Oturum Yönetimi

Amaç

Aşağıdakileri sağlamak

- kimliği doğrulanmış kullanıcıların açtıkları oturumları ile sağlam ve kriptoğrafik olarak güvenli bir bağları olması
- yetkilendirme kontrollerinin uygulanması
- tekrarlama, istek sahteciliği ve araya girme saldırıları gibi yaygın web saldırılarının önlenmesi

Etkilenen Platformlar

Hepsi.

İlgili COBIT Konuları

PO8 – Bütün bölümler gözden geçirilmeli.
PO8.4 – Gizlilik, kişisel mülkler ve veri akışı

Açıklama

Kalın istemci (thick client) uygulamaları, programın yaşam boyunca lokal durum bilgilerini, işletim sistemi tarafından sağlanan global, yığıt ve yığın değişkenleri gibi hafıza bölgelerinde tutarlar. Web uygulamalarında sunucu, istemci isteklerine cevap olarak sayfaları sunar. Dizayn olarak, belirgin bir durum bilgisi olmadığından, sunucu geçmişte sunduğu sayfalar hakkında herşeyi unutmakta serbesttir.

Bu durum broşür veya resim gibi içerikleri sunarken iyi çalışır ama formları doldurmak veya internet bankacılığı uygulamaları gibi birbirinden ayrı tutmanız gereken çok kullanıcınız olması gibi önemli işleri yapmak istediğinizde iyi çalışmaz.

Web sunucuları kişisel kullanıcıların isteklerini cookiede biricik bir “oturum”a kriptoğrafik değer tutarak kullanıcılara “durum bilgili” bir his veren J2EE veya ASP.NET

gibi uygulama çatıları ile desteklenirler. Kullanıcı isteklerini biricik oturum bilgileri ile kısıtlamak ve idame ettirmek web güvenliği için kritiktir.

Bu kılavuz kullanıcılarının çoğu gömülü oturum yönetim kabiliyetleri olan uygulama çatıları kullanıyor olsalar da, Perl CGI gibi dilleri kullanıcılarına bu tür kabiliyetler sunmaz. Bu kullanıcılar kendi oturum yönetim kodlarını kendileri yazmak gibi bir dezavantaj ile karşı karşıyadırlar. Bu gerçeklemeler (implementations) çoğu zaman zayıftırlar ve kırılabilirler. Yapılan diğer bir hata da, güçlü bir oturum yönetim yapısının üzerine daha zayıf bir yapının yazılmasıdır. Teorik olarak kriptoğrafik açıdan güvenli bir oturum yönetim yapısı yazma mümkündür, ki bu bölümün ana konusu budur. Ancak, biz ölümlüler için yeterli bir oturum yönetimi yapısı içeren bir uygulama çatısı kullanmanın önemi ne kadar vurgulansa azdır. Bu tür yapıları tekrar yazmanın hiç bir gerçek değeri yoktur.

J2EE, PHP, ASP ve ASP.NET gibi uygulama çatıları bir çok alt seviye oturum yönetimi detaylarını ve sunucu seviyesinde değil de programlama seviyesinde iyi düzeyde kontrol sağlarlar. Mesela, ASP.NET, her sayfada bir gizli alanda oluşturulan karmaşıklaştırılmış oynamalara dayanıklı “view state” mekanizması kullanır. View state hassas olmayan değişkenlerin, web kontrol yollarının ve bunun gibi şeylerin gönderilmesi için kullanılabilir. Programlama yolları kullanılarak, değişkenlerin view state’a girip girmemesini kontrol etmek mümkündür, özellikle aynı uygulamanın farklı sayfalarında bir kontrol’ün durum bilgisinin tutulmasına gerek yoksa. Eğer kullanıcıya hassas veri yolluyorsanız view state’in kriptolamak ta mümkündür, ancak bu tür değişkenlerin sunucu tarafında tutulması daha iyidir. Bu indirme süresininden kısar, kullanılan bandwith’i düşürür ve çalıştırma zamanını iyileştirir. Dikkatli yazılmış bir view state yönetimi iyi ayarlanmış uygulama ile kötü performanslı bir uygulama arasındaki farktır.

En iyi yöntemler

En iyi yöntem tekerleği yeniden icat etmek değil, iyi bilinen ve sağlam bir oturum yöneticisini kullanmaktır. Popüler web uygulama çatılarının çoğu uygun bir yapı sağlarlar. Ancak, çoğunlukla ilk versiyonları belirgin zayıflıklar içerir. Daha sağlam ve kriptoğrafik açıdan daha güvenli olacağından seçtiğiniz teknolojinin her zaman en yeni

versiyonlarını kullanın. Bunun böyle olduğunu anlamak için de favori arama moturununuzu kullanın.

