Brute Force & Dictionary Saldırılarını CSRF Token ile Önleme

Saldırganların hedef bir web uygulamasındaki csrf korunaksız bir html form'unu uzaktan kurbanlara çeşitli yollarla submit'lettirmesine csrf saldırısı adı verilir. Bu saldırı yoluyla kurbanların istemeden ve farkına dahi varmadan web uygulamasında çeşitli aksiyonlar alması sağlanır. Bu saldırıdan kurbanların istemediği eylemleri yaparak zarar görmesini engellemek için web uygulamalarındaki html form'ları içerisine hidden bir alan ilave edilir. Bu hidden alan csrf jetonudur ve random bir string tutar. Böylece kullanıcılar web uygulaması üzerinde fark etmeden aksiyon alamazlar ve bu yolla yapılan saldırıların büyük bir kısmı engellenir. Ayrıntılı bilgi için bkz. Paketleme İçin Gözden Geçirilecekler / İnternetten Edinilmiş Kıymetli Bilgiler / Acunetix Zafiyetler / HTML Form without CSRF Protection.

Brute Force ve Dictionary saldırılarında login paneller defalarca payload girdileriyle test edilir. Doğru eşleşmeler bulunduğundan hesap ele geçirilir. CSRF Token'lar CSRF saldırılarına karşı koruma sağlayarak kullanıcıların istemeden (ve farkına varmadan) uygulamadaki çeşitli noktalarda yer alan form'ların submit'lettirilmesinin önüne geçerken aynı zamanda eğer login form'larında da kullanılırsa temel düzeyde programlanmış brute force ve dictionary saldırısı yapan araçların işleyişini önler.

Temel düzeyde bir brute force veya dictionary saldırı aracı tanımlanan http paketine kullanıcı adı ve şifre ikililerini dinamik olarak vererek sürekli tekrarlar. Eğer bir login formunda hidden alan değeri olarak sürekli random değer alan csrf token konulursa istemciden saldırı aracı ile gönderilen ilk deneme sonrası http yanıt döndüğünde artık başka bir değerde csrf token sunulmuş olacaktır. Fakat saldırı aracı ikinci denemesinde halen aynı paketi tekrarlayacağı için aynı csrf token değerine sahip pakete yeni kullanıcı adı ve şifre ikilisini koyup sunucuya yollayacaktır. Sunucu ise expired olmuş (artık kullanılabilirliği harcanmış) csrf token'ı gördüğünde kullanıcı adı ve şifre ikilisi doğru dahi olsa http paketi geçersiz sayıp işleme koymayacaktır. Böylece temel düzeydeki brute force & dictionary saldırısı yapan araçların html login sayfalarını istismar ederek hesap ele geçirmesi önlenebilecektir. Eğer gelen http yanıtına göre dinamik olarak csrf token değerini alan ve ona göre sonraki denemelerinde göndereceği http paketini oluşturan araçlar kullanılırsa bu önlem aşılacaktır ve hesap ele geçirme halen mümkün olacaktır. Fakat login panellerde csrf token ile bir nebze güvenlik seviyesi arttırıldığından script kiddies'lerden (ergenlerden) sistemler korunabilecektir.

> Olay:
>
> (*) Birebir deneyimlenmiştir.
>
> **Hydra Tool'u ve Kullanımı** konulu includekarabuk'te yazmaya hazırlandığın blog makalesinde **Uygulama** başlığı altında bahsetmek üzere brute force yapılacak login paneli ararken DVWA'nın ana login sayfasına brute force yapmayı denediğinde bir şey fark ettin. POST edilen değişkenler içerisinde kullanıcı adı ve şifre olduğu gibi user_token adlı ekstradan bir parametre ve 03u43289432894 gibi bir değer göndermekteydin. DVWA ana login sayfasını her refresh'lediğinde ve login panele sağ tık öğeyi denetle yapıp user_token değerine geldiğinde user_token'ın aldığı değerin değiştiğini gözlemledin. Aynı zamanda DVWA ana login sayfasını her refresh'lediğinde bu parametre değeri burpsuite proxy'de gözlemlediğin üzere değişmekteydi. Dolayısıyla hydra'ya mevcut user_token parametre ve değerini koyduğunda bir sonraki yaptığı denemede sunucunun istediği user_token'ı değil de eskisini, sürekli eskisini vereceği için boşa çabalamış olmaktaydı. Yani bu token ile görüldüğü üzere hydra gibi temel araçların brute force yapması önlenebiliyor. Daha dinamik çalışan brute force araçları (örneğin gelen http yanıtını alıp o yanıt paketini okuyup token'ı dinamik olarak ekleyip yeni denemesini onla yapması gibi) ise bu önlemi aşabilir. Ama

CSRF token ile belli ölçüde güvenlik kademesi arttırılabildiğinden script kiddies'lerden (bir nevi ergenlerden) sistemler korunabilir.

Aşağıda CSRF korunaksız html form'ları açıklığının iki raporlanma hali gösterilmektedir:

HTML Form without CSRF Protection // CSRF Token ile zafiyet kapanır
HTML Login Form without CSRF Protection // CSRF Token ile zafiyet kapanır

Korunaksız HTML form ile kastedilen, uygulama içi herhangi bir html form'da csrf token'ın olmayışıdır. Korunaksız HTML Login form ile kastedilen ise uygulama içi herhangi bir html form değil de sadece html bir login form'da csrf token olmayışıdır. Bazı açıklık tarayıcıları csrf korunaksız form gördüğünde her durumda birinciyi (*HTML Form without CSRF Protection*) raporlarken bazı açıklık tarayıcıları csrf korunaksız bir login form'u gördüğünde birinci yerine daha spesifik bir tanım olan ikinciyi (*HTML Login Form without CSRF Protection*) raporlamaktadır.

"*HTML Form without CSRF Protection*" zafiyeti ile yapılabilecekler,

a) Uygulama içi html form'larının çeşitliliği ve yetenekleri doğrultusunda her şey
b) saldırganların kurbana aynı saldırı yoluyla yüzlerce, belki binlerce örneğin aynı form'u submit'lettirmesi ve bu, birden fazla kurbana yapıldığında hedef web uygulaması sunucu meşguliyetinin arttırılması veya veritabanının şişirilmesi
…

"*HTML Login Form without CSRF Protection*" zafiyeti ile yapılabilecekler,

a) saldırganların web uygulamasındaki html login formuna doğrudan "basit" düzeydeki otomatize araçlar ile brute force & dictionary saldırısı düzenleyebilmeleri,
b) saldırganların çaldıkları bir hesabı kurbana login formu üzerinden uygulattırıp oturum açtırarak kurbanı töhmet altında bırakabilmeleri
c) saldırganların kurbana illegal (zararlı) payload'larla login formunu submit'lettirmesi ve hedef sunucu tarafınca kurbanın kötü niyetli kişi zannıyla kayıtlara geçmesi,
d) saldırganların kurbana aynı saldırı yoluyla yüzlerce, belki binlerce örneğin aynı form'u submit'lettirmesi ve bu, birden fazla kurbana yapıldığında hedef web uygulaması sunucu meşguliyetinin arttırılması veya veritabanının şişirilmesi
…

Kaynaklar

https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/cross-site-request-forgery-in-login-form/
https://www.acunetix.com/vulnerabilities/web/html-form-without-csrf-protection/