Selective Framebusting

Seçiçi Çerçeveleme Engelleme

Sema Arık, TurkcellTeknoloji, sema.arik@turkcellteknoloji.com.tr
Bedirhan Urgun, Web Güvenlik Topluluğu, urgunb@hotmail.com

Giriş

PCI-DSS Requirements and Security Assesment Procedures (Gereksinim ve Güvenlik Değerlendirme Prosedürleri) v1.2 dokümanı[1], Protect Cardholder Data (Kart sahibi Verilerinin Korunması) başlığı altında dördüncü gereksinim olarak;

> Açık ağlar boyunca kart sahibi veri gönderiminin şifrelenmesi

zorunluluğunu getirir.

Şirketlerin ana web sayfaları online kullanıcıları ile buluştukları en önemli noktalardır. Bu sitelerde, şirketler kullanıcılarına en yeni haberleri ve gelişmeleri aktardıkları gibi, kullanıcılarından geri dönüş de alırlar. Bu doğrultuda kullanıcıların formlar doldurması ve/veya bazı servislerden faydalanması direk olarak bu site içinden sağlanır.

Problem

Ana web sitelerinin kullanıcılarından aldığı bazı veriler regülasyonlar ve genel güvenlik prensipleri düşünüldüğünde SSL üzerinden aktarılmalıdır. Bu da, HTTP üzerinden açılan ana web sitesinin (frame/iframe etiketleri ile) sağladığı formları ve verileri HTTPS üzerinden aktarması demektir.

Resim 1

Bu durum aslında HTTP ve HTTPS kaynaklarının karıştırılması anlamına gelir ve temelinde güvenlik/güven problemleri taşır. Yapılması gereken bu iki tür trafik için açık bir farklılık oluşturmaktır (ayrı pencerelerde mesela). Ancak bunun proje ve marketing bölümlerine kabul ettirilmesi her zaman mümkün değildir.

Ancak iframe/frame tabanlı çözümler ve genelinde bütün web sayfaları bir tür leeching (sömürme) saldırılarına açıktır. Direct linking/Inline Linking/Bandwith Theft [2] olarak da adlandırılan bu açık "saldırgan" sitenin içeriğinizi bulunduran sayfanızı kendi sayfanızda iframe/frame etiketleri yardımı ile barındırmasıdır.

Resim 2

Bu durumun getirdiği problemler arasında ana web sitenizin hit sayısının azalması ve belki bazı istemci taraflı saldırılar (clickjacking [3]) olabilir. Bu dokümanın geri kalan bölümü bahsedilen problemin mümkün olan en standard çözümünü bulmaya yöneliktir.

Çözüm #1 (Alternatif)

Direct linking saldırılarında en önemli çözümlerden biri Referer kontrolüdür. Ana web sitesinde uygulanabilecek bu yöntem, korunmak istenen içeriği bulunduran sayfaya gelen isteklerdeki HTTP Referer başlığının değerinin kontrol edilmesidir. Eğer bu değer beklenen (http://www.anawebsitesi.com.tr/biryol) değerden farklı ise farklı bir içerik cevap olarak gönderilebilir.

Ancak istemci tarafında kullanılan Internet'te iz bırakmayan gezinme sağlayan programlar veya bazı proxyler bu HTTP başlığının (Referer) silinmesini sağlarlar. Her ne kadar bu programları kullanan son kullanıcılar bu durumun farkında olsalar da, Referer başlığına güvenmek prensip olarak standard bir yol değildir. Ayrıca kullanıcı kaybı endişesi bu korumayı uygulamakta problem yaratabilir. Ancak saldırıların büyük bir bölümü bu basit kontrol sayesinde ortadan kalkacaktır.

Çözüm #2 (Yanlış)

Bahsedilen problem Leeching fenomeninin özel bir durumudur ve kontrol için istemci tarafında özel bir teknik geliştirilmiştir; Frame busting [4]. Genel olarak korumak istenen sayfaya aşağıdaki gibi bir javascript kodu gömülerek, sayfanın çerçeve altına alınması önlenebilir. (Safari tarayıcısında bu kodun çalışmadığı belirtilmiştir [4])

```
if(top != self)
  top.location.href = "http://www.anawebsitesi.com.tr/";
```

Genel framebusting koruması bazı durumlarda saldırganlar tarafından çalışmaz hale getirilebilirler [6]. Ama bunun da icabına bakacak basit html etiketlerini kullanmak mümkündür [7].

Yukarıdaki kod korunması istenen **ama zaten çerçevelenmiş** olan sayfada çalışmayacaktır. Bu nedenle aşağıdaki parça kontrol için daha uygundur.

```
try{
  if(top.document.domain != "www.anawebsitesi.com.tr") {
    top.location.href = document.location.href;
  }
}
catch(e){
  top.location.href = " http://www.anawebsitesi.com.tr/";
}
```

Bu şekilde seçici davranılacak ve www.anawebsitesi.com.tr dışında başka bir sitenin sayfayı çerçeve içine alması engellenecektir.

