products (underlined), Pricing, and Resources. The main content area features a yellow banner with promotional text about the IT Heroes Summit. Below the banner, the 'Secret Manager' title is displayed in large, bold letters, followed by a descriptive paragraph and a 'Go to console' button.

Google Cloud

Overview Solutions Products **Products** Pricing Resources

Docs Support  Language  Console 

Secret Manager 

Learn optimization tips from IT leaders from Uber, Air Asia, Deloitte, ADT and more at our free IT Heroes Summit. [Watch now.](#)

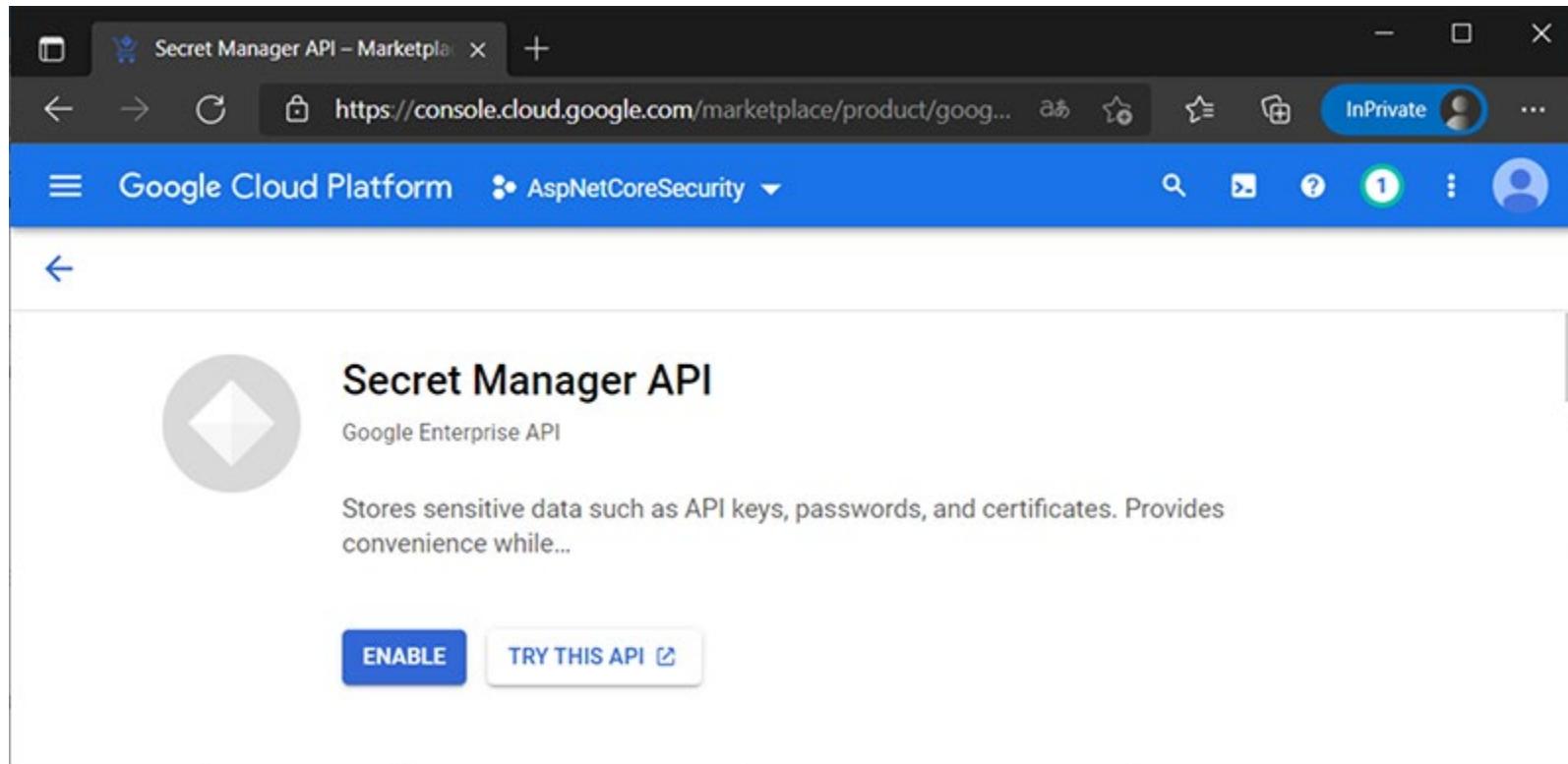
Secret Manager

Store API keys, passwords, certificates, and other sensitive data. New customers get \$300 in free credits to spend on Secret Manager. All customers get six secret versions for analyzing and storing sensitive data.

[Go to console](#)

[View documentation](#) for this product.

Google Cloud – Secret Manager



Google Cloud – Secret Manager

The screenshot shows the 'Create secret' page in the Google Cloud Platform console. The left sidebar is titled 'Security' and lists various Google Cloud security services. The main form is titled 'Create secret' and contains fields for 'Secret details' and 'Secret value'. The 'Secret details' section includes a 'Name' field with 'ApiToken' entered, a note about uniqueness, and a link to 'Learn more'. The 'Secret value' section includes a 'Upload file' input field with a 'BROWSE' button, a note about maximum size (64 KiB), and a 'Secret value' text area containing the string 'abc123def456ghi789'. At the bottom are 'CREATE SECRET' and 'CANCEL' buttons.

Create secret – Security – AspNetCoreSecurity

https://console.cloud.google.com/security/secret-manager/create?folder=&orga...

InPrivate

Google Cloud Platform

AspNetCoreSecurity

Search products and reso...

SECRET

CREATE SECRET CANCEL

Security

Security Command Center

reCAPTCHA Enterprise

BeyondCorp Enterprise

Identity-Aware Proxy

Access Context Manager

VPC Service Controls

Binary Authorization

Data Loss Prevention

Key Management

Certificate Authority Servi...

Create secret

Secret details

This will create a secret with the secret value in the first version. [Learn more](#)

Name
ApiToken

The name should be identifiable and unique within this project.

Secret value

Input your secret value or import it directly from a file.

Upload file BROWSE

Maximum size: 64 KiB

Secret value
abc123def456ghi789

Google Cloud – Secret Manager

- Google.Cloud.SecretManager.V1

Google Cloud – Secret Manager

The screenshot shows the Google Cloud Secret Manager documentation page. At the top, there's a navigation bar with links for Overview, Solutions, Products, Pricing, Resources, a search bar, and options for Docs, Support, English (language dropdown), Console, and more. Below the navigation is a secondary navigation bar with links for Secret Manager (which is active), Overview, Guides, Resources, Reference, Samples, Contact Us (button), and Help (button). The main content area has a sidebar on the left with links for Secret Manager documentation, Overview, Training and tutorials, and Videos. The main content area features a summary of Secret Manager's purpose and a "Learn more" link. It also contains four main sections: Guides, Resources, and References, each with a list of related topics.

Secret Manager documentation

Secret Manager stores API keys, passwords, certificates, and other sensitive data. It provides convenience while improving security. [Learn more.](#)

Overview

Training and tutorials

Videos

Guides

- Create a secret with Secret Manager
- Add a secret version
- Choose a replication policy
- Encryption of secrets
- Access control with IAM
- Set an expiration date for a secret
- Create and manage rotation schedules

Resources

- Pricing
- Quotas and limits
- Release notes
- Locations
- Service Level Agreement (SLA)

References

- Client libraries
- REST API
- RPC API
- `gcloud secrets command`

Veri Koruma API'sini Kullanma

- Sakladığımız verileri kendimiz şifreleme yoluna gidebiliriz.
- Simetrik veya asimetrik olarak şifreleme yapabiliriz.
- Fakat bu durumda bir sızıntı olmadığından emin olunmalıdır.
- Diğer bir seçenek ise böylesi bir işlem için hazır kütüphanelerin kullanımıdır.

Veri Koruma API’I Kullanma

VeriSizintisi Veri Koruma

Veri Koruma API

Data

abcdef

Şifrele

Veri Koruma API’I Kullanma

[VeriSizintisi](#) [Veri Koruma](#)

Veri Koruma

Şifreli veri: CfDJ8Llw3h-0PdGmxRpMszTrs1H4jmk8SeNyPF3qJCpjM3eUdsDqdDZN5KtrKdGFD-k9WgMsijTNvl7jo67Xaj06GKBbchKsLB1TFiX5KoT0Rg1oFtwldRxRvGNMFd3vl4og

Şifresi çözülmüş veri: abcdef

[Şifrelemeye Geri Dön](#)

Veri Koruma API'I Kullanma

VeriSizintisi Veri Koruma

Veri Koruma

Şifreli veri: CfDJ8Llws3h-0PdGmxRpMszTrs1H4jm8SeNyPF3qJCpjM3eUdsDqdDZN5KtrKdGFD-k9WgMsijTNvl7jo67XaJ06GKBbchKsLB1TFiX5KoT0Rg1oFtwldRxRvGNMFd3vl4og

Şifresi çözülmüş veri: abcdef

[Şifrelemeye Geri Dön](#)

Veri Koruma API'sını Kullanma

- Eğer uygulama birden fazla serverda parallel şekilde çalışıyor ise uygulamalar arasında şifreleme anahtarı paylaşımı gereklidir.
- Kendi serverlarınızda

```
builder.Services.AddDataProtection()  
.PersistKeysToFileSystem(new DirectoryInfo('...'));
```

Veri Koruma API'sını Kullanma

- Azure Blob'da

```
builder.Services.AddDataProtection()
```

```
    .PersistKeysToAzureBlobStorage('Connection string', 'Container',  
    'Blob')
```

```
    .ProtectKeysWithAzureKeyVault('Key URI', new  
    DefaultAzureCredential());
```

WebAssembly

- Uygulama tamamen kullanıcının browserında çalışır
- Cookieler, local storage ve session storage erişimi mümkündür
- Şifreleme ve şifre çözme işlemi tamamen kullanıcı tarafından gerçekleşecektir ise herhangi bir şifre çözme anında XSS yöntemi ile gizli kalması gereken verilere yetkisiz erişim sağlanabilir
- Fakat eğer şifreli veri kullanıcıda saklanırken, şifre çözme işlemi ise server tarafında gerçekleşir ise verilerin korunması sağlanabilir.
- .NET Blazor'da veri şifreli şekilde kullanıcıda saklanırken server ile kurulan WebSocket bağlantısı ile bu verinin şifresi server tarafında çözülür.
- ProtectedSessionStorage, ProtectedLocalStorage kütüphaneleri kullanılır.

Şifre Güvenliği

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Örnek Vaka

- 2013 yılında Adobe firmasının serverlarına bir sızıntı gerçekleşti. Muhtelif kaynaklar firmanın yazılım ürünlerinin kaynak kodlarına ve kullanıcı bilgilerine erişildiğini raporladı.
 - [Adobe To Announce Source Code, Customer Data Breach – Krebs on Security](#)
 - [Over 150 million breached records from Adobe hack have surfaced online - The Verge](#)
- 150 milyon kullanıcı ait bilginin sızdığını tespit edildi. Bu bilgiler arasında kullanıcı şifreleri de bulunmaktaydı.
- Kullanıcı şifreleri düz yazı olarak saklanmadığından deşifre etmesi kolay olmamakla birlikte imkansız da değildi.

Kullanıcı Şifrelerinin Deşifre Edilmesi

- Adobe veritabanında şifreler simetrik şifreleme yöntemi ile saklanmakta, yani şifrelemek ve şifreyi çözmek için aynı anahtar kullanılmaktaydı.
- Eğer iki kullanıcının şifresi aynı ise veritabanında saklanan şifrelenmiş veri de aynı olmaktadır.
- Diğer bir yöntem ise en sık karşılaşılan şifrelenmiş verileri listeleyip sık kullanılan şifreler listesi ile eşleştirmeye çalışmaktadır.

Örnek Vaka

- Adobe vakasında işin kötü tarafı kullanıcılarda şifrelerini unuturlarsa hatırlamalarını sağlayacak bir ifade girilmesi istenmişti.
- Bu bilgi veritabanına şifrelenmeden kaydedilmiştir.

Wayback Machine (archive.org)

Top 100 Adobe Passwords with Count			
We do not (yet) have the keys Adobe used to encrypt the passwords of 130,324,429 users affected by their most recent breach. However, thanks to Adobe choosing symmetric key encryption over hashing, selecting ECB mode, and using the same key for every password, combined with a large number of known plaintexts and the generosity of users who flat-out gave us their password in their password hint, this is not preventing us from presenting you with this list of the top 100 passwords selected by Adobe users.			
#	Count	Ciphertext	Plaintext
1.	1911938	EQ7fIpT7i/Q=	123456
2.	446162	j9p+HwtWNT86aMjgZFLzYg==	123456789
3.	345834	L8qbAD3jl3jioxG6CatHBw==	password
4.	211659	BB4e6X+b2xLioxG6CatHBw==	adobe123
5.	201580	j9p+HwtWNT/ioxG6CatHBw==	12345678
6.	130832	5djv7ZCI2ws=	qwerty
7.	124253	dQi0asWPYvQ=	1234567
8.	113884	7lqYzKVeq8I=	111111
9.	83411	PMDtbP0Lzxu03SwrFUvYGA==	photoshop
10.	82694	e6MPX05G6a8=	123123
11.	76910	j9p+HwtWNT8/HeZN+3oiCQ==	1234567890
12.	76186	di0+ie23vAA=	000000
13.	70791	kCcUSCmonEA=	abc123
14.	61453	ukxzEcXU6Pw=	1234
15.	56744	5wEAInH22i4=	adobe1
16.	54651	WqflwJFYW3+PszVFzo1Ggg==	macromedia
17.	48850	hjAYsdUA4+k=	azerty
18.	47142	rpkvF+oZzQvioxG6CatHBw==	iloveyou
19.	44281	xz6PIeGzr6g=	aaaaaa
20.	43670	Ypsmk6AXQTk=	654321
21.	43497	4V+mGczxDEA=	12345
22.	37407	yp2KLBDbiQxs=	666666
23.	35325	2dJY5hIJ4FHioxG6CatHBw==	sunshine
24.	34963	1McuJ/7v9nE=	123321
25.	33452	yxzNXPisFno=	letmein
26.	32549	dcG824yq9Bw=	monkey
27.	31554	dA8D80YD55E=	asdfgh

Aynı Şifre İçin Girilmiş Hatırlatıcılar

- let me in
- knock
- I'm in
- open sesame
- let who in?
- the usual
- Standard

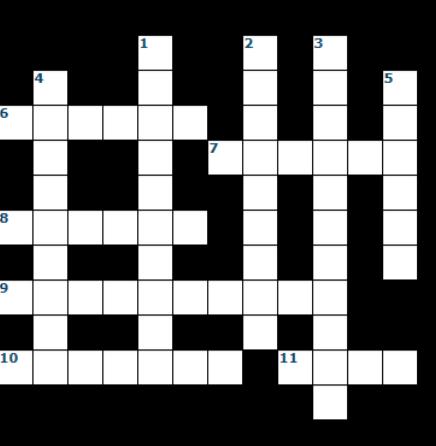
Hepsinin hatırlatmaya çalıştığı şifre: **letmein**

Adobe Crossword (zed0.co.uk)

Adobe Crossword [Crossword](#) [FAQ](#) [Github](#) [Tweet](#)

A crossword based on the Adobe password leak.
Inspired by [xkcd #1286: Encryptic](#)

Password popularity: [1-100](#) [101-200](#) [201-300](#) [301-400](#) [401-500](#) [501-600](#) [601-700](#) [701-800](#) [801-900](#) [901-1000](#)



Reveal Check Hide

Across

- ▼ 6: zk8NJJgAOqc4= dog; cat; pet; dark; dogs name; Dog; my dog; black dog; dog name; dog's name; darkness; black cat; sonic; black; kitty; horse; Cat; pets name; sombra; puppy; cats name; old dog; shade; first dog; pet name; doggy; hedgehog; cat's name; bike; my cat; nickname; Pet; me; light; favorite pet; usual; sha; doggie; pet's name; first pet; animal; sh; shad; s; car; first cat; Dog's name; chien; favorite dog; ombre
- ▼ 7: WIMTLimQ5b4=
- ▼ 8: FTeB5SkrOZM=
- ▼ 9: WqflwJFYW3+PszVFZo1Ggg=
- ▼ 10: yxzNxPlsFno=
- ▼ 11: L3uQHNDf6Mw=

Down

- ▼ 1: 2aZl4Ouarwm52NYYI936YQ== adobe; adobex2; adobe2; adobe twice; twice; adobetwice; adobe2x; site; ??????; name; software; 2x; company; 2xadobe; program; adobe x 2; program; adobe x2; ???; ad; adobe*2; ???; Adobe; double; namename; 2adobe; ?????; x2; a; name twice; photoshop; company name; adobe adobe; adobe?; ado; aa; company twice; 2; marca; website; none; adobe 2x; product; company name twice; adobeX2; this; logiciel; ??, ???????, what is this
- ▼ 2: L8qbAD3jl3jSPm/keox4fA==
- ▼ 3: 7Z6uMyq9bpxe1EB7HijrBQ==
- ▼ 4: vp6d18mfGL+5n2auThm2+Q==
- ▼ 5: dA8D8OYD55E=

NordPass Sık Kullanılan Şifreler Listesi

- [Top 200 Most Common Passwords List | NordPass](#)

RANK	PASSWORD	TIME TO CRACK IT	COUNT
1	123456	< 1 Second	4,524,867
2	admin	< 1 Second	4,008,850
3	12345678	< 1 Second	1,371,152
4	123456789	< 1 Second	1,213,047
5	1234	< 1 Second	969,811
6	12345	< 1 Second	728,414
7	password	< 1 Second	710,321
8	123	< 1 Second	528,086

171	qwerty1234	< 1 Second	20,415
172	123abc	< 1 Second	20,318
173	theworldinyourhand	Centuries	20,176
174	123456a@	2 Seconds	20,046
175	Aa102030	10 Seconds	19,731
176	987654	< 1 Second	19,728

Entropi

- 123456 – 20 bitlik entropiye sahiptir. Çözülmesi için 2^{20} deneme yapılmalıdır. 1 milyonun biraz üzerinde.
- 6'sar adet küçük harf, büyük harf ve rakamdan oluşan bir şifrenin entropisi 100 bittir. Deneme sayısı 33 haneli bir rakamdır.
- NIST tarafından önerilen en az 8 karakterlik şifre girilmesidir. Bu şifrenin 64 karaktere kadar da uzayabileceği belirtilmiştir. Şifrelerde özel karakter kullanım önerisi ise kaldırılmıştır.

