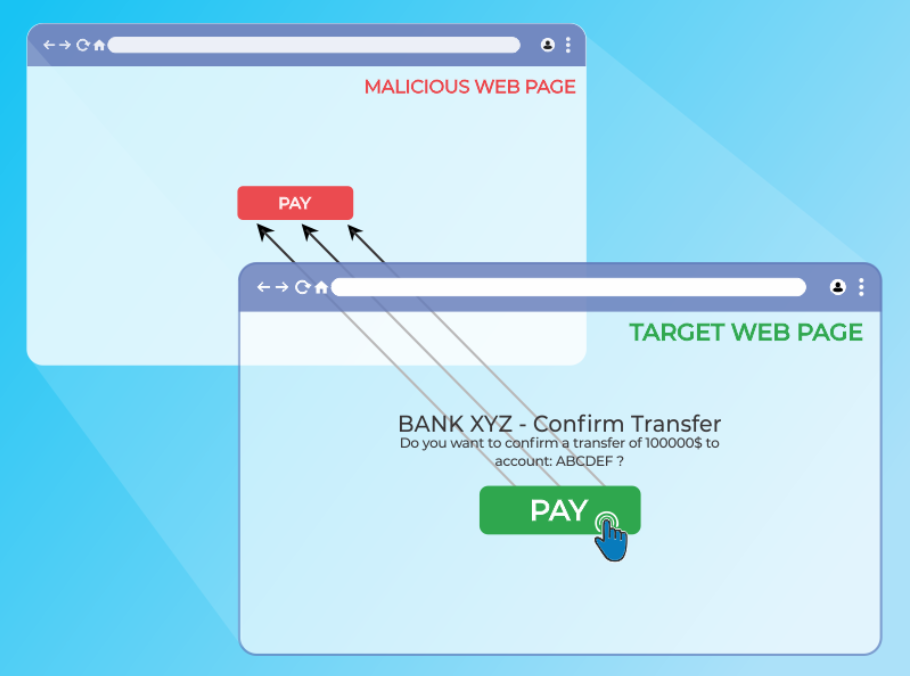
**Clickjacking Nedir?**

Clickjacking (UI redressing), kullanıcıyı bir butona veya bağlantıya tıklaması için kandırmak üzere görünmez bir iframe ya da CSS ile kaplanmış gerçek bir web sayfası yerleştirme yöntemidir. Kullanıcı gördüğünü düşündüğüyle değil, arkasındaki gizli içerikle etkileşime girer.

Gerçek dünya örneği:  
Bir sayfada “Hediye Kazan!” butonuna tıklıyorsun ama arka planda gizli bir ödeme sayfasına tıklıyorsun ve hesabından para gidiyor.

****

Frame kullanılarak, görünür buton ve görünmez iframe’in üst üste konumlandırılması.

Bu görselde saldırgan sayfa, iframe ile gerçek siteyi şeffaf şekilde yükleyip, kullanıcıyı kandırıyor.

**Nasıl Test Edilir?**

***<html>***

***<head><title>Clickjack Test</title></head>***

***<body>***

***<iframe src="http://hedef.site/secret-button" width="500" height="500"></iframe>***

***<p>Eğer iframe içeriği görünürse, hedef site clickjacking korumasına sahip değil demektir.</p>***

***</body>***

***</html>***

Bu sayfayı başka bir siteye yerleştirip tarayıcıda açarsan, iframe içeriği yüklüyorsa sitelerde X-Frame-Options ya da benzeri koruma yok demektir.

**Clickjacking Test Araçları**

* **Burp Suite Clickbandit:** GUI ile PoC (proof-of-concept) oluşturur, tıklamaları kaydetmenizi sağlar.
* **Python script’ler**: Örneğin “Clickjacking-Tester” github script’i iframe yüklenebilirliği test eder ve PoC oluşturur.
* **Chrome Uzantıları**: “Clickjacking Test” uzantısı, sayfanın iframe’e alınabilirlik durumunu kontrol eder.

**Clickjacking Türleri (Zafiyetin Farklı Senaryoları)**

Clickjacking, yalnızca kullanıcıyı tıklamaya kandırmakla sınırlı değildir. Farklı varyantları da bulunmaktadır:

* **Likejacking:** Kullanıcı aslında Facebook “Beğen” butonuna tıklarken başka bir işlem yaptığını sanır. Sosyal medya manipülasyonunda yaygındır.
* **Cursorjacking:** Kullanıcının fare imlecini başka bir yerde gösterirken, gerçek konum farklıdır. Yanlış alana tıklaması sağlanır.
* **File Upload Clickjacking:** Dosya yükleme/indirme pencereleri arka plana gizlenir. Kullanıcı zararlı dosyayı istemeden indirir.
* **Camera/Microphone Clickjacking:** HTML5 ile çalışan kamera/mikrofon izni isteme kutusu gizlenir. Kullanıcı başka bir şeye tıkladığını sanarak izni verir.

**HTTP Header Açıklamaları (Savunma Tarafı)**

**Clickjacking'e karşı alınabilecek HTTP başlıklarının detayları:**

**X-Frame-Options**

* **DENY:** Hiçbir sitede iframe'e alınamaz.
* **SAMEORIGIN:** Sadece aynı domain içerisinden iframe’e alınabilir.
* **ALLOW-FROM uri:** Sadece belirtilen alan adından yüklenebilir (eski tarayıcılarda destekli).

**Content-Security-Policy (CSP)**

***Content-Security-Policy: frame-ancestors 'self'***

Bu başlık, yalnızca kendisinin iframe’e alınmasına izin verir. Modern çözüm olarak X-Frame-Options’tan daha güçlüdür.

