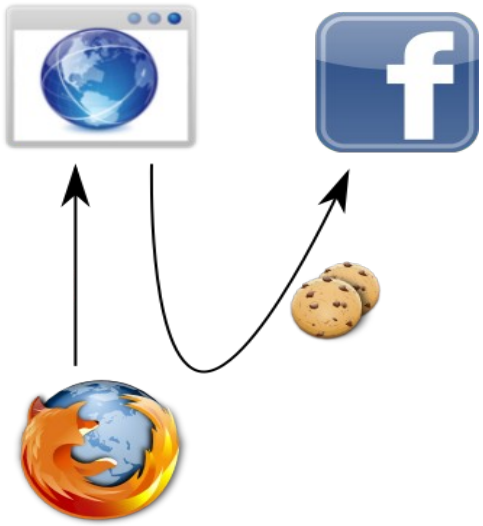
CSRF Nedir ve Nasıl SameSite ile Tamamen Önlenir?

Cross Site Request Forgery nedir ve nasıl SameSite bayrağı ile önlenir konulu bu yazı için önce csrf'nin temel nedeni olan third party cookie'den, sonra csrf 'nin tam olarak ne olduğundan ve son olarak da samesite bayrağının ne olduğu ve csrf'yi nasıl durdurulabileceğinden bahsedilecektir.

a. Third Party Cookie Nedir?

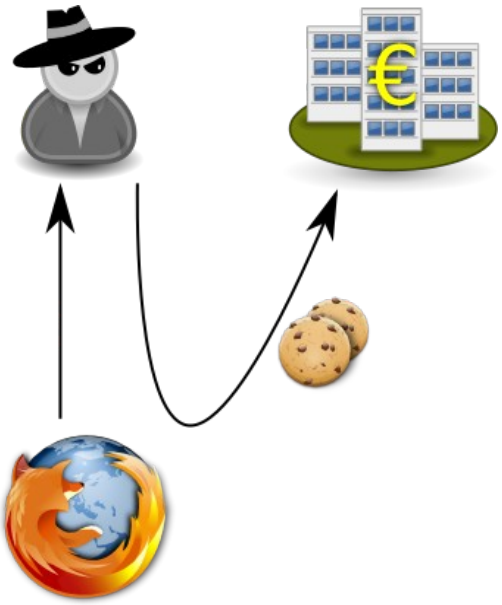
Bir web sayfasını ziyaret ettiğimizde talep ettiğimiz web sitesi başka bir web sitesinden CDN linki ile javascript kütüphanesi ya da font talep edebilir. Örneğin birçok web sitesi script'lerini ve font'larını Google sunucularından, share butonlarını ise Facebook'tan ve Twitter'dan temin etmektedirler. Web uygulamalarının bu kaynakları (font'ları, script'leri ve widget'leri) kendi içlerinde çalıştırmak için yaptığıkları http taleplerine cross-origin talepleri (yani siteler arası talep) adı verilir. Bu talepler böyle adlandırılır, çünkü bir web sitesi (yani bir origin) bir başka web sitesinden veri talep etmektedir. Bir web sitesi bir başka web sitesinden veri talep ederken veri talep edilen web sitesi için bir çereze sahipsek mevcut web sitesini görüntülerken talebin yapıldığı diğer web sitesine ait çerezimiz de talebe eklenecektir ve diğer siteye gönderilecektir. Örneğin Facebook'ta oturumumuz açık durumda olsun ve yan sekmede ise herhangi bir web sitesini görüntülüyor olalım. Görüntülüyor olduğumuz web sitesinde Facebook share butonu varsa facebook oturum çerezimiz görüntülüyor olduğumuz sayfada üçüncü taraf olan facebook'a gönderilecektir. Dolayısıyla gönderdiğimiz facebook oturum çerezlerine third party cookie denecektir. Eğer facebook'u direk görüntülüyor olsaydık aynı oturum çerezleri bu sefer first party çerezler olacaklardı. Aşağıdaki resimde görüntülüyor olduğumuz sayfadan üçüncü taraf bir siteye çerez gönderişimiz gösterilmiştir.



Not: Web sitelerine yerleştirilen share button'ları ile facebook, kullanıcıların üçüncü parti çerezlerini toplamaktadır ve böylece nereleri ziyaret ettiklerini takip edebilmektedir.

b. Cross Site Request Forgery Nedir?

Cross Site Request forgery saldırısı kullanıcının oturum açtığı bir web sitesi yan sekmedeyken saldırganın sitesine uğraması ya da saldırgandan gelen mail'i açması sonrası gelen linki tıklamasıyla (ya da tetiklemesiyle) beraber oturum açtığı web sitesinde istemediği bir eylemi gerçekleştirmesine denir. Örneğin kullanıcı bir bankacılık sitesinde oturum açmış olabilir ve yan sekmede ise saldırganın sitesine ya da mail'ine yönlenmiş olabilir. Saldırganın sayfasında ya da mail'inde img tag'ına eklenmiş form submit'leme linki ile istemeden bankacılık web sitesine bir talepte bulunabilir ve belki de saldırganın hesabına para transfer edebilir.