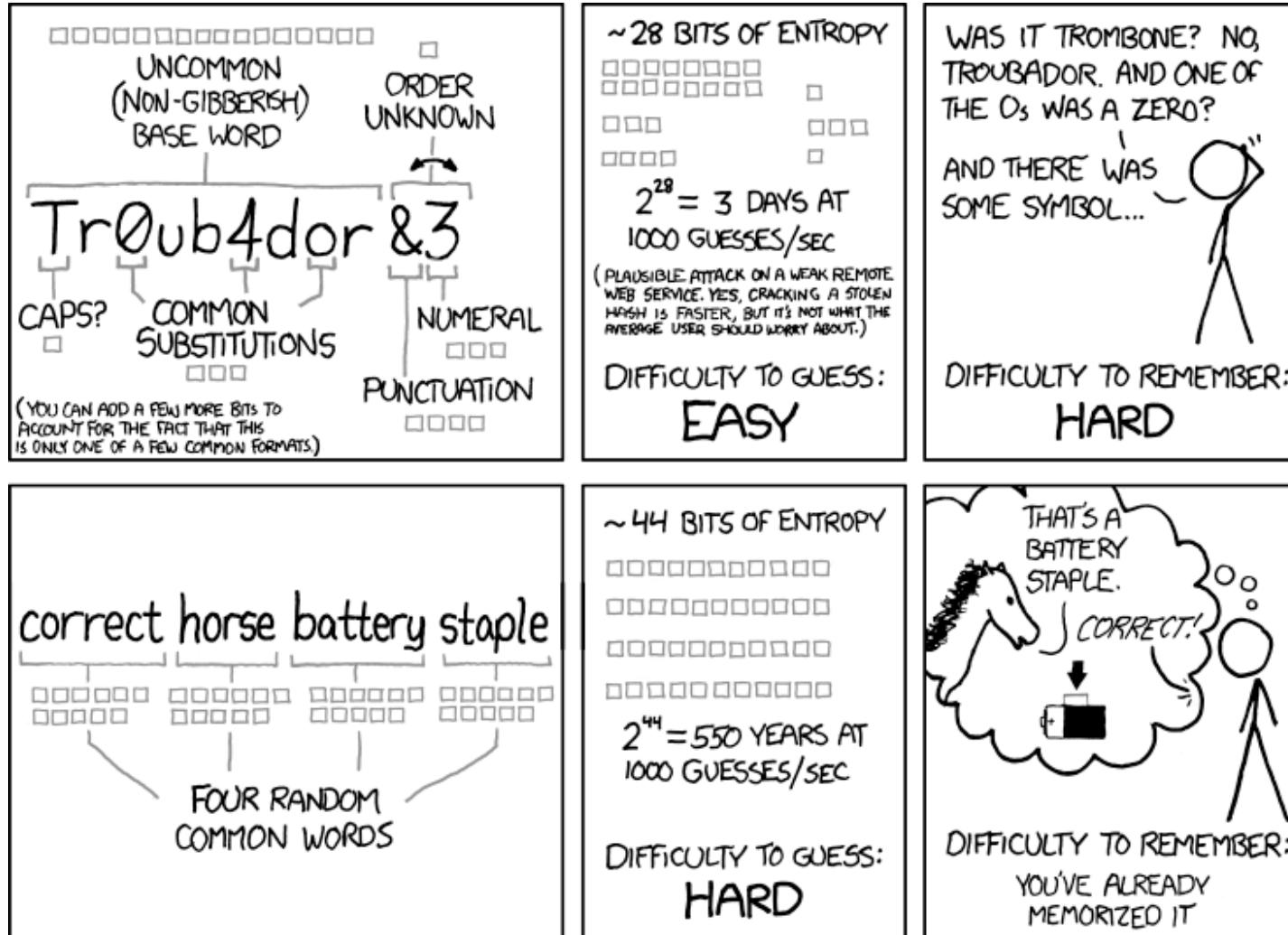
NIST Special Publication 800-63B

5.1.1.1 Memorized Secret Authenticators

Memorized secrets SHALL be at least 8 characters in length if chosen by the subscriber. Memorized secrets chosen randomly by the CSP or verifier SHALL be at least 6 characters in length and MAY be entirely numeric. If the CSP or verifier disallows a chosen memorized secret based on its appearance on a blacklist of compromised values, the subscriber SHALL be required to choose a different memorized secret. No other complexity requirements for memorized secrets SHOULD be imposed. A rationale for this is presented in [Appendix A Strength of Memorized Secrets](#).

5.1.1.2 Memorized Secret Verifiers

Verifiers SHALL require subscriber-chosen memorized secrets to be at least 8 characters in length. Verifiers SHOULD permit subscriber-chosen memorized secrets at least 64 characters in length. All printing ASCII [\[RFC 20\]](#) characters as well as the space character SHOULD be



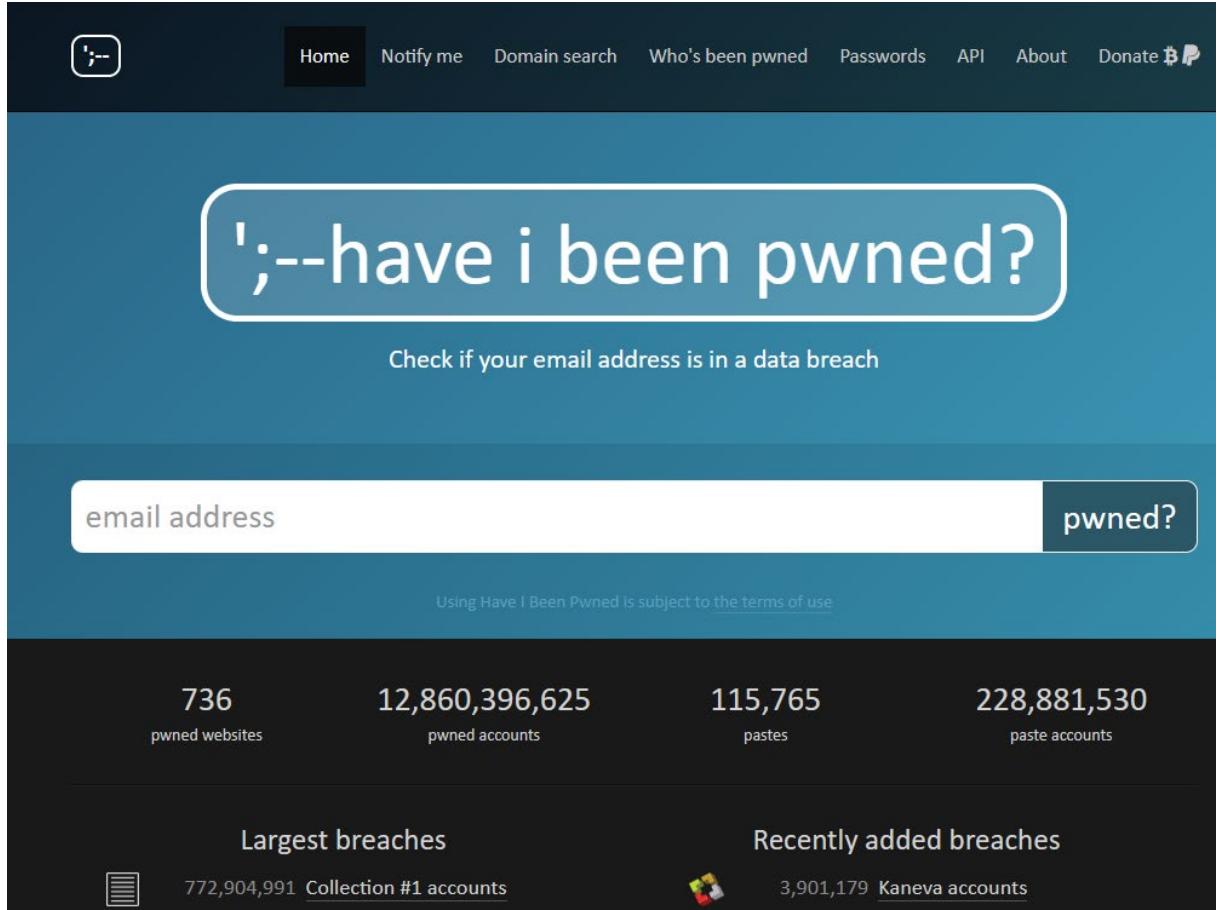
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

- correct horse battery staple
- 44 bit entropiye sahiptir

Şifre Kasaları

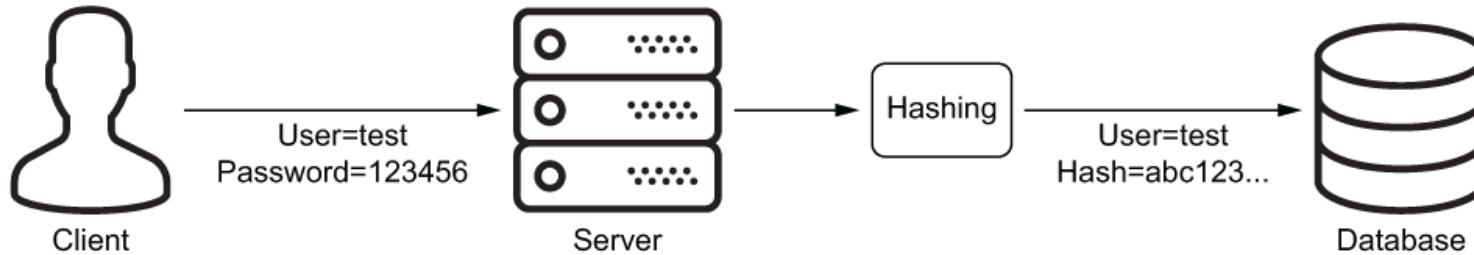
- KeePass Password Safe

Have I Been Pwned: Check if your email has been compromised in a data breach

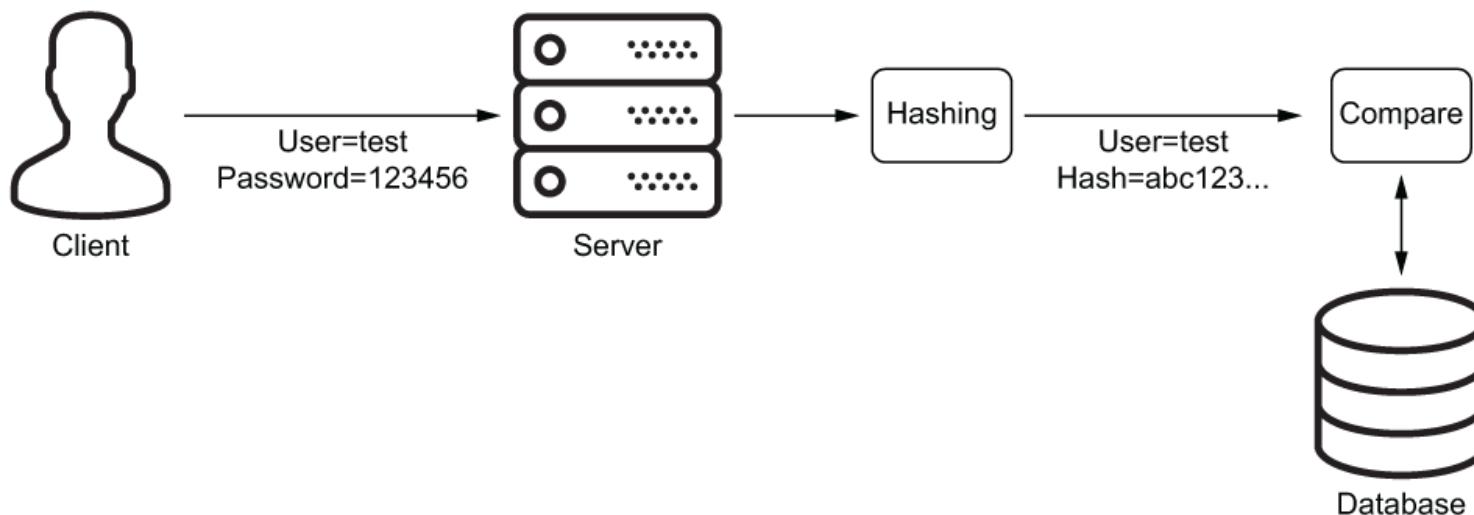


Hashing Yöntemi

Registration



Login



Hashing

- Bir hashing algoritmasından beklenenler:
 - Ortaya çıkan çıktıının tahmin yöntemi ile tekrar oluşturulabilmesi imkansız olmalıdır
 - Orijinal bilginin tekrar oluşturulması imkansız olmalıdır
 - Orijinal bilgideki ufak değişimler çıktıda büyük değişime yol açmalıdır
 - Çıktı sabit uzunlukta olmalıdır

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10
K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20
U	V	W	X	Y	Z				
21	22	23	24	25	26				

Hashing Algoritmaları

- MD5 – 128bit
- SHA/SHA1 – 160bit
- SHA2:
 - SHA224 – 224bit
 - SHA256 – 256bit
 - SHA384 – 384bit
 - SHA512 – 512 bit

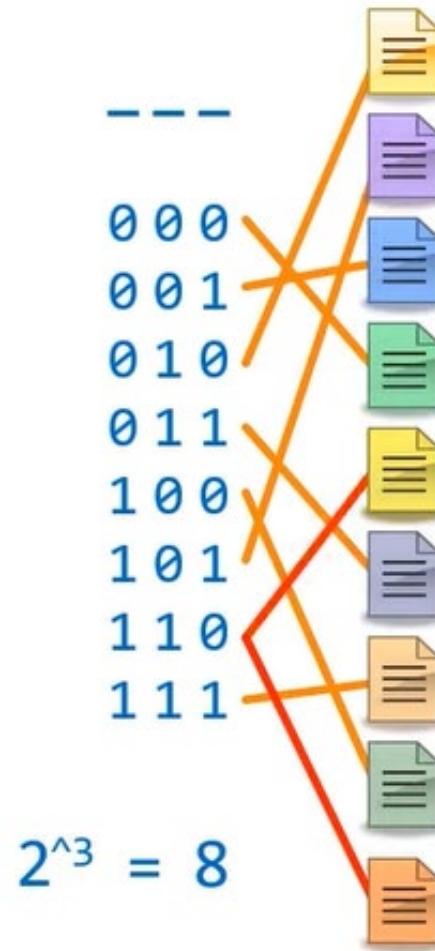
MD5

- 16 byte uzunlukta Hash oluşturur
- Günümüzde tercih edilmemektedir
- Collision saldırısı yöntemi ile çözülebilmektedir
- [Single-block collision attack for MD5 \(marc-stevens.nl\)](#)

Collision Sorunu

- İki farklı girdinin aynı çıktıyı vermesi durumudur
- Tamamen önlenmesi mümkün değildir
- Önlenememesinin sebebi hashing algoritmalarının aynı uzunlukta çıktı verme zorunluluğudur
- Algoritmanın çıkışının uzun olması sorunu bir nebze önleyebilmektedir.

Collision Sorunu – 3 bitlik çıktı veren hashing algoritması örneği



MD5

- correct horse battery staple
- MD5: 9cc2ae8a1ba7a93da39b46fc1019c481
- Sıklıkla kullanılan birçok şifrenin MD5 yöntemi ile hashlenmiş hali mevcuttur.

MD5 Center

MD5 conversion and reverse lookup

MD5 hash for « correct horse battery staple »

The MD5 hash of [correct horse battery staple](#) is [9cc2ae8a1ba7a93da39b46fc1019c481](#)

You can attempt to reverse the MD5 hash which was just generated, to reverse it into the originally provided string:

[Reverse a MD5 hash](#)

x

Reverse

MD5

- Tek tek deneme yöntemi ile de hashlenmiş şifre çözülebilir fakat bu çok uzun zaman alacaktır.
- Bu süreyi de kısaltmak için “rainbow tables” denilen veritabanları mevcuttur. Bu tablolarda daha önce hesaplanmış hashler tutulmaktadır. Yani daha önce hesaplanmış şifreler için tekrar hesaplama yapılmamakta böylece tüm olasılıkların deneme süresi kısaltılmaktadır.
- Salting adı verilen yöntem ile hashlenmiş şifrelerin çözülmesi zorlaştırılmaktadır.