**Otomatik Tespit için Kullanılabilecek Araçlar**

* **Nikto:** Web sunucusunu tararken clickjacking koruması olup olmadığını kontrol eder.
* **OWASP ZAP:** Passive scan sırasında iframe yüklenmesine izin verilip verilmediğini tespit eder.
* **TestMySite GUI Tools:** Online bazı test siteleri hedef URL girerek otomatik clickjacking testi yapar:

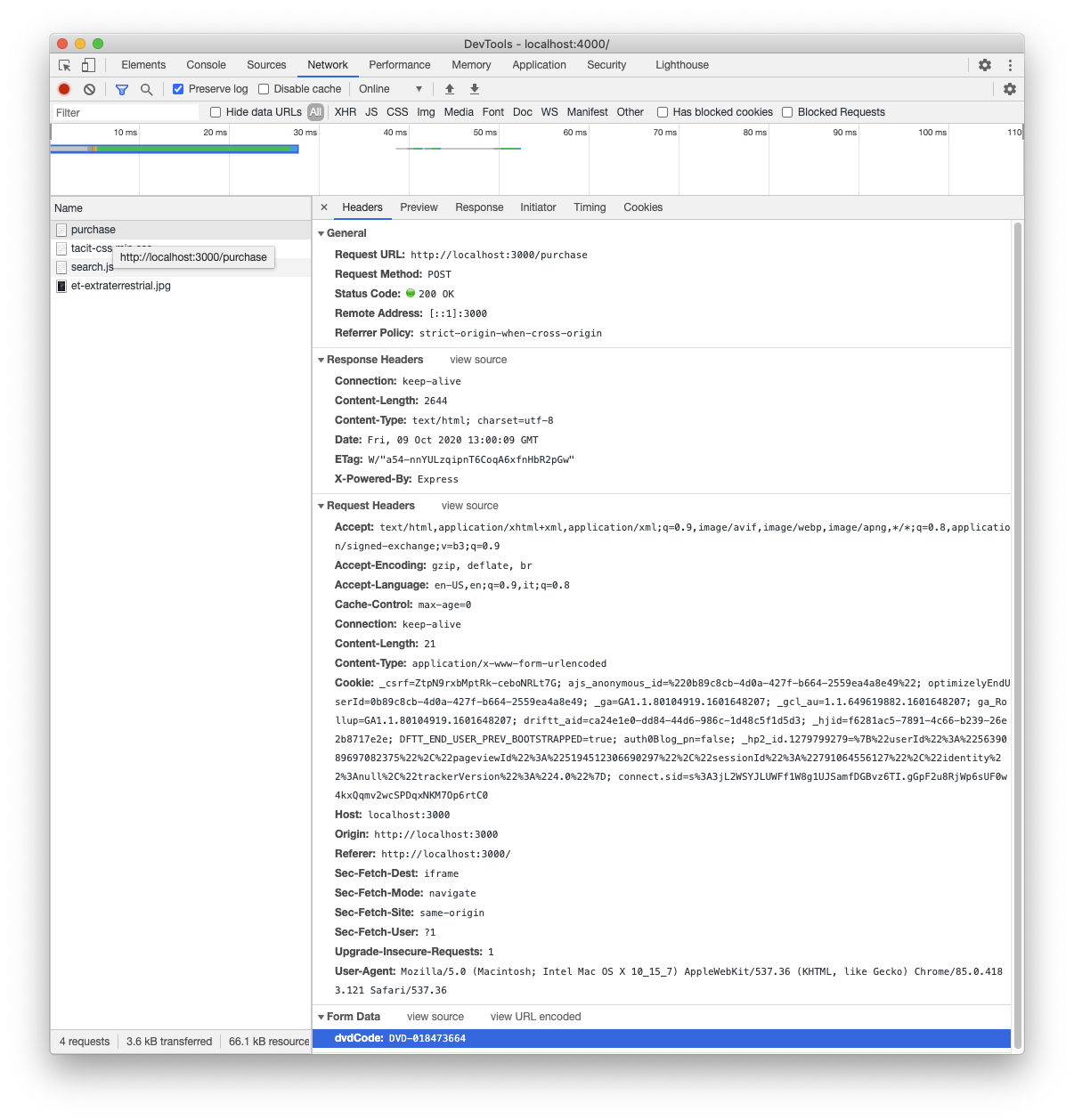
**Ne Zaman Tehlikeli Olur?**

Kullanıcı giriş yapmışsa ve arka planda önemli işlemler yapıyorsa,

Ödeme, ayar değişikliği, hesap bilgisi gibi “tek tıkla yapılabilen” işlemler varsa.

**Korunma Yöntemleri:**

* X‑Frame‑Options header kullanın: DENY veya SAMEORIGIN değeri iframe yüklenmesini engeller.
* **Content Security Policy (CSP)** frame-ancestors 'self' kuralını ekleyin.
* **Frame‑busting JavaScript** ile üst frame kontrolü yapın.
* **NoScript/ClearClick** tarayıcı eklentileriyle iframe koruması sağlanabilir.



HTTP yanıtında X‑Frame‑Options: DENY header’ı eklendiğinde sayfa iframe içinde yüklenemez.

**Clickjacking Koruması Nasıl Uygulanır?**

Apache: ***Header always append X-Frame-Options DENY***

Nginx: ***add\_header X-Frame-Options "DENY";***

PHP: ***header('X-Frame-Options: DENY');***

**Öğrencilere Tavsiyeler**

* Kendi geliştirdiğiniz test sayfasıyla iframelendirmeyi deneyin.
* Burp Clickbandit kullanarak adım adım PoC oluşturun.
* Zafiyetli sayfaları bulduğunda geliştiriciye veya site yöneticisine geri bildirim yapın (GDPR, etik).
* Koruma olmadığında kullanıcıya ne tür tehlikeler doğabileceğini düşünün.