Uygulama durum bilginizi nerde tuttuğunuzu dikkatlice düşünün:

- Yetkilendirme ve rol bilgileri sadece sunucu tarafında tutulmalıdır
- Denetlemeden geçtiği ve yetkilendirme kontrolleri olduğu sürece Navigasyon bilgisinin URL'de olması nerdeyse tamamen kabul edilebilir.
- Sunum bayrakları (theme veya kullanıcı dili) cookie'lerde kalabilir.
- Form verileri gizli alanları içermemelidir – eğer gizli ise korunmalı ve sunucu tarafında saklanmalı demektir. Ancak, gizli alanlar sıra (sequence) ve kaba kuvvet pharming saldırılarına karşı koruma olarak kullanılabilirler (hatta kullanılmalıdırlar).

Bir çok sayfadan oluşan formlardan gelen veri tekrar kullanıcıya iki durumda gönderilebilir:

- Veri oynanmasını engellemek için bütünlük kontrolleri varlığında
- Her form gönderiminde veya en azından gönderim sürecinin en sonunda veri denetlendiğinde
- Uygulama hassas bilgileri (sunucu tarafı ile ilgili kimlik ve rol bilgileri) kesinlikle istemciye gitmemelidir. Bunlar bir oturumda ve sunucu tarafında tutulmalıdırlar.

Eğer şüpheye düşerseniz, risk almayın ve veriyi sunucu taraflı oturumda tutun.

Yanılgılar (Yanlış Bildiklerimiz)

Oturumlar özellikle kolay programlanabilen durum bilgisiz (stateless) sunucu taraflı bileşenleri seçen Java tabanlı bazı programcılar yüzünden haksız bir kötü üne sahiptirler. Ancak, durum bilgilerinin bir yerlerde saklanması ve bunun sunucu tarafında bir şekilde sağlanması gerekliliklerinden, bu durum güvenlik ve performans perspektiflerinden yanlıştır. Alternatif ise, bütün durum bilgilerinin her istekte tutulmasıdır. Ancak bu durum çok fazla gizli alanların kullanılasına ve extra veritabanı sorgularına ve bunun da sonuç olarak tekrar oynama (replay) ve istek sahteciliği saldırılarına ve kod karmaşasına yol açmaktadır.

Diğer bir ortak yanılgı oturum bilgilerinin değerli sunucu kaynaklarını harcamasıdır. Bu RAM'lerin kısıtlı olduğu zamanlar için doğruydu ama bugün değildir. Gerçektende, istemciye oturum bilgisini göndermek ve sonra kod çözmek (decode), serı halini açmak, kurcalama kontrolleri uygulamak ve en son olarak da her istekte veriyi denetlemek sunucu kaynaklarını oturum bilgisini sunucu tarafında tutmaktan daha fazla harcar.

Serbest Oturum Üretmek

Bir çok we uygulama çatıları yeni isteyene eğer önceden oluşturulmamış ise yeni bir oturum ID'si üretir. Bu "serbest oturum üretimi" olarak adlandırılır. Bu durum palıkçılık ve yetkilendirme yetersilikleri ile birleştiğinde yıkıcı saldırılara yol açabilir.

Saldırıya açık olup olmadıgınızı anlama (yolları)

- Korunan bir sayfaya veya uygulamada derindeki bir aksiyona ulaşmak için yeni bir tarayıcı açın
- Eğer yeni bir oturum IDsi üretilir ve sayfa çalışırsa, yetkilendirme kontrolleri oturum değişkenin geçerliliği ile alakalı olarak yanlış bir varsayımda bulunuyor demektir.

Kendinizi Koruma Yolları

- Bir kullanıcı için yeni bir oturum nesnesi başlatıyorsanız, "çıkış" (logged off) durumunda ve herhangi bir rol verilmemiş olduğundan emin olun.
- Korunan bir sayfanın veya aksiyonun, sayfayı servis etmek gibi önemli işleri yapmadan önce kimlik doğrulama durumunu ve yetkilendirme rolünü kontrol ettiğinden emin olun.
- Bütün korunmayan sayfaların hizmet dışı bırakma saldırılarını önlemek için kaynakları mümkün olabilecek minimum düzeyde kullandığına ve uygulamanın korunan bölümleri ile alakalı bilgi ifşa etmediğinden emin olun.

Korunmasız Oturum Değişkenleri

Bazı çatılar web sunucunun paylaşılan disk alanını oturum verisini saklamak için kullanırlar. Özellikle, PHP Unix sistemlerde /tmp ve Windows'da c:\windows\temp

yollarını kullanır. Bu alanlar oturum verisi için koruma sağlamazlar ve web sunucu paylaşılıyorsa veya ele geçirilmişse uygulamanın ele geçirilmesine yol açarlar.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Uygulama çatısının yapılandırmasını inceleyin
- Oturumu hafızada mı, disk de mi veya bir veritabanında mı tutuyor?
- Bir veritabanında veya diskde tutuyorsa oturum verilerini başka kimler okuyabiliyor?