MD5

- Salting örneği:
 - Şifre: correct horse battery staple**Aydin**
 - MD5 Hash: fc350193e8cc5e84507ab002a39c0b24
- Fakat bu örnekte de zayıflık var. Salting için kullandığımız kelime çok kısa, tahmin edilebilir ve sabit. Bunun yerine herbir çalışmada rastgele değerler üreten bir salting mekanizması daha güvenilir olacaktır. Ayrıca aynı şifre girilse bile salting kodu farklı olacağından hash kodu da farklı olacaktır.

MD5

- Tüm bunlara ek olarak hem şifreye hem de salting'e peppering adı verilen sabit bir işlem daha uygulanarak ek bir güvenlik katmanı daha oluşturulabilir. Böylece uygulamanın bir kopyası saldırının eline geçmediği müddetçe şifreyi çözmesi çok zorlaşacaktır.

Diğer Alternatifler – Password Hash Algorithms

- PBKDF2 – NIST tarafından önerilen algoritmadır. FIBS-140 standardını sağlamaktadır.
- Argon2 – 2013 yılından 2015 yılına kadar şifre hashleme yarışmasında birinci seçildi. <https://www.password-hashing.net/>
- Scrypt – Alternatiflerine göre hashleme için çok daha fazla kaynak kullanmaktadır. Deneme yanılma yöntemi ile çözme işleme karşı alternatiflerine göre çok daha güvenlidir. Salt ve şifrenin yanında cpu, hafıza ve parallelization için cost parameter alır. Çıktı uzunluğu bilgisini de arguman olarak alır. Özünde PBKDF2 kullanır fakat
- Bcrypt – 1999 yılında geliştirilmiştir fakat halen güvenli kabul edilmektedir. Fakat bazı limitler sözkonusudur. En fazla 72byte uzunlukta veriyi hashlemektedir.

PBKDF2 – Password Based Key Derivative Functions

- Bazı kaynaklarda RFC 2898 olarak geçmektedir
- <https://www.ietf.org/rfc/rfc2898.txt>
- .NET kütüphanelerinde Rfc2898DeriveBytes olarak geçmektedir
- Bu sınıf 3 argüman beklemektedir:
 - Hashlenecek şifre
 - Byte cinsinden salt uzunluğu
 - Yineleme sayısı. Algoritma kendisini ne kadar çok yineler ise çözmesi o kadar zor hash kodu oluşturulur ama oluşturulma süresi uzar.
 - Algoritmayı

PBKDF2

PBKDF2 [Home](#) [Privacy](#)

PBKDF2 Hashing

Kullanıcı Adı

aydin

Şifre

.....

Hash

CxJDtJ2sV0TA2zj4cMKZfGl8I=

Salt

MOKa8xw2ISZ5H2kk+IQeSadDSzi8dXlI7Fc+dmRPt0I=

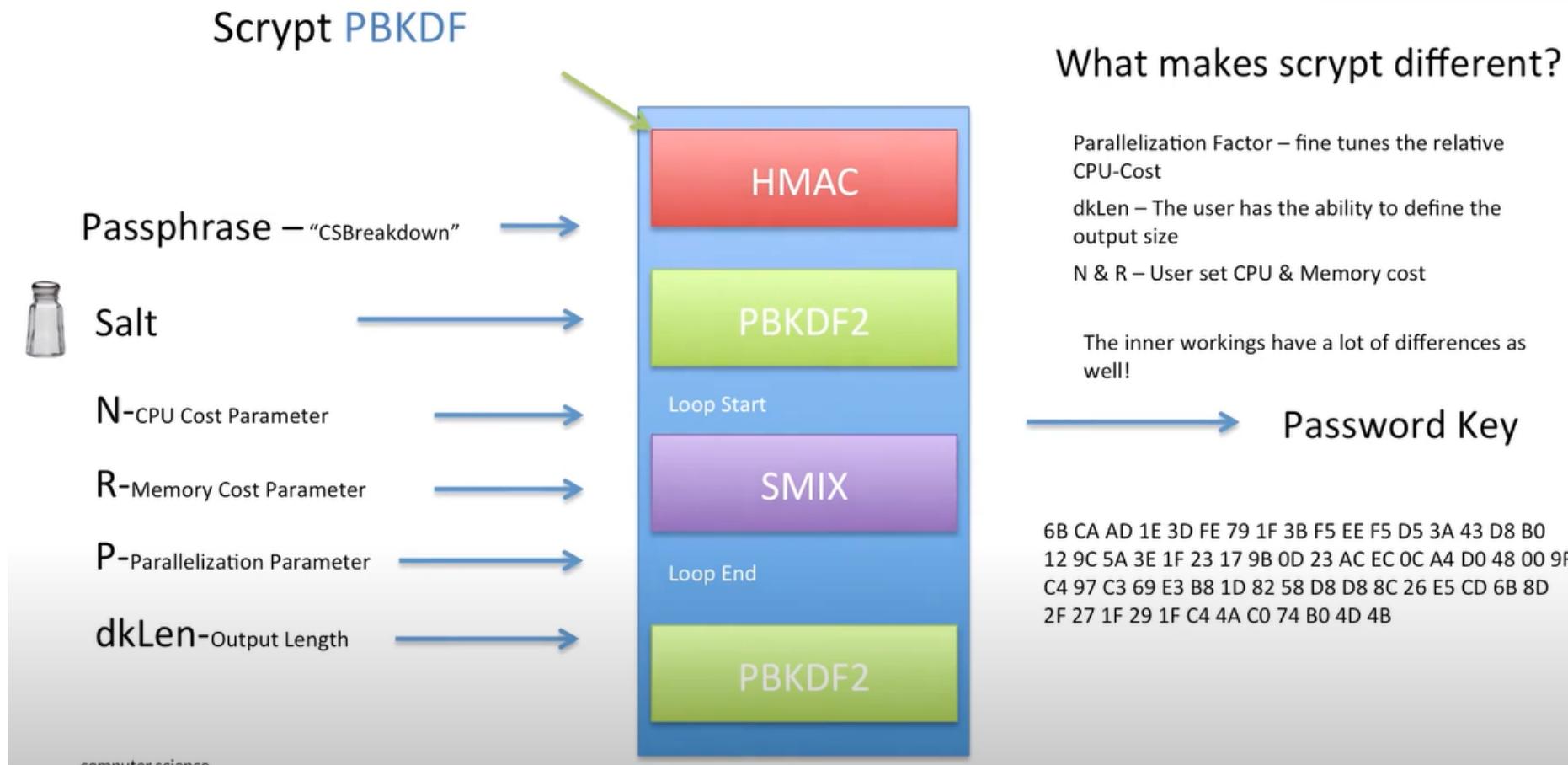
Kayıt

Giriş

PBKDF2

- [Password Storage - OWASP Cheat Sheet Series](#)
- 210.000 kez yineleme önerisi ilgili dokumanda PBKDF2'de SHA512 algoritması kullanılması halinde önerilen yineleme sayısıdır.

Scrypt



Bcrypt

- Yavaş çalışmak üzere tasarlanmıştır
- Şifre girdisi 72byte ile sınırlıdır
- İş maliyeti de girilmelidir
- Şifre: abc123xyz iş maliyeti: 12 için çıktı:

```
$2a$12$R9h/cIPz0gi.URNNX3kh20PST9/PgBkqquzi.Ss7KIUg02t0jWMUW  
\\ / \\ \_____  
Alg Cost      Salt          Hash
```

HTTP Headerlar

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Genel Yazılım Sorunları

- [Home | CVE](#)
- [CVE security vulnerability database. Security vulnerabilities, exploits, references and more \(cvedetails.com\)](#)

CVE About Partner Information Program Organization Downloads Resources & Support Report/Request

Enter CVE ID (CVE-YYYY-NNNN) **Find** Site Search

Find CVE Records by keyword.

i Welcome to the new CVE Beta website! CVE Records have a new and enhanced [format](#). View records in the new format using the CVE ID lookup above or download them on the [Downloads](#) page. [CVE List keyword search](#) will be temporarily hosted on the legacy [cve.mitre.org](#) website until the [transition](#) is complete. ▾

CVE® Program Mission

Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

Currently, there are **220,038** CVE Records accessible via [Download](#) or [Search](#).

The CVE Program partners with community members worldwide to grow CVE content and expand its usage. Click below to learn more about the role of [CVE Numbering Authorities \(CNAs\)](#) and [Roots](#).

[Learn More](#) [Become a Partner](#)

News

- [OpenSSF Publishes Guide to Becoming a CNA as an Open Source Project](#)
- [New CVE Board Member from GitHub Security Lab](#)
- [ARC Informatique Added as CVE Numbering Authority \(CNA\)](#)
- [HiddenLayer Added as CVE Numbering Authority \(CNA\)](#)

[Access](#)

[Learn](#)

[Report/Request](#)

Microsoft Internet Information Services : Security vulnerabilities, CVEs (cvedetails.com)

Documentation

Enter a CVE id, product, vendor, vulnerability type...

Search Log in

CVEDetails.com powered by SecurityScorecard

Vulnerabilities

- By Date
- By Type
- Known Exploited
- Assigners
- CVSS Scores
- EPSS Scores
- Search

Vulnerable Software

- Vendors
- Products
- Version Search

Vulnerability Intel.

- Newsfeed
- Open Source Vulns
- Emerging CVEs
- Feeds
- Exploits
- Advisories
- Code Repositories
- Code Changes

Microsoft » Internet Information Services : Security Vulnerabilities

Published in: 2023 January February March April May June July August September October November December

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9 In CISA KEV Catalog

Sort Results By: Publish Date ↑↓ Update Date ↑↓ CVE Number ↑↓ CVE Number ↑↓ CVSS Score ↑↓ EPSS Score ↑↓

91 vulnerabilities found

> 1 2 3 4 Copy

CVE ID	Description	Max CVSS	Published	Updated	EPSS
CVE-2014-4078	The IP Security feature in Microsoft Internet Information Services (IIS) 8.0 and 8.5 does not properly process wildcard allow and deny rules for domains within the "IP Address and Domain Restrictions" list, which makes it easier for remote attackers to bypass an intended rule set via an HTTP request, aka "IIS Security Feature Bypass Vulnerability."	5.1	2014-11-11	2018-10-12	0.82%
CVE-2011-5279	CRLF injection vulnerability in the CGI implementation in Microsoft Internet Information Services (IIS) 4.x and 5.x on Windows NT and Windows 2000 allows remote attackers to modify arbitrary uppercase environment variables via a \n (newline) character in an HTTP header.	5.0	2014-04-23	2020-11-23	1.13%
CVE-2010-3972	▲ Public exploit exists	10.0			

Microsoft Asp.net Core : Security vulnerabilities, CVEs (cvedetails.com)

Documentation

Enter a CVE id, product, vendor, vulnerability type...

Search

Log in

CVEDetails.com
powered by SecurityScorecard

Vulnerabilities

- By Date
- By Type
- Known Exploited
- Assigners
- CVSS Scores
- EPSS Scores
- Search

Vulnerable Software

- Vendors
- Products
- Version Search

Vulnerability Intel.

- Newsfeed
- Open Source Vulns
- Emerging CVEs
- Feeds
- Exploits
- Advisories
- Code Repositories
- Code Changes

Microsoft » Asp.net Core : Security Vulnerabilities

Published in: 2023 January February March April May June July August September October November December

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9 In CISA KEV Catalog

Sort Results By: Publish Date ↗️ Update Date ↗️ CVE Number ↗️ CVE Number ↗️ CVSS Score ↗️ EPSS Score ↗️

31 vulnerabilities found

1 2

CVE-2023-44487 ⚠ Known Exploited Vulnerability

The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.

Max CVSS	7.5
Published	2023-10-10
Updated	2023-12-02
EPSS	60.16%
KEV Added	2023-10-10

CVE-2023-38180 ⚠ Known Exploited Vulnerability

.NET and Visual Studio Denial of Service Vulnerability

Max CVSS	7.5
Published	2023-08-08
Updated	2023-08-20
EPSS	1.05%
KEV Added	2023-08-09

Headers Preview Response Initiator Timing Cookies

▼ Response Headers [View source](#)

Cache-Control: no-cache, no-store
Content-Type: text/html; charset=utf-8
Date: Sun, 17 Oct 2021 09:00:45 GMT
Pragma: no-cache
Referrer-Policy: no-referrer
Server: Microsoft-IIS/10.0
Set-Cookie: .AspNetCore.Antiforgery.W1GJeJkD1Ak=CfDJ8ITL8J_jMs1LuJ4XLUH5FtUx26G0WOLACQfhL08DHmYNBVpxAcIEy4xDdDmWaOA09qR6yQe_52Vu32bxNa6oD3WluFufHHD8wqY1LQyzWVqsLhC5Kw_k7cIHIpMZ1DG0zaaHr4xxfD0WL0bIFRvgDpw; path=/AspNetCoreSecurity.RazorSamples; samesite=strict; httponly
Transfer-Encoding: chunked
X-Frame-Options: SAMEORIGIN
X-Powered-By: ASP.NET

Servera Ait Bilgileri Açıga Çıkartan Headerlar

- Server
- X-Powered-By

Kestrel Server Bilgilerini Gizleme

```
var builder = WebApplication.CreateBuilder(args);
```

```
builder.WebHost.UseKestrel(options =>
```

```
{
```

```
    options.AddServerHeader = false;
```

```
});
```

IIS Server Bilgilerini Gizleme

Web.config dosyası

```
<?xml version='1.0' encoding='utf-8'?>
<configuration>
    <system.webServer>
        <security>
            <b><requestFiltering removeServerHeader='true' /></b>
        </security>
    </system.webServer>
</configuration>
```

X-Powered-By Headerini Gizleme

- Bu header IIS server tarafından eklenmektedir. Uygulama içerisinde önlenmesi mümkün değildir.

Internet Information Services (IIS) Manager

DESKTOP-Q3R45US > Sites > Default Web Site >

File View Help

Connections

- DESKTOP-Q3R45US (DESKTOP)
 - Application Pools
- Sites
 - > Default Web Site

Actions

Add... Set Common Headers... Help

HTTP Response Headers

Use this feature to configure HTTP headers that are added to responses from the Web server.

Group by: No Grouping

Name	Value	Entry Type
X-Powered-By	ASP.NET	Inherited

X-Powered-By Headerini Kullanici Arayuzunden Gizleme

Features View Content View

Configuration: 'Default Web Site' web.config

X-Powered-By için web.config dosyası düzenleme

```
<?xml version='1.0' encoding='utf-8'?>
<configuration>
  <system.webServer>
    <security>
      <b><requestFiltering removeServerHeader='true' /></b>
    </security>
    <httpProtocol>
      <customHeaders>
        <b><remove name='X-Powered-By' /></b>
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

Response Headerlardan Bu İki Bilgi Kaldırıldı

X	Headers	Preview	Response	Initiator	Timing
▼ Response Headers					
content-encoding: gzip					
content-length: 773					
content-type: text/html; charset=utf-8					
date: Tue, 28 Sep 2021 18:54:52 GMT					
vary: Accept-Encoding					

Browser Güvenlik Headerları

- 2009 yılında Internet Explorer 8 ile birlikte güvenlik ile ilgili headerlar kullanılmaya başlandı.
- Örneğin X-XSS-Protection XSS saldırısını önleme amaçlı kullanılmaya başlandı. Günümüzde bu headerin kullanımı sonlanmıştır.

Referrer Policy

- Referer headeri ilgili öge yüklenirken ögenin yüklemesi talebinin nereden gittiğini göstermektedir.
- Bu header browser tarafından otomatik olarak iletilmektedir.
- Bu durum risk yaratmaktadır. İlgili verinin gizlenmesi gerekiyor olabilir.
- Hangi koşullar altında bu headerin gönderilmesi gereği referrer policy ile ayarlanabilir.
- [Referrer Policy \(w3.org\)](https://w3.org)
- 2017 yılında önerilmiştir. Halen “candidate” yani aday statüsündedir. Öte yandan tüm modern tarayıcılar desteklemektedir.

Referrer Policy

- Bu policy Referrer-Header headeri ile ayarlanmaktadır.
- **no-referrer** : referrer headeri iletilmez
- **no-referrer-when-downgrade** : sadece mevcut ve yönlendirilen url https protokolünü destekliyor ise iletilir
- **origin** : https://example.com/ formatında iletilir yani origin+/-
- **origin-when-cross-origin** : Farklı originden talep gidecek olursa https://example.com/ formatında iletilir yani origin+/-
- **same-origin** : sadece mevcut ve talebin iletiliği originler aynı ise iletilir
- **strict-origin** : https://example.com/ formatında iletilir yani origin+/-, diğer bir hususta sadece http formatında veya her iki tarafta https formatını destekliyor ise iletilir
- **strict-origin-when-cross-origin** : eğer same origin ise tüm URL, değil ise https://example.com/ formatında iletilir yani origin+/- ek olarak sadece http formatında veya her iki tarafta https formatını destekliyor ise iletilir
- **unsafe-url** : same origin olsun olmasın tüm URL iletilir

Referrer Policy

- Browserlarda uzun süre varsayılan olarak **unsafe-url** aktif durumdaydı. Dolayısı ile talebin geldiği siteye ait tüm detaylar elde edilebiliyordu.
- Daha sonra **no-referrer-when-downgrade** varsayılan olarak kullanılmaya başlandı.
- Chrome tabanlı browserlarda Kasım 2020'de **strict-origin-when-cross-origin** varsayılan olarak ayarlandı. Böylece talep farklı bir originden geliyor ise sadece origin bilgisi iletilecektir. Ek olarak HTTPS protokolinden HTTP'ye bir düşüş halinde gönderilmemektedir.
- Aynı kuralları Firefox Ağustos 2021'de varsayılan olarak uygulamaya başladı.
- Safari ise Intelligent Tracking Prevention (ITP) ile siteler arası iletilen taleplerde tüm URL'yi değil sadece origini göndermektedir.

Referrer Policy'i ASP.NET'te kullanmak için iki seçenek mevcuttur.

IIS kullanıyorsanız web.config dosyası:

```
<?xml version='1.0' encoding='utf-8'?>
<configuration>
  <system.webServer>
    <httpProtocol>
      <customHeaders>
        <b><add name='Referrer-Policy' value='no-referrer' /></b>
      </customHeaders>
    </httpProtocol>
  </system.webServer>
</configuration>
```

Referrer Policy'i ASP.NET'te kullanmak için iki seçenek mevcuttur.

Kestrel serverda:

Bu kod satırı tüm diğer tüm middleware'lerden önce eklenmelidir.