Resim 3'te gösterilen durumda çalışan bu koruma, Resim 4'te tarayıcılarda meşhur same-origin policy (aynı kaynak politikası) prensibine [5] takılıp, çökecektir.

Resim 3

Resim 4

Çözüm #3 (Yanlış)

Resim 4 gibi bir durumda seçici frambusting kontrolünü kullanmak istemek için form açılan sayfanın iframe/frame ile açılırken HTTP ile açılması, ve sadece FORM submit aşamasında HTTPS kullanılmasıdır (Resim 5). Regülasyonlar ve güvenlik açısından senaryo ile çelişmeyen bu vaziyette script'in çalışması same-origin policy'e takılmayacaktır.

Resim 5

Ancak bu düzenleme uygulama tarafında şöyle bir probleme yol açacaktır. Kullanıcı formu doldurup POST ettiğinde, sunucu tarafında eğer bir hata durumu ile kaşılaşılırsa (eksik bilgiler), sayfa HTTPS'e döndüğünden dönülen hata sayfasında selective framebusting kodu çalışacak ve aynı kaynak politikası nedeniyle kullanıcıya hatasını düzeltmek için şans tanımadan, kullanıcıyı ana websitesine yönlendirecektir.

Çözüm #4 (Eksik)

Yukarıdaki sorun selective framebusting kodumuzu biraz düzenleyerek bertaraf edilebilir.

```
if(document.location.protocol == "http:"){
  try{
    if(top.document.domain != "www.anawebsitesi.com.tr") {
      top.location.href = document.location.href;
    }
  }
  catch(e){
    top.location.href = " http://www.anawebsitesi.com.tr/";
  }
}
```

Bu şekilde HTTP protokolü kullanıldığında çalışacak kontrol, HTTPS protokolüne geçildiğinde atlatılacaktır. Ancak saldırganın Resim 6'daki gibi bir çerçeveletme kullanmasına mani olacak hiç bir kontrol yoktur.

Resim 6

Çözüm #5 (Eksik)

Tam bir çözüm için sadece istemci taraflı standard bir çözüm bulunamamıştır. Bu denemede, hem Çözüm #2'deki orjinal selective framebusting kodu, Çözüm #3'teki "HTTP sayfa HTTPS submit" metodu hem de sunucu tarafında (sunucu taraflı kod içerisinde) aşağıdaki algoritma adımları kullanılmıştır.

1. HTTP/S GET isteklerinde kullanıcıyı selective framebusting kontrolü olan normal form sayfasına
2. HTTP/S POST isteklerinde,
    2.1. İsteği işle
        2.1.1.Hata yoksa framebusting kontrolü olmayan normal bir sayfaya
        2.1.2.Hata varsa framebusting kontrolü olmayan orjinal form sayfasına

yönlendir. Daha da sadeleştirilirse;

3. HTTP/S GET isteklerinde, selective framebusting kontrolü olan normal form sayfasına
4. HTTP/S POST isteklerinde, selective framebusting kontrolü olmayan normal form veya başka sayfaya

yönlendir.

Ancak bu durumda bile saldırganın elinde sayfayı çerçeleyebileceği bir durum vardır. Saldırgan, kendi sayfasında biraz Javascript ve DOM programming ile dinamik olarak iframe oluşturabilecek, bu iframe'in içine dinamik olarak form oluşturup, korumak isteğimiz sayfaya HTTPS submit gerçekleştirebilecek, eksik bilgi girdiği için de sunucu taraflı kodumuz tarafından içinde selective frame busting kodu olmayan form sayfasını (1-2 hata mesajı ile) çerçeveleyebilecektir!

Çözüm #6 (Tam)

Çözüm #5'in üzerine GET isteğinde sunucu tarafında oluşturulacak rasgele bir token hem form sayfasında hidden bir değişken olarak isetmciye hem de oturumda (session) saklanmalıdır. Bu şekilde, HTTP/S üzerinden POST yapan kişinin saldırgan olmadığının kontrolü sunucu tarafında yapılabilecektir.

Bu da klasik bir CSRF korumasıdır ki zaten istenen bir güvenlik kontrolüdür.

Referanslar

[1] https://www.pcisecuritystandards.org/security_standards/download.html?id=pci_dss_v1-2.pdf

[2] http://en.wikipedia.org/wiki/Direct_linking

[3] http://ha.ckers.org/blog/20081007/clickjacking-details

[4] http://www.quirksmode.org/js/framebust.html

[5] https://developer.mozilla.org/En/Same_origin_policy_for_JavaScript

[6] http://crypto.stanford.edu/framebust/

[7] http://ha.ckers.org/blog/20081007/clickjacking-details/#comment-87102