Bu saldırı üçüncü parti çerezler nedeniyle başarılı olmaktadır. Eğer üçüncü parti çerezler aktif olmasaydı kullanıcı saldırganın sayfasındayken ya da saldırganın mail'ini görüntülüyorken saldırganın bankacılık sitesine yaptırdığı talebe kullanıcının bankacılık çerezi eklenemeyecekti ve böylece bankacılık sitesi kullanıcıyı oturum açmamış göreceğinden bir işlem gerçekleştirmeyecekti. Dolayısıyla üçüncü parti çerezlerin kapatılması CSRF'yi öldürmektedir.

c. Cross Site Request Forgery Saldırısı SameSite Bayrağı ile Nasıl Önlenir?

Set-cookie kullanıcı çerezlerini sunucudan kullanıcıya göndermek için kullanılan bir http response başlığıdır. SameSite bayrağı ise Set-Cookie başlığının aldığı anahtar kelimelerden biridir. Syntax'ı ise şu şekildedir:

Set-Cookie: <key>=<value>; Expires=<expiryDate>; Secure; HttpOnly; SameSite="..."

Yukarıda önce çerez değişken ve değeri, sonra çerezin kullanım ömrü ve son olarak da çerezi denetim altına alan bayraklar listelenmiştir. Bayrakların açıklaması şu şekildedir:

Secure Flag

Secure flag'i ile işaretlenmiş çerezler yalnızca https trafiği olduğunda istemci ve sunucu arasında gidip gelir. Eğer istemci ve sunucu arasında iletişim http 'ye dönerse çerez gönderimi engellenir. Böylece web uygulamasını kullanan kullanıcı çerezlerinin MITM yapan kişilerce okunabilmesinin önüne geçilmiş olur.

HttpOnly Flag

HttpOnly flag'i ile web uygulamasında istemci tarafındaki javascript kodlarının kullanıcı çerezine erişimi engellenir. Böylece ileride çıkabilecek olası xss zafiyetlerine karşı çerezlerin üçüncü parti konumlara gidişi engellenmiş olur.

SameSite Flag

SameSite flag'i third party çerezlerin üçüncü taraf konumlara gidişini denetler. Mod olarak Lax ve Strict değerlerini alabilir. Lax gevşek moddur. Üçüncü taraf yerlere çerezin gidişine izin verir. Strict katı moddur. Üçüncü taraf yerlere çerezin gidişine izin vermez.

SameSite bayrağına strict değeri konarak çerezlerimizin üçüncü taraf yerlere gönderimini engellemiş oluruz. Yani çerezlerimiz böylelikle sadece görüntülüyor olduğumuz mevcut siteye gönderilebilir. Yan sekmedeki web sitelerine gönderilemez. Böylece csrf saldırılarını tamamen durdurulabilir.

Not: Third party cookie'ler tarayıcı ayarlarından tarayıcı genelinde disable edilebilir ve böylece csrf'nin önüne geçilebilir. SameSite bayrağı ise tarayıcı genelinde third part cookie'leri değil de sadece spesifik cookie'lerin üçüncü tarafa gönderimini engeller ve böylelikle daha iyi yönetilebilir bir çerez denetimi sağlar.

Ekstra

Third parti çerezler sıklıkla tarayıcı ayarlarından ve same origin policy gibi güvenlik ayarlarından engellenirler ve silinirler. Firefox tarayıcısı varsayılan olarak third party çerezleri bloklamaktadır. Third parti çerezleri bloklamak internette daha az reklam görmemizi sağlar. Çünkü gezindiğimiz sitelerdeki widget'ler , flash'lar google'a daha az bilgi gönderebilir durumda olur ve google bizden daha az sayıda bilgi sahibi olur. Böylece gizliliğimiz temin edilmiş olur ve google bize hitap eden (daha önce ziyaret ettiğimiz şeylere dair) reklamlar veremez duruma gelir. Third parti çerezleri bloklamak üstelik oturum sorunlarına da yol açmaz. Çünkü sadece first party çerezler silinirse oturum problemleri ortaya çıkar.

Kaynaklar

https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

https://www.sjoerdlangkemper.nl/2016/04/14/preventing-csrf-with-samesite-cookie-attribute/

http://whatis.techtarget.com/definition/third-party-cookie

http://www.ravelrumba.com/blog/third-party-cookies/

https://blog.appcanary.com/2017/http-security-headers.html