```
app.Use(async (context, next) =>
{
    context.Response.Headers.Add('Referrer-Policy', 'no-referrer');
    await next.Invoke();
});
```

Referrer Policy Sonuç

x	Headers	Preview	Response	Initiator	Timing	Cookies
▼ Response Headers						
content-encoding: gzip						
content-type: text/html; charset=utf-8						
date: Wed, 29 Sep 2021 08:55:43 GMT						
referrer-policy: no-referrer						
vary: Accept-Encoding						

Feature and Permissions Policy

- Javascript ve browserların sağladıkları API'lar ile yazılımcılara çok sayıda imkan sunulmuştur. Bu durum muhtelif tehtidlere de imkan vermektedir. Örneğin XSS saldırılarına.
- Feature&Permissions Policy ile browserda kullanılabilecek özelliklere ve API'lara sınırlama getirilebilmektedir.
- [Permissions Policy \(w3.org\)](https://w3.org) – 18 Aralık 2023'te güncellenmiştir
- Örneğin: Permissions-Policy: fullscreen=() – ilgili sayfa ve alt sayfaların tam ekran görüntülenmesini öner
- Permissions-Policy: fullscreen=(self 'https://example.com')

Diğer Bazi Yetkilere Örnekler

- Autoplay – ilgili medyanın otomatik şekilde oynatılması
- Camera – kamera erişimi
- Microphone – mikrofon erişimi
- ch-downlink – bant genişliği bilgisi
- display-capture – ekran resmi çekebilme
- Tüm liste için: [webappsec-permissions-policy/features.md at main · w3c/webappsec-permissions-policy · GitHub](https://github.com/w3c/webappsec-permissions-policy/blob/main/features.md)
- Diğer bir yöntem browser konsulunda şu komutu girmek:
`document.featurePolicy.allowedFeatures()`

Browserın İçeriği Tahmin Etmeye Çalışmasını Önleme

- Content-Type headeri browsera ne türde bir içeriğin iletiliği bilgisini iletmektedir. Yani MIME türünü.
- Bir html sayfası için bu bilgi şu şekildedir: Content-Type: text/html; charset=utf-8
- Bazı eski browserlar (bilhassa Internet Explorer) bu bilgiyi göz ardı edip içeriğin ne olduğunu kendisi tahmin etme yoluna gitmekteydi. Bu da muhtelif saldırılara zemin hazırlamaktaydı.
- [Konu ile ilgili akademik bir çalışma](#)
- X-Content-Type-Options: nosniff – bu header browserın tahmin etmeye çalışmasını kesin olarak Content-Type headerine uyma talimatı verir

Browserın İçeriği Tahmin Etmeye Çalışmasını Önleme

Uygulama içerisinde:

```
app.Use(async (context, next) =>  
{  
    context.Response.Headers.Add('X-Content-Type-Options', 'nosniff');  
    await next.Invoke();  
});
```

IIS serverda:

```
<add name='X-Content-Type-Options' value='nosniff' />
```

Cross-Origin Policies

- Farklı originlerden gelen taleplerin ne gibi sorunlara yol açabildiğini daha önce görmüştük.
- Ek olarak çeşitli header başlıkları da koruma sağlamaktadır.
- Cross-Origin-Embedder-Policy: unsafe-none – farklı orijinden içerikleri de yükler
- Cross-Origin-Embedder-Policy: require-corp – kısıtlamalar getirir
 - İlgili kaynak CORS uygulamış olmalı ve Access-Control-Allow-Origin headerinde da uygun talimatı iletmelidir
 - Kaynağı yüklerken crossorigin isimli HTML özelliği kullanılmalıdır
 - İlgili kaynak Cross-Origin-Resource-Policy headerini uygulayarak açık şekilde kaynağın yüklemesine müsaade etmelidir. Bu header üç opsyon alabilir: same-site, same-origin ve cross-origin

<https://securityheaders.com/>

- Sitenizi security headerlar konusunda tarar ve önerilerde bulunur.
- Bu taramanın sonuçlarının kesin olduğu yanılığısına düşülmemelidir. İlla her site her headeri uygulayacak diye bir kural yoktur. İhtiyaca göre header seçiliip kullanılmaktadır.
- AWS bile bu sitede C notu almaktadır

Scan your site now

Scan

Hide results Follow redirects

Security Report Summary



Site:	https://aws.amazon.com/
IP Address:	2600:9000:20aa:c600:1c:a813:8512:c241
Report Time:	20 Dec 2023 11:59:27 UTC
Headers:	✓ X-Frame-Options ✓ Strict-Transport-Security ✓ X-Content-Type-Options ✗ Content-Security-Policy ✗ Referrer-Policy ✗ Permissions-Policy

Advanced: Not bad... Maybe you should perform a deeper security analysis of your website and APIs:

Try Now

Hataların Ele Alınması

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Örnek Vaka

- 2018 yılında ZingBox isimli güvenlik firması tıbbi IoT cihazlardan alınan hata mesajlarının sırra olarak tutulması gereken aşağıdaki bilgileri ifşa ettiğini tespit etmiştir:
 - Veritabanı kullanıcı adları ve şifreleri
 - Serverdaki dosya pathleri
 - Kaynak kod dosya isimleri, programa ait detaylar (sınıf ve metod isimleri)
 - Kaynak kod satır numaraları
 - V.b.
- İlgili haber: [Hackers exploit data in error messages to attack connected medical devices, report finds | Healthcare Dive](#)



Deep Dive Opinion Library Events Press Releases

Hospitals Payer Health IT Government Finances Medical Groups Telehealth

DIVE BRIEF

Hackers exploit data in error messages to attack connected medical devices, report finds

Published Sept. 27, 2018

By [Meg Bryant](#)
Contributor



Web Uygulamalarında Hata Sayfaları

- Server detayları, kullanılan framework, veritabanı vb bilgilerin ifşa olması daha odaklı saldırıların düzenlenebilmesine imkan vermektedir.
- Web uygulamalarında eninde sonunda hata çıkmaktadır. Bu haftalar gösterilirken arka plan ile ilgili detay bilgi verilmediğinden emin olunmalıdır.
- Bu sorunlar bilhassa yazılımda “exception” adı verilen istisnalarda ortaya çıkmaktadır.

ASP.NET

```
if (env.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
}
Else
{
    app.UseExceptionHandler('/Error');
...
}
```

An unhandled exception occurred while processing the request.

InvalidOperationException: JavaScript interop calls cannot be issued at this time. This is because the component is being statically rendered. When prerendering is enabled, JavaScript interop calls can only be performed during the OnAfterRenderAsync lifecycle method.

```
Microsoft.AspNetCore.Components.Server.Circuits.RemoteJSRuntime.BeginInvokeJS(long asyncHandle, string identifier, string argsJson, JSCallResultType resultType, long targetInstanceId)
```

Stack Query Cookies Headers Routing

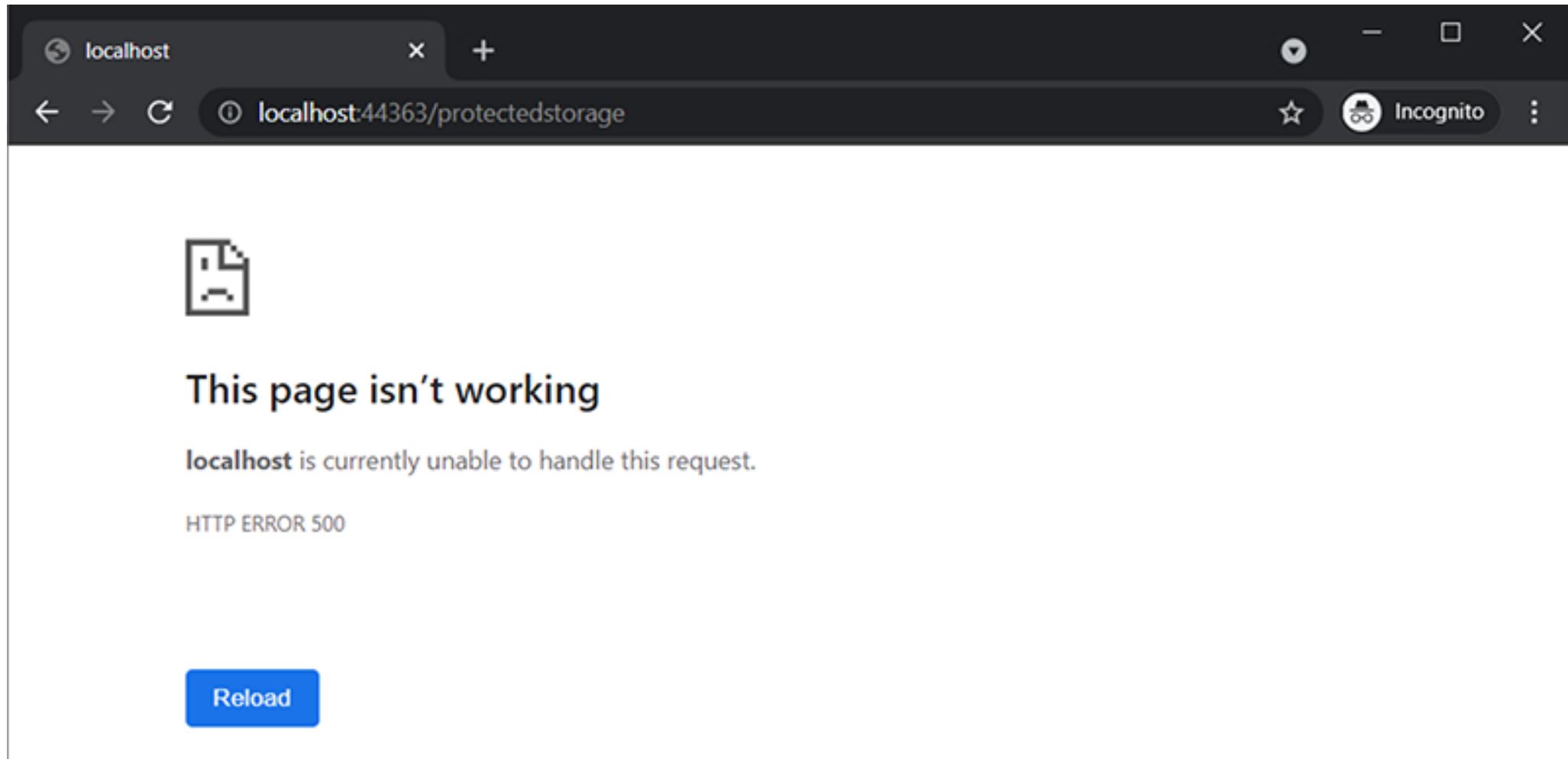
InvalidOperationException: JavaScript interop calls cannot be issued at this time. This is because the component is being statically rendered. When prerendering is enabled, JavaScript interop calls can only be performed during the OnAfterRenderAsync lifecycle method.

```
Microsoft.AspNetCore.Components.Server.Circuits.RemoteJSRuntime.BeginInvokeJS(long asyncHandle, string identifier, string argsJson, JSCallResultType resultType, long targetInstanceId)
Microsoft.JSInterop.JSRuntime.InvokeAsync< TValue >(long targetInstanceId, string identifier, CancellationToken cancellationToken, object[] args)
Microsoft.JSInterop.JSRuntime.InvokeAsync< TValue >(long targetInstanceId, string identifier, object[] args)
System.Threading.Tasks.ValueTask< TResult >.get_Result()
System.Runtime.CompilerServices.ValueTaskAwaiter< TResult >.GetResult()
Microsoft.AspNetCore.Components.Server.ProtectedBrowserStorage.ProtectedBrowserStorage.GetAsync< TValue >(string purpose, string key)
System.Threading.Tasks.ValueTask< TResult >.get_Result()
System.Runtime.CompilerServices.ValueTaskAwaiter< TResult >.GetResult()
AspNetCoreSecurity.BlazorServerSamples.Pages.ProtectedStorage.OnInitializedAsync() in ProtectedStorage.razor
+ 15.      var lastAccess = await ProtectedSessionStorage.GetAsync< DateTime >("lastAccess");
Microsoft.AspNetCore.Components.ComponentBase.RunInitAndSetParametersAsync()
Microsoft.AspNetCore.Components.Rendering.HtmlRenderer.HandleException(Exception exception)
Microsoft.AspNetCore.Components.RenderTree.Renderer.AddToPendingTasks(Task task)
Microsoft.AspNetCore.Components.Rendering.ComponentState.SetDirectParameters(ParameterView parameters)
Microsoft.AspNetCore.Components.RenderTree.RenderTreeDiffBuilder.InitializeNewComponentFrame(ref DiffContext diffContext, int frameIndex)
Microsoft.AspNetCore.Components.RenderTree.RenderTreeDiffBuilder.InitializeNewSubtree(ref DiffContext diffContext, int frameIndex)
```

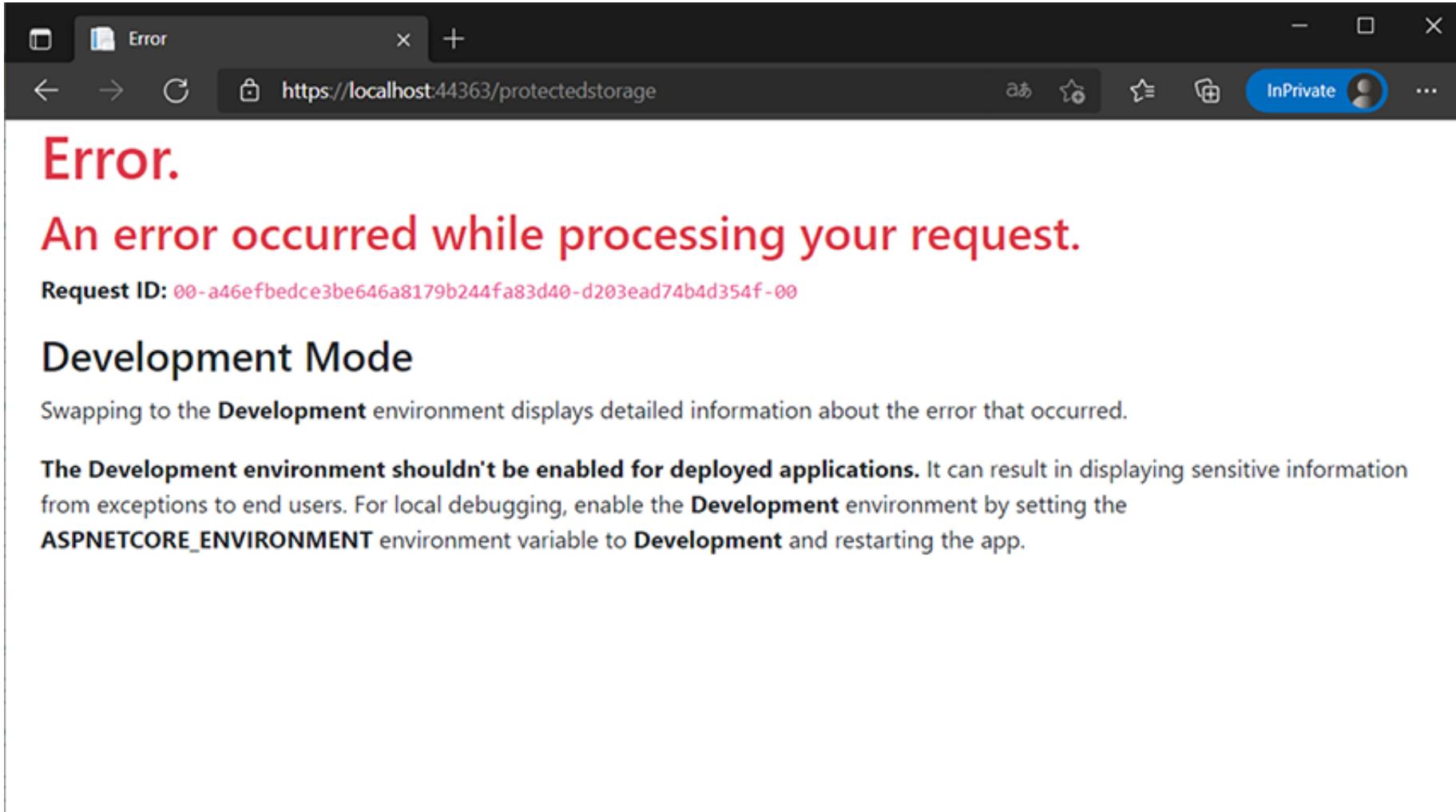
Development Mode'da Raporlananlar

- Hata mesajı
- Stack trace
- HTTP talebi ile ilgili bilgi
- Yönlendirme bilgisi

Development Mode'da Değilsek 500 Hatası Geri Döndürülür



Daha Spesifik Hata Mesajlarının Verilmesinin Sağlanması



Development Mode

Switching to the **Development** environment displays detailed information about the error that occurred.

The **Development environment shouldn't be enabled for deployed applications**. It can result in displaying sensitive information from exceptions to end users. For local debugging, enable the **Development** environment by setting the **ASPNETCORE_ENVIRONMENT** environment variable to **Development** and restarting the app.

Daha Spesifik Hata Mesajları Verilmesinin Sağlanması

- Bir önceki sayfadaki örnekte hata ID'sinin gösterilmesi bir sorun yaratmamaktadır çünkü anlamlı bir bilgi içermemektedir
- Bu ID ve ilave bilgiler kayıt altına alınır. Kullanıcı daha sonra bu sorun ile ilgili destek istedığında bu ID bilgisini de ileterek daha hızlı çözüm üretilmesi sağlanabilir.

Status Code Hata Sayfaları

- Bazı browserlar hata koduna göre bilgilendirici hata sayfalarını otomatik olarak oluşturmaktadır.
- Her kullanıcı 404 veya 500 hata kodunun ne olduğunu bilmek zorunda değildir ve bu tarz sayfalar kullanıcılaraya yardımcı olmaktadır.
- Kendi uygulamamız için tarayıcının hazırladığı varsayılan bilgilendirme sayfası yerine site tasarımımız ile uyumlu hata bilgilendirme sayfaları hazırlamak daha uygun bir çözüm olacaktır.
- `app.UseStatusCodePagesWithReExecute('/Error/{0}');`

API'larda Hata Mesajı

The screenshot shows the Postman application interface. In the center, there is a list of API requests. One request is highlighted with a GET method and the URL `https://localhost:44354/api/DayOfWeek?name=`. Below this, the detailed view for this request shows the following:

- Method:** GET
- URL:** `https://localhost:44354/api/DayOfWeek?name=`
- Params:** ●
- Headers:** (8)
- Body:** (Pretty, Raw, Preview, Visualize, JSON)
- Pre-request Script:** None
- Tests:** None
- Settings:** None

On the right side of the detailed view, it indicates **Status: 500** with a red error icon.

The **Body** tab displays the error response in JSON format:

```
1 {  
2   "type": "https://tools.ietf.org/html/rfc7231#section-6.6.1",  
3   "title": "An error occurred while processing your request.",  
4   "status": 500,  
5   "traceId": "00-486e3f8ca16f1c49a444390706029d28-ff3d9cf43f42d04c-00"  
6 }
```

At the bottom of the Postman window, there are buttons for **Find and Replace** and **Console**.

Log Kaydı Tutma ve Kontroller

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

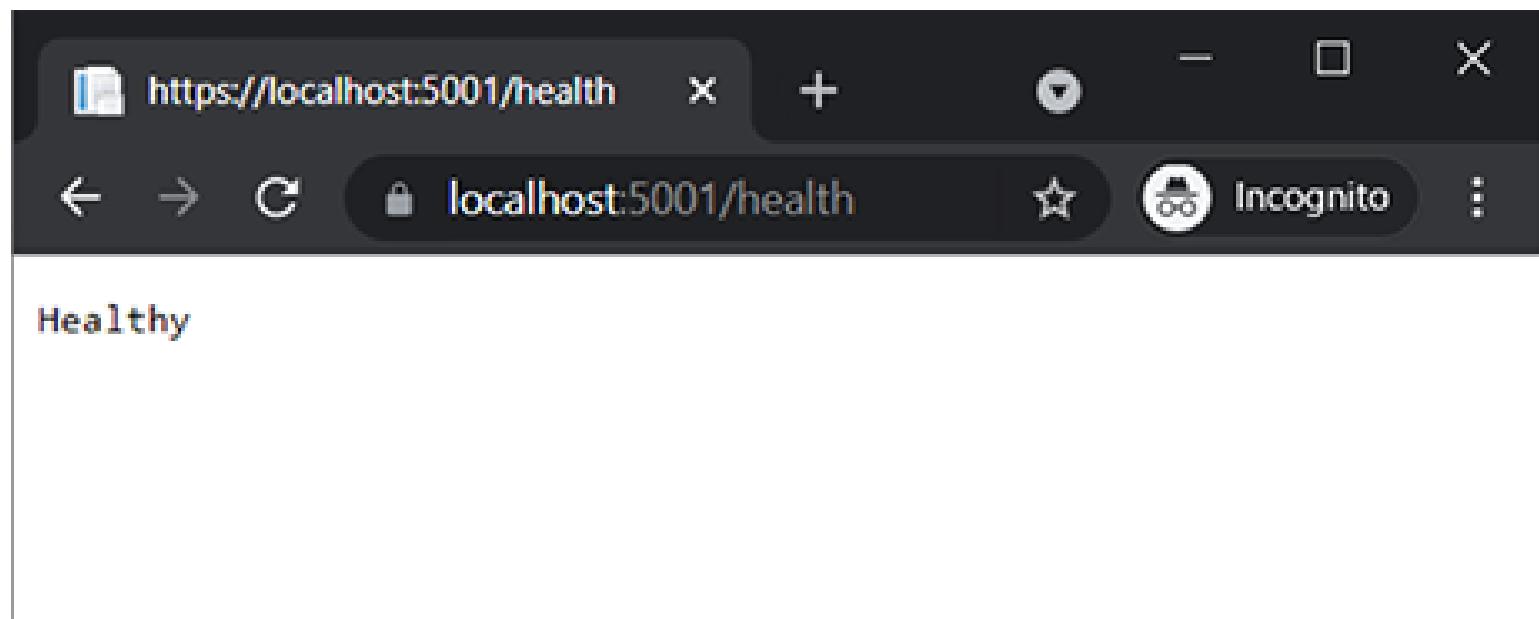
Dr. Aydın Erden

- Uygulamanın herşeyin yolunda olduğu konusunda izlenmesi gerekmektedir
- Olası sorunların irdelenebilmesi için mutlaka log kayıtları tutulmalıdır

Mevcut Durumun İzlenmesi

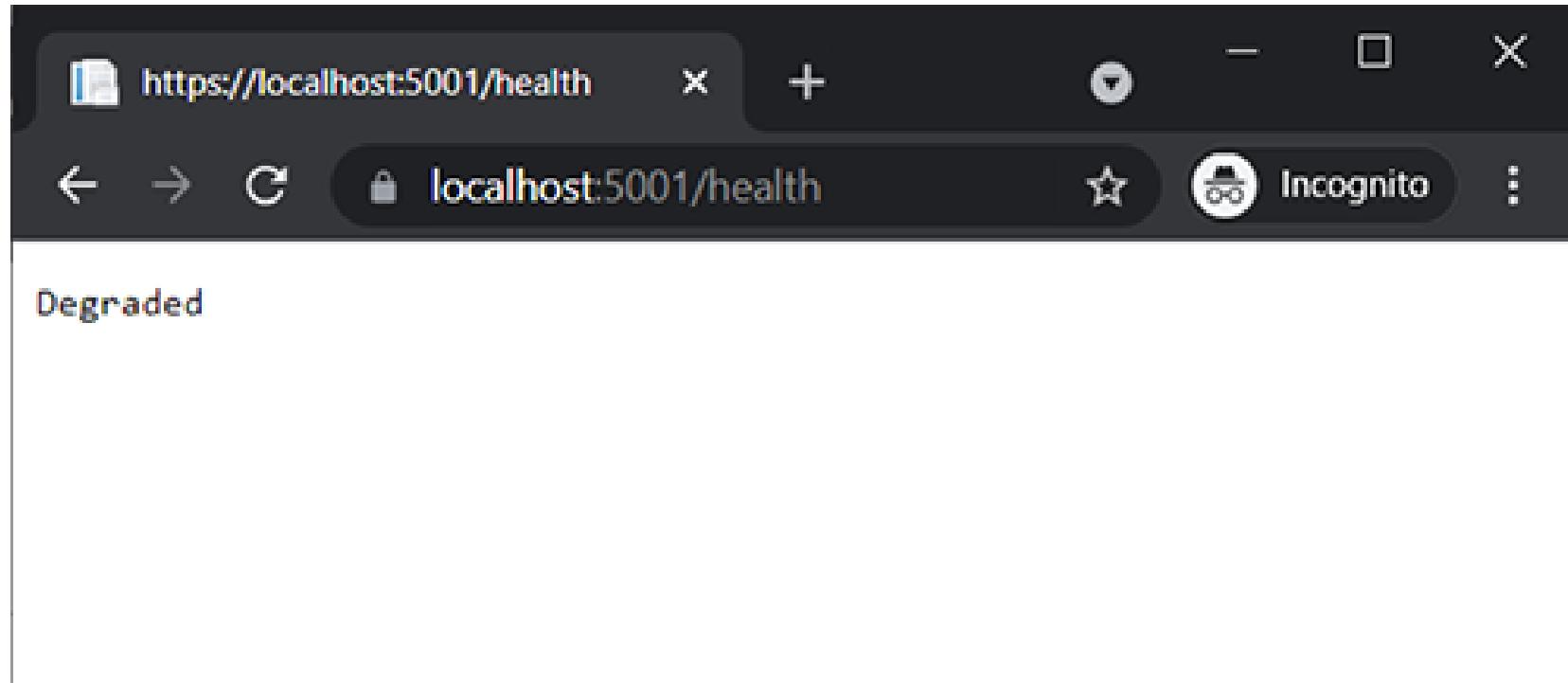
- Uygulamada oluşturulacak çeşitli bağlantı noktaları ile uygulamanın o anda, daha önce belirlenen parametrelere göre çalışıp çalışmadığı ara kontrol edilmelidir.
- Bu servis ASP.NET'te otomatik olarak aktive değildir.
- Microsoft.Extensions.Diagnostics.HealthChecks paketi yüklenip uygun ayarlamalar yapılarak active edilmelidir.

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapRazorPages();
    endpoints.MapHealthChecks('/health');
});
```



- Healthy()—Sorunsuz çalışıyor
- Degraded()—Halen çalışmakta fakat bazı kısımlarda problem var
- Unhealthy()—Uygulama bekleniği şekilde çalışmıyor

```
builder.Services
    .AddHealthChecks()
    .AddCheck('Changing health states', () =>
{
    return (DateTime.Now.Second % 3) switch
    {
        0 => HealthCheckResult.Healthy(),
        1 => HealthCheckResult.Degraded(),
        _ => HealthCheckResult.Unhealthy(),
    };
});
```



IHealthCheck

```
public Task<HealthCheckResult>
CheckHealthAsync(HealthCheckContext context,
CancellationToken cancellationToken = default)
```

```
using Microsoft.Extensions.Diagnostics.HealthChecks;

namespace AspNetCoreSecurity.RazorSamples.Classes
{
    public class ChangingHealthStates : IHealthCheck
    {
        public Task<HealthCheckResult> CheckHealthAsync(HealthCheckContext context, CancellationToken cancellationToken = default)
        {
            return (DateTime.Now.Second % 3) switch
            {
                0 => Task.FromResult(HealthCheckResult.Healthy()),
                1 => Task.FromResult(HealthCheckResult.Degraded()),
                _ => Task.FromResult(HealthCheckResult.Unhealthy()),
            };
        }
    }
}
```

Program.cs

```
builder.Services
```

```
    .AddHealthChecks()
```

```
    .AddCheck<ChangingHealthStates>('Changing health states');
```

AspNetCoreSecurity.RazorSamples - NuGet: AspNetCoreSecurity.RazorSamples

NuGet: AspNetCore...urity.RazorSamples

Browse Installed Updates

HealthCheck Include prerelease

Packag source: nuget.org

NuGet Package Manager: AspNetCoreSecurity.RazorSamples

	AspNetCore.HealthChecks.Redis by Xabril Contributors, 4,73M downloads	5.0.2
	HealthChecks.Redis is the health check package for Redis.	
	AspNetCore.HealthChecks.Rabbitmq by Xabril Contributors, 4,45M downloads	5.0.1
	HealthChecks.RabbitMQ is the health check package for RabbitMQ.	
	AspNetCore.HealthChecks.Uris by Xabril Contributors, 6,06M downloads	5.0.1
	HealthChecks.Uris is a simple health check package for Uri groups.	
	AspNetCore.HealthChecks.Npgsql by Xabril Contributors, 3,32M downloads	5.0.2
	HealthChecks.Npgsql is a health check for Postgres Sql.	
	AspNetCore.HealthChecks.MongoDb by Xabril Contributors, 2,95M downloads	5.0.1
	HealthChecks.MongoDb is the health check package for MongoDb.	

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

Kontroller için ek paketler

- [GitHub - Xabril/AspNetCore.Diagnostics.HealthChecks: Enterprise HealthChecks for ASP.NET Core Diagnostics Package](#)
- [AspNetCore.Diagnostics.HealthChecks/src at master · Xabril/AspNetCore.Diagnostics.HealthChecks · GitHub](#)

AspNetCore.Diagnostics.HealthChecks.Uris

```
services.AddHealthChecks().AddUrlGroup(  
    new Uri('https://www.marmara.edu.tr/'),  
    timeout: TimeSpan.FromSeconds(3))  
    tags: new string[] { 'URLs' });
```

```
endpoints.MapHealthChecks(  
    '/health-uris',  
    new HealthCheckOptions()  
    {  
        Predicate = (item) => item.Tags.Contains('URIs')  
    });
```

Kullanıcı Arayüzü Ekleme

- AspNetCore.HealthChecks.UI
- AspNetCore.HealthChecks.UI.Client
- AspNetCore.HealthChecks.UI.InMemory



Health Checks Status

Polling interval: 10 secs

Stop polling

⊕	NAME	HEALTH	ON STATE FROM	LAST EXECUTION
-	Razor Site Health Checks	!	2021-09-30T16:03:48.7528367+02:00	9/30/2021, 4:04:22 PM

NAME	TAGS	HEALTH	DESCRIPTION	DURATION	DETAILS
Changing health states		!	Unhealthy	00:00:00.0000112	⌚
uri-group	URLs	✓	Healthy	00:00:01.0474298	⌚

Health Checks

Webhooks

Log Kaydı Tutma

- ASP.NET uygulamalarında varsayılan olarak log servisi mevcuttur
 - Console
 - Debug - /var/log/message
 - Event
 - EventLog – Windows işletim sistemlerindeki eventlere kaydeder

ILogger

```
public interface ILogger<out TCategoryName> : ILogger
{
    IDisposable BeginScope<TState>(TState state);

    bool IsEnabled(LogLevel logLevel);

    void Log<TState>(LogLevel logLevel, EventId eventId,
                      TState state, Exception exception,
                      Func<TState, Exception, string> formatter);
}
```

Log metodu argümanları

- `logLevel`
- `eventId`—kaydedilen durumun event id'si
- `state`—kaydedilen değer
- `exception`—ilişkili exception
- `Formatter` –format fonksiyonu

amples.Pages.LoggingModel[9]
gging

Debug



```
urity.RazorSamples.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
urity.RazorSamples.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
urity.RazorSamples.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
urity.RazorSamples.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
urity.RazorSamples.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files  
urity.RazorSamples.Pages.LoggingModel: Information: Calling OnGet met
```

oints

Exception Settings

Command Window

Immediate Window

Output

Er

Log Level

1. Trace- LogTrace()
2. Debug - LogDebug()
3. Information - LogInformation()
4. Warning - LogWarning()
5. Error - LogError()
6. Critical - LogCritical()
7. None

API ve SPA'ların Korunması

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

API güvenliği

- Standard web sitelerinde kullanıcılar login olabilir ve cookie vasıtası ile erişim yetisi var mı control edilebilir.
- Fakat API'lara bağlanan uygulamalar her zaman browser değildir bu nedenle cookie kaydetme ve gönderme işlemlerini yapamayabilir

Token kullanımı

- JWT – JSON Web Token
- <https://datatracker.ietf.org/doc/html/rfc7519>
- JWT yapısı içerisindeki veri imzalanabilir veya şifrelenebilir

JWT

- iss—JWT sağlayıcısı
- sub—Konusu (örneğin: kullanıcı adı veya ID'si)
- aud—Varsayılan alıcı
- exp—Geçerlilik bitiş tarih ve saatı
- nbf—Geçerlilik başlangıç zamanı
- iat—Tanımlantığı zaman
- jti—JWT ID'si

Uygulamada Kullanım için Gerekli Adımlar

- Web uygulamasına API eklenir
- Uygulamaya JWT middleware'i eklenir
- Login API'I eklenir. Bu API token döndürür
- Bir başka API daha eklenir ve bu API token Kabul eder

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman Invite Upgrade

No Environment

https://localhost:5001/api/tokenauth/login

POST https://localhost:5001/api/tokenauth/login Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies Beautify

Body (1)

```
1 {
2   "Email": "new-user@example.com",
3   "Password": "Secret+123"
4 }
```

Body Cookies Headers (4) Test Results Status: 200 OK Time: 12.47 s Size: 529 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "token": "eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJuZXctdXNlcjFAZXhhbXBsZS5jb20iLCJlbWFpbCI6Im5ldy11c2VyMUBleGFtcGxlLmNvbSIiImp0aSI6IjI2M2UxNmUxLTE4NzgtNDE5MS04YwQxLTUxYTFiNTExM2I4NyIsIm5iZiI6MTYzODQ3OTUwOCwiZXhwIjoxNjM4NDgwMTA4LCJpYXQiOjE2Mzg0Nzk1MDh9.eyJzb21cblZtW8Q9gcUi0RhewM4lYTY7ijSmsuDMailTubOpDcf20G988uGU5L28fsXAH00MoSg-S0ml2RuA",
3   "expires": "2021-12-02T21:21:48Z"
4 }
```

Find and Replace Console Bootcamp Runner Trash

The screenshot shows the Postman application interface. At the top, there are tabs for Home, Workspaces, API Network, Reports, and Explore, along with a search bar and various icons for invite, settings, and notifications. Below the header, a sidebar lists environments: 'No Environment' (selected), 'Save', 'Edit', 'Delete', and 'Copy'. The main workspace displays a POST request to 'https://localhost:5001/api/tokenauth/login'. The request method is set to POST, and the URL is https://localhost:5001/api/tokenauth/login. The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "Email": "new-user@example.com",
3   "Password": "Secret+123"
4 }
```

Below the body, the response status is shown as 200 OK with a response size of 529 B. The response body is displayed in pretty JSON format:

```
1 {
2   "token": "eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJuZXctdXNlcjFAZXhhbXBsZS5jb20iLCJlbWFpbCI6Im5ldy11c2VyMUBleGFtcGxlLmNvbSIiImp0aSI6IjI2M2UxNmUxLTE4NzgtNDE5MS04YwQxLTUxYTFiNTExM2I4NyIsIm5iZiI6MTYzODQ3OTUwOCwiZXhwIjoxNjM4NDgwMTA4LCJpYXQiOjE2Mzg0Nzk1MDh9.eyJzb21cblZtW8Q9gcUi0RhewM4lYTY7ijSmsuDMailTubOpDcf20G988uGU5L28fsXAH00MoSg-S0ml2RuA",
3   "expires": "2021-12-02T21:21:48Z"
4 }
```

At the bottom, there are buttons for Find and Replace, Console, Bootcamp, Runner, and Trash.

JWT Token Kısımlar

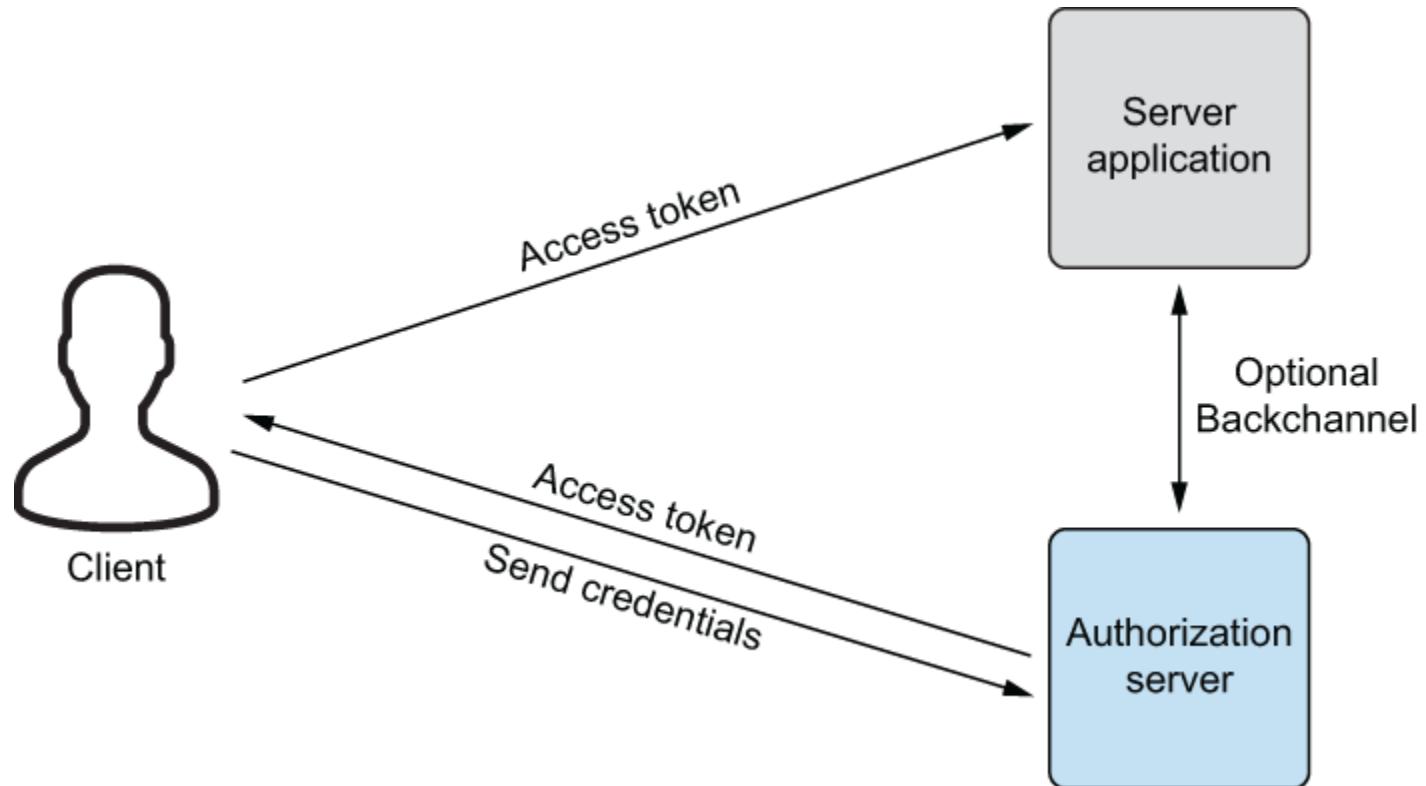
1. Header bilgisi ve imza için kullanılan algoritma
2. İçerik
3. İmza

[JSON Web Tokens - jwt.io](https://jwt.io)

Token Kullanımı Riski

OAuth ve OpenID connect

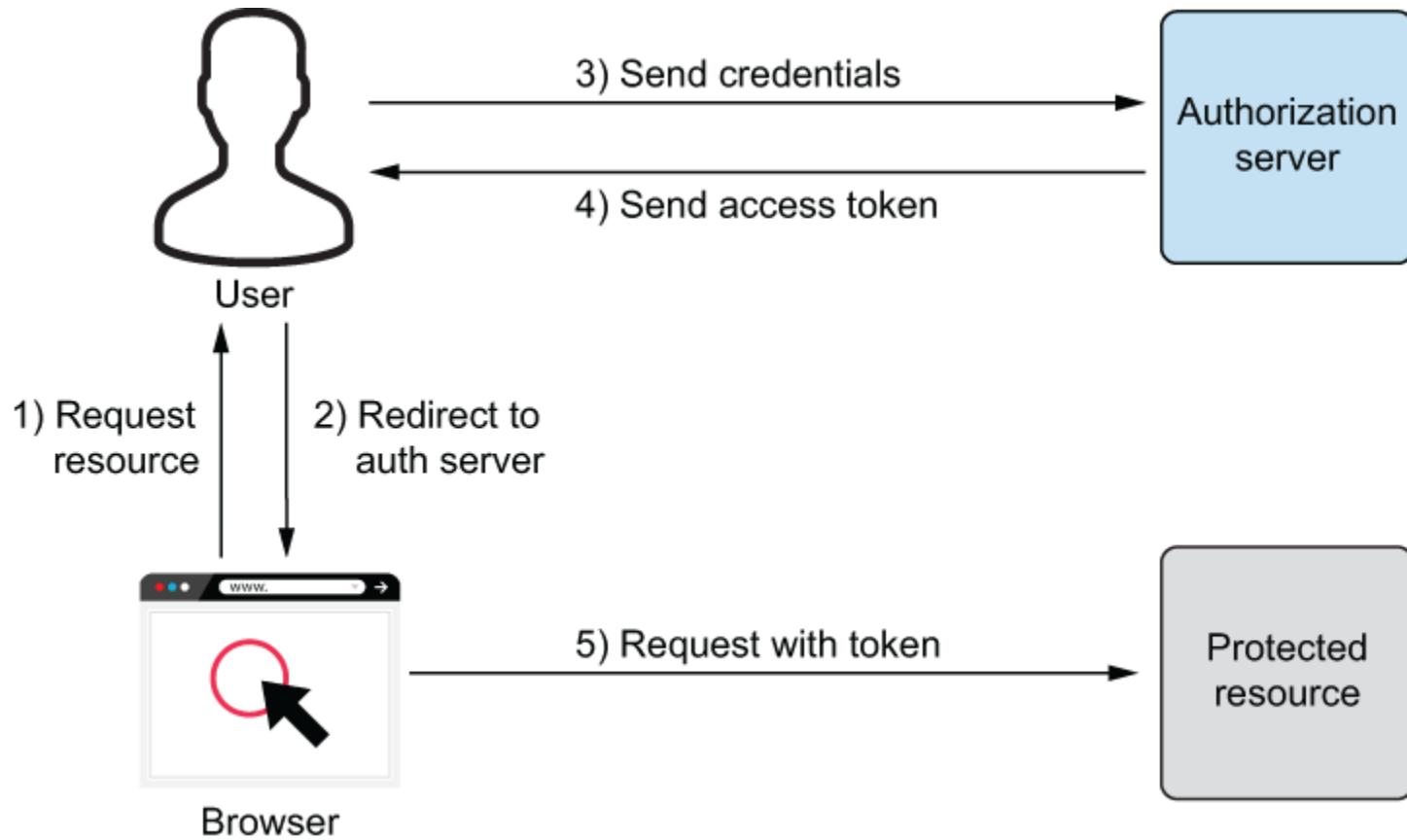
- OAuth – client ve server arasında bir de authorization server mevcuttur



OpenID

- <https://openid.net/developers/how-connect-works/>
- [Certified OpenID Connect Implementations - OpenID Foundation](#)
- OAuth – kullanıcının nelere erişebileceğini tanımlar
- OpenID – kullanıcının kim olduğunu tanımlar

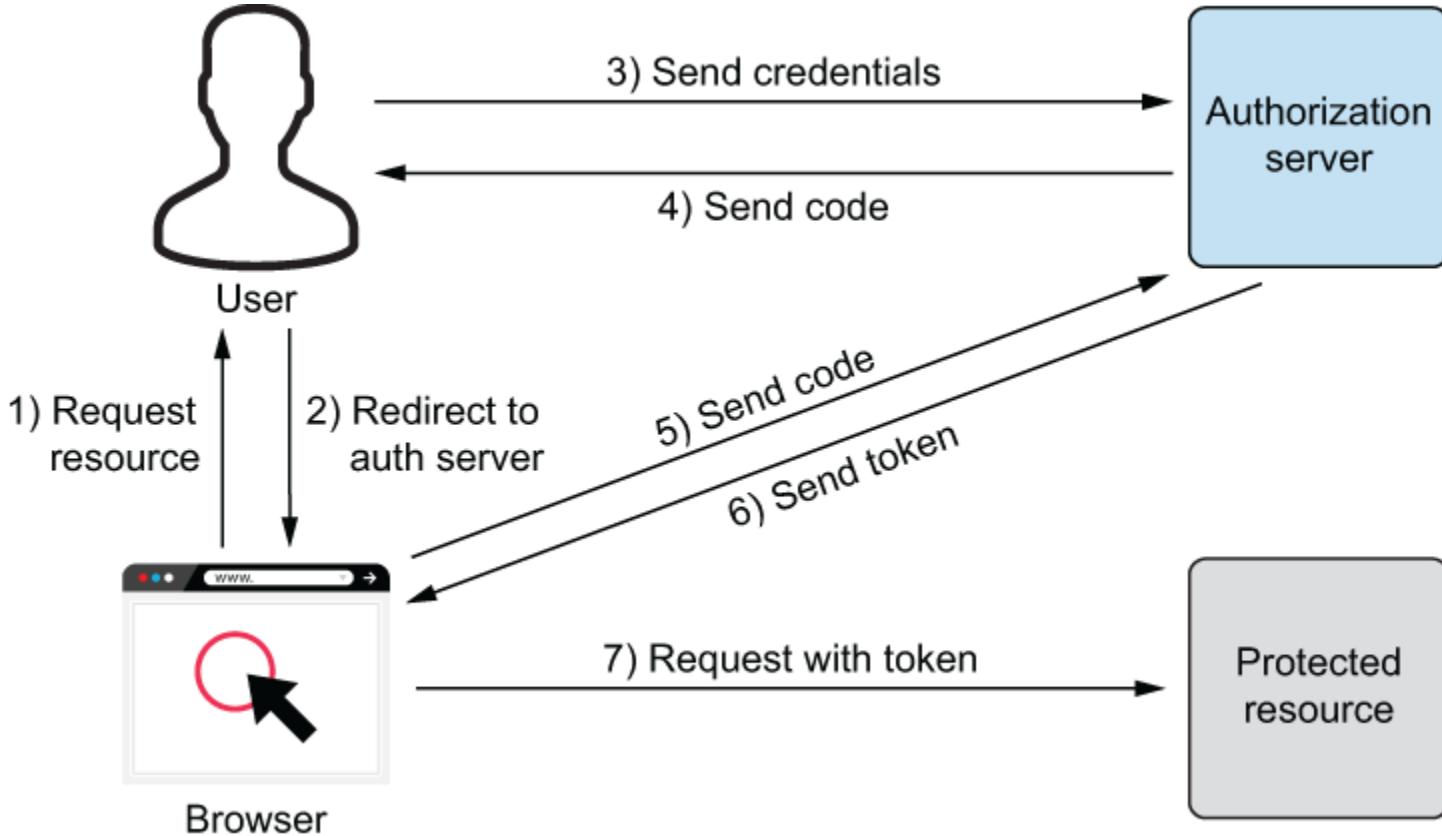
OAuth Implicit Flow – SPA için



OAuth Implicit Flow – SPA için

- Bu kullanım tarzından vazgeçilmektedir
- Döndürüler token yetkisiz şekilde elde edilebilir:
 - Referrer HTTP headarı
 - Broswer history
 - Log kayıtları
 - Proxy serverlar

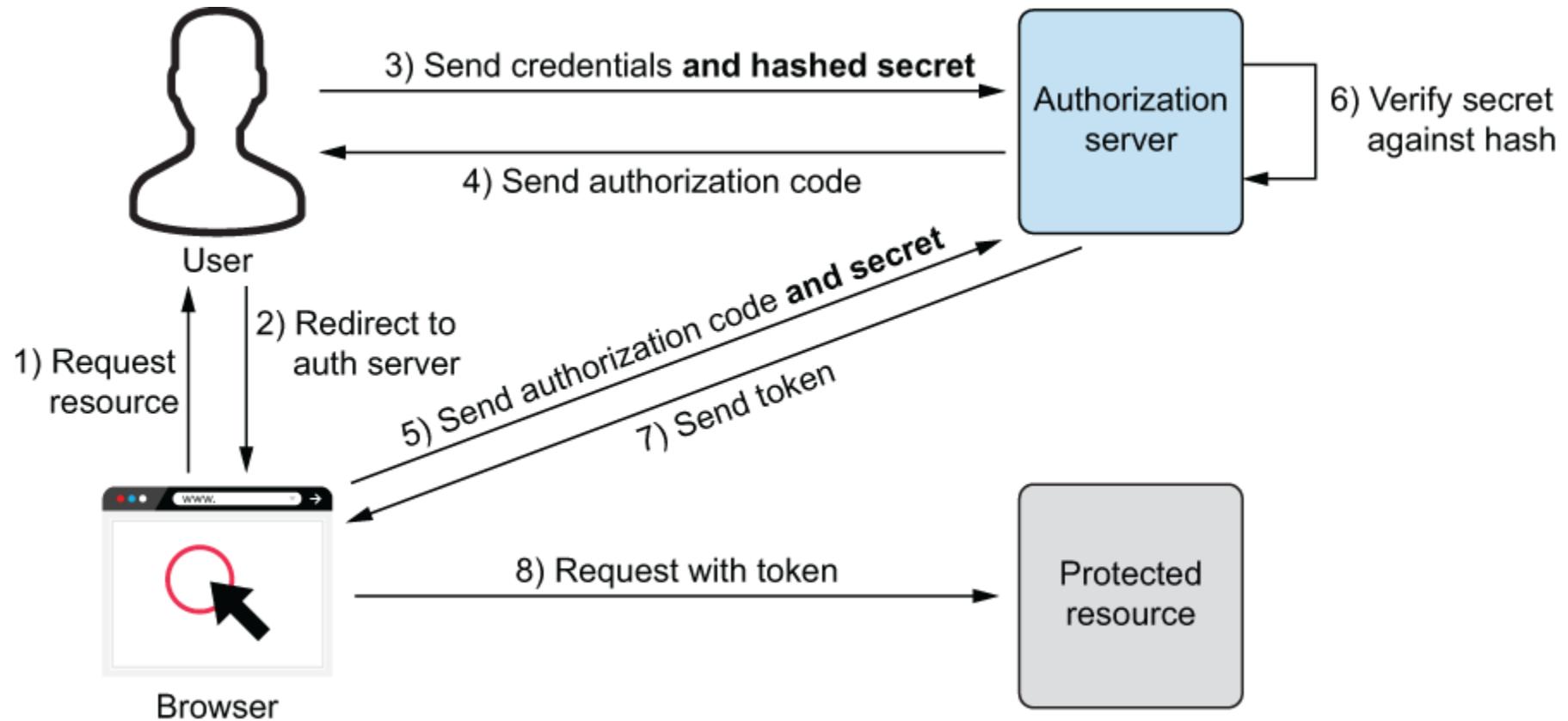
Authorization Code Flow



Authorization Code Flow

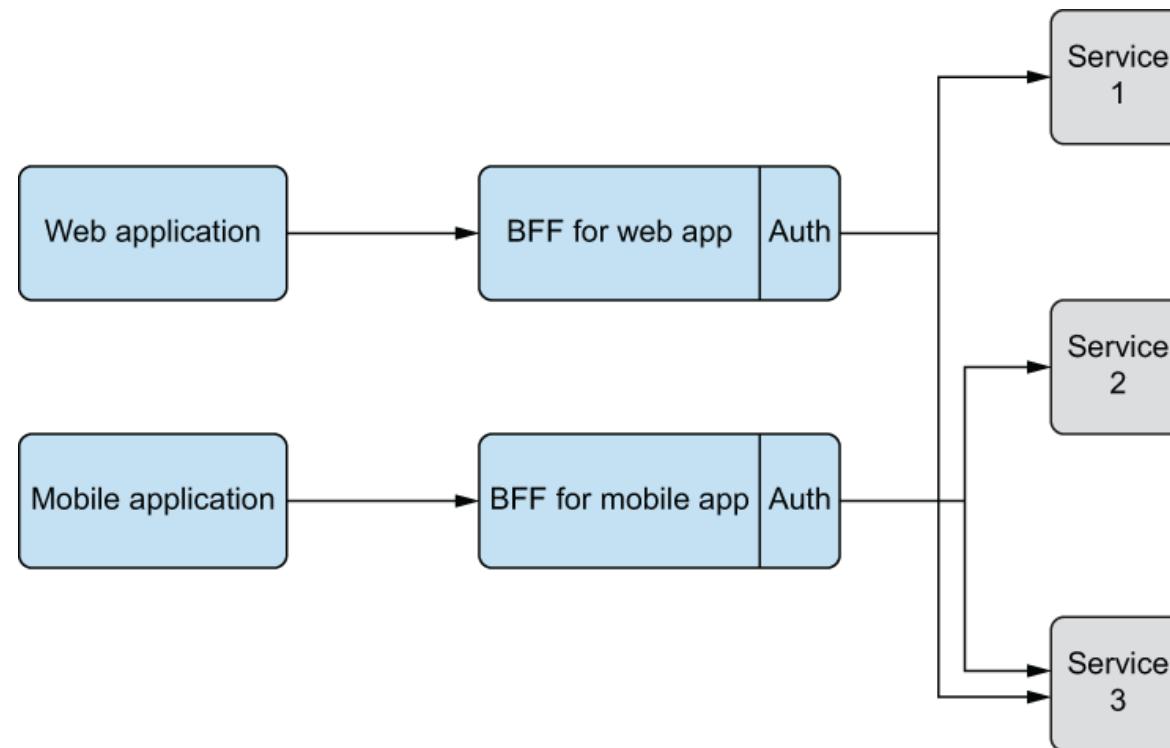
- Bu yöntemin de zayıflıkları bulunmaktadır.
 - Yetkilendirme kodu referrer HTTP headarı nedeni ile network üzerinde ifşa olabilir
 - Web uygulamasında açık bir yönlendirme var ise yetkisiz erişim ile istenilen sayfaya yönlendirilerek kod elde edilebilir
 - Authorization code injection ile daha önce gördüğümüz cross site request forgery yönetmine benzer şekilde yetkisiz işlem yapılabilir.

Authorization code + PKCE (Proof Key for Code Exchange)



BFF – Back-end for Front-end

- The Back-end for Front-end Pattern (BFF) (philcalcado.com)



Harici Paketlerin Güvenliği

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

Ua-parser.js

- [ua-parser-js - npm \(npmjs.com\)](#)
- Bir javascript paketidir
- Browser türünün, işletim sisteminin, cpu türünün vs tespitini sağlar
- Haftalık indirme sayısı 11 milyonun biraz üzerindedir
- 0.7.0 versiyonu 2014'te, 0.7.28 versiyonu ise 2021'de yayınlanmıştır.
- 2000'in üzerinde başka paket bu pakete bağımlıdır
- Ekim 2021'de CISA 0.7.29 versiyonunda malware olduğu uyarısını yayınlamıştır (izinsiz kripto madenciliği)

Malware Discovered in Popular NPM Package, ua-parser-js | CISA

CYBERSECURITY &
INFRASTRUCTURE
SECURITY AGENCY



AMERICA'S CYBER DEFENSE AGENCY

Search

Topics ▾

Spotlight

Resources & Tools ▾

News & Events ▾

Careers ▾

About ▾

[Home](#) / [News & Events](#) / [Cybersecurity Advisories](#) / [Alert](#)

ALERT

Malware Discovered in Popular NPM Package, ua-parser-js

Last Revised: October 22, 2021

Open source developer corrupts widely-used libraries, affecting tons of projects - The Verge

TECH / SECURITY

Open source developer corrupts widely-used libraries, affecting tons of projects

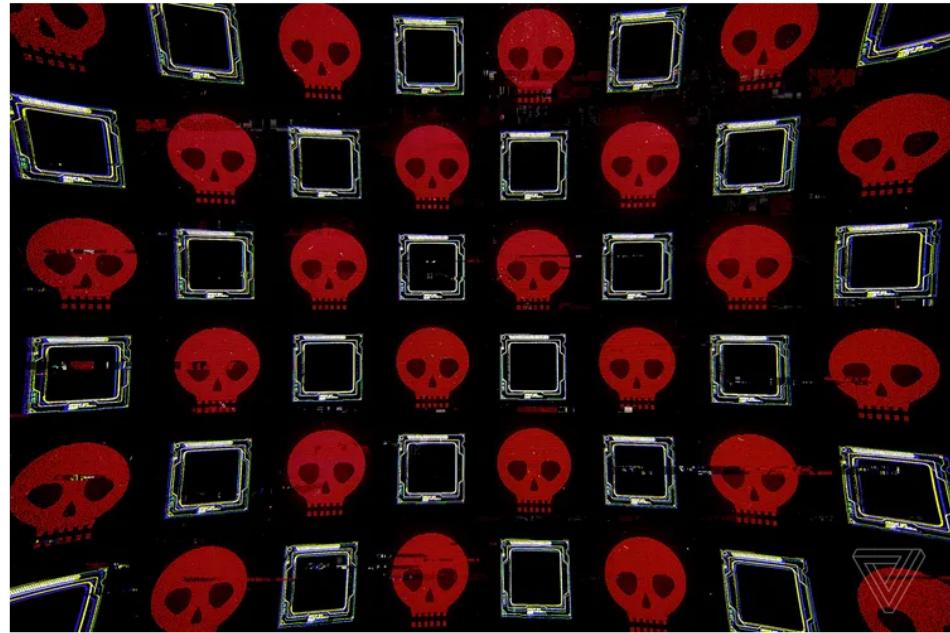


Illustration by Alex Castro / The Verge

/ He pushed corrupt updates that trigger an infinite loop

By [Emma Roth](#), a news writer who covers the streaming wars, consumer tech, crypto, social media, and much more. Previously, she was a writer and editor at MUO.

Jan 9, 2022, 11:58 PM GMT+3 | □ 0 Comments / 0 New



**Color.js ve
Faker.js
paketlerinin
kasıtlı
olarak
bozulması**

- Temmuz 2021'de yapılan bir çalışmada NuGet paket sisteminde yer alan 51 pakette kötüye kullanılabilecek açıklar bulunduğu tespit edilmiştir
- [Third-party code comes with some baggage \(reversinglabs.com\)](#)

Threat Research | July 7, 2021

Third-party code comes with some baggage

Recognizing risks introduced by statically linked third-party libraries



BLOG AUTHOR

Karlo Zanki, Reverse Engineer at ReversingLabs. [READ MORE...](#)

Angular Paketinin Yüklenmesi

```
Windows PowerShell
PS D:\data\projects\blank-angular-app> npm install @angular/cli
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 235 packages, and audited 236 packages in 21s

23 packages are looking for funding
  run `npm fund` for details

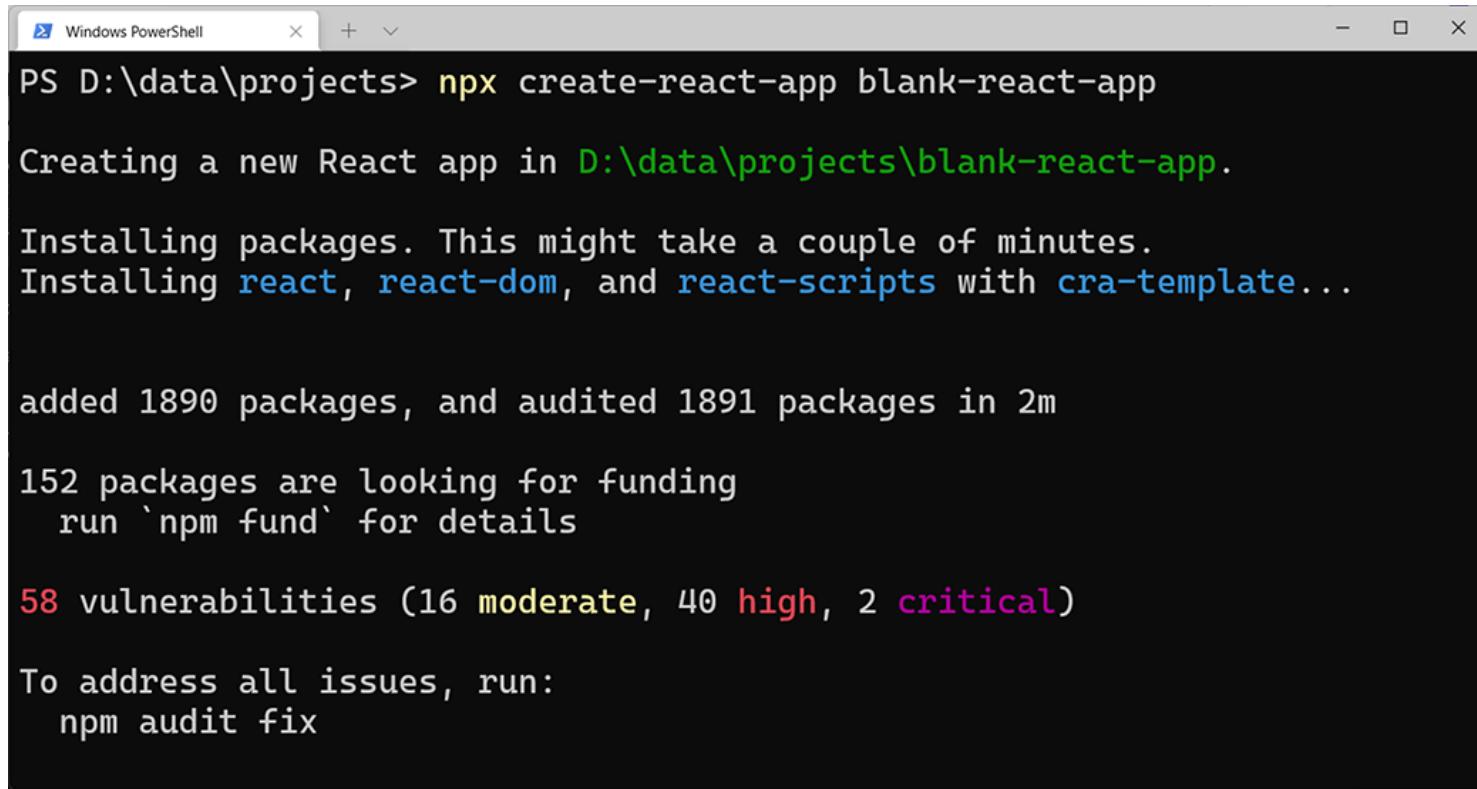
found 0 vulnerabilities
PS D:\data\projects\blank-angular-app> |
```

Angular Paketinin Yüklenmesi Sorunları

- Bağımlılıklardan birtanesi artık desteklenmiyor
- Diğer bir bağımlılığın yeni versiyonu var
- Diğer bir bağımlılık ise kullanımından kalkmış (deprecated) durumunda

React Uygulaması Oluşturulması

- 58 ayrı güvenlik açığı raporlanmakta



```
PS D:\data\projects> npx create-react-app blank-react-app
Creating a new React app in D:\data\projects\blank-react-app.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
added 1890 packages, and audited 1891 packages in 2m
152 packages are looking for funding
  run `npm fund` for details
58 vulnerabilities (16 moderate, 40 high, 2 critical)

To address all issues, run:
  npm audit fix
```

Tek Tek Paketleri Kontrol Etmek

- Angular paketinin 350 ayrı pakete bağımlılığı söz konusudur
- Tek tek paketleri kontrol etmek verimsiz olacaktır
- Npm audit aracının kullanımı daha kolay ve verimli bir yöntemdir
- Yüklenen tüm bağımlılıkları tek tek, zayıflıkları bulunan paketlerin listelendiği veritabanı ile kontrol etmektedir.
- 4 ayrı uyarı seviyesi vardır:
 - Düşük
 - Orta
 - Yüksek
 - Kritik

Npm Audit Kullanımı

- Önceki iki örnek ekranada npm audit komutu otomatik olarak çalıştırılmıştır. Vulnerability raporlamaları o nedenle görülmektedir.
- **Npm audit fix** – tüm sorunlu paketleri en yeni versiyonlarına günceller. Fakat bu durum da başka sorunlara yol açabilir. Sorunun sebebi SemVer – semver.org ‘dir

Semantic Versioning – semver.org

- Uygulama 1.2.3
 - 1 major version
 - 2 minor version
 - 3 patch version
- Temel SemVer kuralları
 - Geriye doğru uyumluluğu etkileyen değişiklikler yeni bir major version numarası verilmesini gerektirir
 - Geriye doğru uyumluluğu etkilemeyen yeni özellikler eklenmesi yeni bir minor version numarası verilmesini gerektirir
 - Uygumaladaki bir hatanın geriye doğru uyumluluğu etkilemeden düzeltilmesi yeni bir patch numarası verilmesini gerektirir

Package.json

- Bağımlı olunan tüm paketlerin bilgisini içerir
- React uygulaması için örnek

```
'dependencies': {  
  '@testing-library/jest-dom': '^5.14.1',  
  '@testing-library/react': '^11.2.7',  
  '@testing-library/user-event': '^12.8.3',  
  'react': '^17.0.2',  
  'react-dom': '^17.0.2',  
  'react-scripts': '^0.9.5', - bu versiyondan 1.0.0'a kadar olan (fakat 1.0.0 hariç) tüm versiyonların  
  kullanılabileceğini belirtmektedir  
  'web-vitals': '^1.1.2'  
}
```

- 'react-scripts': '^0.9.5', - bu versiyondan 1.0.0'a kadar olan (fakat 1.0.0 hariç) tüm versiyonların kullanılabileceğini belirtmektedir
- Eğer sonraki version 1.0.0 ise **npm audit fix** paketi yeni versiyona güncelleyemez. Çünkü bu version geriye doğru uyumlu değildir.
- SemVer güvenlik mekanizmasını **npm audit fix-force** ile kapatabiliriz. Fakat bu durumda da geriye doğru uyumlu olmayan bir version yükleneneğinden uygulamanın eskisi gibi çalışacağı garanti edilemez.

- Kesin doğru bir yöntem mevcut değildir. Ancak aşağıdaki yöntemlerin izlenmesi tavsiye edilir.
 - Bağımlılıkları uygulamanın çalışmasını bozmayacak şekilde en güncel versiyonunda tutmaya özen gösterin
 - Her bir version yükseltmesinde uygulamanızın halen beklentiği şekilde çalıştığını test edin
 - Eğer yükseltme şansınız yok ise ve bir vulnerability olarak raporlandı ise bu vulnerability'nin ne olduğunu kendiniz bakıp tedbir almak gerekir mi karar verin
 - Her vulnerability gerçek anlamda ciddi bir sorun içermiyor olabilir.
<https://overreacted.io/npm-audit-broken-by-design/>

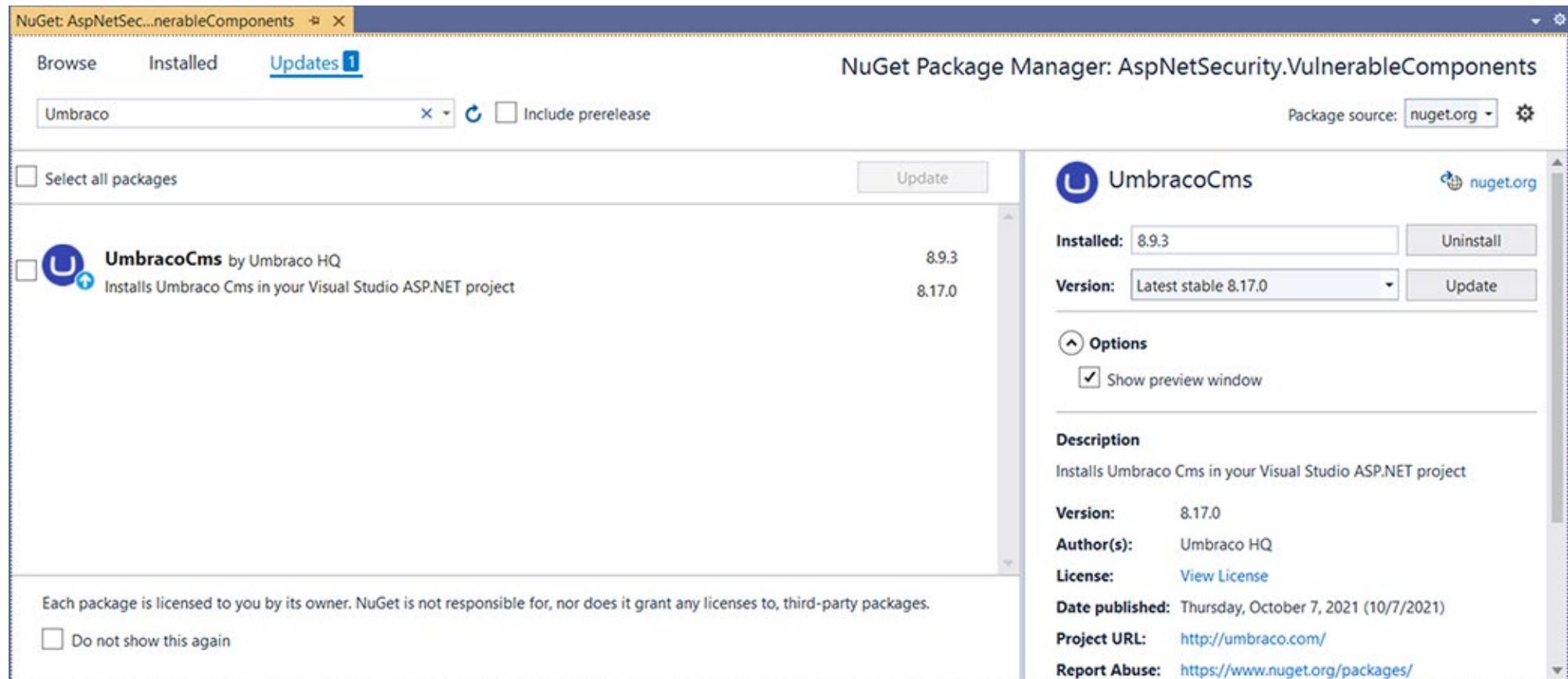
NuGet Sitesinde İlgili Versiyonda Sorun Olduğunun Raporlanması

The screenshot shows a Microsoft Edge browser window displaying the NuGet Gallery page for the UmbracoCms package. The URL in the address bar is <https://www.nuget.org/packages/UmbracoCms/8.9.3>. The page header includes the NuGet logo, navigation links for Packages, Upload, Statistics, Documentation, Downloads, and Blog, and an InPrivate sign-in button.

The main content area displays the package details for UmbracoCms version 8.9.3. It features a large yellow warning box containing the text: "⚠ This package has at least one **vulnerability** with **moderate** severity. It may lead to specific problems in your project. Try updating the package version." Below this, there is a message indicating a newer version is available: " ⓘ There is a newer version of this package available. See the version list below for details." A note at the bottom states "Requires NuGet 4.1.0 or higher."

On the right side, there is a "Downloads" section with "Full stats →" and metrics: "Total 3.3M", "Current version 829", and "Per day average 982". At the bottom right, there are "About" and "Last updated 7 months ago" links.

VisualStudio içerisinde de controller yapılmaktadır



```
PS C:\Users\cwenz\source\repos\AspNetSecurity.VulnerableComponents> dotnet list package --vulnerable
```

The following sources were used:

<https://api.nuget.org/v3/index.json>

C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\

Project `AspNetSecurity.VulnerableComponents` has the following vulnerable packages

[net5.0]:

Top-level Package	Requested	Resolved	Severity	Advisory URL
> UmbracoCms	8.9.3	8.9.3	Moderate	https://github.com/advisories/GHSA-4vp3-vfww-8648

Komut
Satırından
Kontrol

- dotnet list package --vulnerable // sadece üst düzey paketler
- dotnet list package --vulnerable --include-transitive // üst düzey ve alt düzey paketler birlikte

Denetim Araçları

SGP7023 – Güvenli Kodlama ve Yazılım Güvenliği

Dr. Aydın Erden

- 2019 yılında GitHub Semmle adlı kod analiz uygulaması geliştiren şirketi satın aldı.
- Bir yıl sonra bu şirketin ürününü sistemlerine entegre ederek hali hazırda kayıtlı kodları denetlediler.
- 12.000 repository tarandı. 20.000 açık tespit edildi
- [GitHub makes code vulnerability scanning feature public | Computer Weekly](#)
- 10 web uygulamasından 9'unda sorun bulunmaktadır

Analiz Yöntemleri

- Dynamic analysis – uygulama çalışırken analiz edilir
- Static analysis – uygulamanın kaynak kodları analiz edilir

Dynamic Analysis

- İlgili web uygulamasının haritası çıkartılır. Tüm sayfalar, API'lar listelenir
- Daha sonra tespit edilen tüm bu sayfa ve API'lar taranır
- Tarama esnasında analiz uygulamasında kayıtlı olan bütün potansiyel saldırı vektörleri herbir sayfa ve API'a uygulanır
- Ayrıca HTTP headerlara ve cookie ayarlarına bakılır

Static Analysis

- Uygulamanın kaynak kodunda daha önce sorun yarattığı tespit edilen potansiyel kod örüntüleri aranır.
- Mesela @Html.Raw()

ZAP

- [ZAP \(zaproxy.org\)](http://zaproxy.org)
- Uygulamayı test için site: [Bitcoin Web Site \(testsparker.com\)](http://testsparker.com)
- İki tarama türü mevcut
 - Traditional spider
 - Ajax spider – Single page applicationların testi için. Uygulamayı bir browsera yükler ve oradan test eder

Untitled Session - 2024U103-145420 - ZAP 2.14.0

File Edit View Analyse Report Tools Import Export Online Help

Standard Mode

Sites +

Contexts Default Context

Sites

Please be aware that you should only attack applications that you have been specifically given permission to test.

Quick Start Request Response Requester +

URL to attack: http://aspnet.testsparker.com/ Select...

Use traditional spider:

Use ajax spider: with Firefox Headless

Attack Stop

Progress: Attack complete - see the Alerts tab for details of any issues found

History Search Alerts Spider Active Scan +

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts (17)

- > Absence of Anti-CSRF Tokens (158)
- > Application Error Disclosure
- > Content Security Policy (CSP) Header Not Set (163)
- > Directory Browsing
- > Missing Anti-clickjacking Header (111)
- > Cookie No HttpOnly Flag (4)
- > Cookie without SameSite Attribute (5)
- > Cross-Domain JavaScript Source File Inclusion (468)
- > Server Leaks Information via "X-Powered-By" HTTP Response Header (1)
- > Server Leaks Version Information via "Server" HTTP Response Header (1)
- > X-AspNet-Version Response Header (158)
- > X-Content-Type-Options Header Missing (121)
- > Authentication Request Identified (3)
- > Modern Web Application (156)
- > Session Management Response Identified (2)
- > User Agent Fuzzer
- > User Controllable HTML Element Attribute (Potential XSS) (133)

Tespit edilen zayıflıkların değerlendirilmesi

- Her uyarı hata demek değildir.
- Hatalar 4 ayrı kategoride raporlanır
 - Kırmızı – yüksek tehdit
 - Turuncu – orta düzey tehdit
 - Sarı – düşük tehdit
 - Mavi – sadece bilgilendirme amaçlıdır

Analiz için diğer bir örnek site

- [OWASP Juice Shop | OWASP Foundation](https://juice-shop.owasp.org)
- [OWASP Juice Shop \(juice-shop.herokuapp.com\)](https://juice-shop.herokuapp.com)

Statik Kod Analizi

 **SecurityCodeScan.VS2019** 5.6.7

Requires NuGet 2.8 or higher.

.NET CLI Package Manager PackageReference Paket CLI Script & Interactive Cake

```
> dotnet add package SecurityCodeScan.VS2019 --version 5.6.7
```

README **Frameworks** Dependencies Used By Versions Release Notes

There are no supported framework assets in this package.
Learn more about [Target Frameworks](#) and [.NET Standard](#).

Downloads [Full stats →](#)

Total **6.0M**

Current version **2.6M**

Per day average **5.7K**

About

Last updated 9/5/2022

[Project website](#)

[Source repository](#)

[LGPL-3.0-or-later license](#)

[Download package \(440.32 KB\)](#)

[Open in NuGet Package Explorer](#)

[Open in FuGet Package Explorer](#)

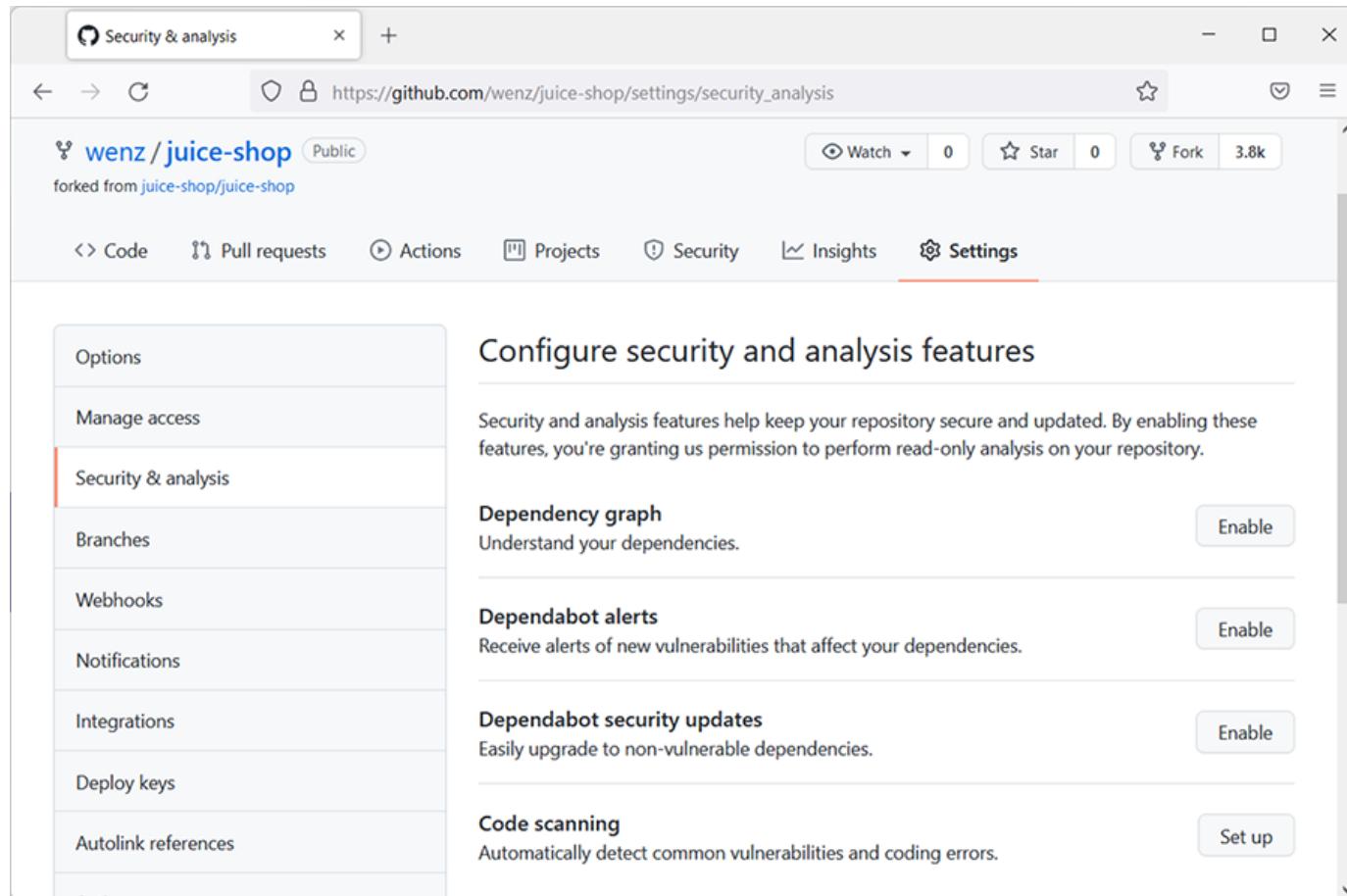
[Report package](#)

- [Security Code Scan \(security-code-scan.github.io\)](https://security-code-scan.github.io)

GitHub CodeQL

- [GitHub - github/codeql: CodeQL: the libraries and queries that power security researchers around the world, as well as code scanning in GitHub Advanced Security](#)

YAML dosyası ayarlanmalıdır



- **codeql-analysis.yml** dosyası projede **.github/workflows** klasörüne eklenmelidir
- Temel olarak hangi durumlarda kod analizinin çalıştırılacağını (push veya pull request sonrası mesela), hangi dillerin taranacağını, analiz öncesi neler yapılması (uygulamayı derlemek vb) ayarlanır

Actions · wenz/juice-shop

https://github.com/wenz/juice-shop/actions/workflows/codeql-analysis.yml

wenz / juice-shop Public

forked from juice-shop/juice-shop

Watch 0 Star 0 Fork 3.8k

Code Pull requests Actions Projects Security Insights Settings

New workflow

Workflows All workflows CI/CD Pipeline CodeQL Let me lint:fix that for you

CodeQL

codeql-analysis.yml

Filter workflow runs

1 workflow run

Event Status Branch Actor

Create codeql-analysis.yml

CodeQL #1: Commit d59b7c3 pushed by wenz

master 3 minutes ago In progress

A screenshot of a web browser displaying the GitHub Actions page for the repository 'wenz/juice-shop'. The URL in the address bar is 'https://github.com/wenz/juice-shop/actions/workflows/codeql-analysis.yml'. The page shows a single workflow named 'CodeQL' with a configuration file 'codeql-analysis.yml'. There is one workflow run listed, which was triggered by a commit ('CodeQL #1: Commit d59b7c3 pushed by wenz') and is currently in progress ('In progress'). The status bar indicates it was 3 minutes ago. The browser interface includes standard navigation buttons, a search bar, and various GitHub navigation links like 'Pull requests', 'Issues', 'Marketplace', and 'Explore'.

Code scanning alerts · wenz/juice · GitHub

https://github.com/wenz/juice-shop/security/code-scanning

Code Pull requests Actions Projects Security 51 Insights Settings

Overview Security policy Security advisories Dependabot alerts Code scanning alerts 51

Code scanning

Latest scan Branch Workflow Lines scanned Duration Result
15 minutes ago master CodeQL 53.9k / 54k ⓘ 3m 56s 51 alerts

Add more scanning tools

Filters is:open branch:master

51 Open 0 Closed Tool Rule Branch Severity Sort

- Type confusion through parameter tampering Critical routes/search.ts:14 • Detected 15 minutes ago by CodeQL master
- Type confusion through parameter tampering Critical lib/insecurity.js:117 • Detected 15 minutes ago by CodeQL master
- Template Object Injection Critical routes/dataErasure.ts:78 • Detected 15 minutes ago by CodeQL master
- Template Object Injection Critical routes/dataErasure.ts:63 • Detected 15 minutes ago by CodeQL master
- Hard-coded credentials Critical (Test) test/server/verifySpec.ts:303 • Detected 15 minutes ago by CodeQL master
- Hard-coded credentials Critical (Test) test/server/verifySpec.ts:291 • Detected 15 minutes ago by CodeQL master