Kendinizi Koruma Yolları

- Uygulama sunucunun her istemci ve uygulama için özel geçici dosya alanlarını kullandığından emin olun
- Bu mümkün değilse, oturum verileri şifrelenmeli veya sadece hassas olmayan bilgileri tutmalıdır

Sayfa ve Form Tokenları

Sayfaya özel tokenlar veya "nonce"lar istemci istekleri ile uğraşıldığında belli bir seviyeye kadar kimlik doğruluğunu sağlamak adına oturuma özel tokenlar ile kullanılabilirler. Aktarma seviyesi güvenlik mekanizmaları ile kullanıldığında, sayfa tokenları belirli bir oturumda en son sayfayı isteyen kullanıcının oturumun diğer sonundaki kullanıcı ile aynı kişi olduğuna emin olunmasına yardım eder. Sayfa tokenları çoğunlukla cookielerde veya sorgu dizgilerinde (query strings) kullanılırlar ve tamamen rasgele olmalıdırlar. Sunucu tarafında aralarında bağ kurup, sayfa tokenlarını kullanarak oturum token bilgilerini istemciye göndermeyi de tamamen önleyebilirsiniz. Bu teknik oturum tokenlarının kaba kuvvet saldırılarında dirençlerini de arttıracaktır.

Saldırıya açık olup olmadığınızı anlama (yolları)

Uygulamanız:

- Geri butonunun gizli olmasını gerektiriyor mu?
- "Önceden belirlenmiş" oturum saldırılarından müzdarip mi?

Kendinizi Koruma Yolları

- Form nonce'ı veya kriptoprafik olarak güvenli bir sayfa ile gizli bir alan kullanın
- Nonce sayfa veya form tekrar gönderimini engelleme için gönderildikten hemen sonra aktif listeden silinmelidir

Zayıf Oturum Kriptoğrafik Algoritmaları

Eğer bir oturum yöneticisi tahmin edilebilir tokenlar üretiyor ise saldırganın uzak bir kullanıcının oturum bilgisini yakalamasına gerek yoktur – bu bilgiyi tahmin edebilir ve büyük ihtimalle şansız bir kurban bulabilirler. Oturum tokenları biricik, tahmin edilemez ve geri mühendisliğe dirençli olmalıdır.

Saldırıya açık olup olmadığınızı anlama (yolları)

- 1000 oturum IDsi üretin ve tahmin edilir olup olmadıklarına (grafiğini çizmek bu noktada faydalı olabilir) bakın
- Oturum yöneticisinin kaynağını oturum IDlerinin nasıl üretildiğini araştırın. Oturum IDleri yüksek kalitede rasgele kaynaklar kullanılarak üretilmelidirler.

Kendinizi Koruma Yolları

- Tokenı oluşturmak için güvenilir bir rasgele sayı üreteci kullanılmalıdır (mesela yarı-gerçek rasgele sayı üreteçleri, Yarrow, EGADS, v.b. gibi).
- Ek olarak, daha fazla güvenlik için, oturum tokenları korsanlık ve geri oynama saldırılarını önlemek için bir şekilde HTTP istemci bilgisine (oturum IDsi veya IP adresi) bağlanmalıdır .

Bu kısıtlamaları sağlayan mekanizmalar için verilebilecek örnekler üretilen her sayfa için biricik sayfa tokenlarının veya sunucu tarafında oturum tokenlarının kullanılmasıdır.

Uygun Anahtar Uzayı

Kriptoğrafik olarak güvenli algoritmalar bile eüer anahtar uzayı yeteri kadar geniş değilse tahmin edilebilir aktif oturum tokenları üretebilirler. Saldırganlar tokenın anahtar uzayını bir çok olasılığı otomatik kaba kuvvet betikleri kullanarak öğütürler (grind).

Oturum Token Entropisi

Oturum tokenları mümkün olan en fazla karakter setini kullanmalıdırlar. Eğer bir oturum tokenı diyelim ki 8 karakter ise bunun 7 biti efektiftir ve anahtar uzunluğu 56 bittir. Ancak karakter seti sadece tam sayılardan oluşursa bu 4 bit ile sağlanabilir ve sadece 32 bitlik bir anahtar uzayı verir. İyi bir oturum tokenı büyük küçük harfleri de dahil edecek şekilde mümkün olan maximum karakter setini kullanmalıdırlar

Oturum Zaman Aşımı

HTTP sunucularında zaman aşımına uğramayan oturum tokenları saldırgana sınırsız kaba kuvvet şansı verir. Bir örnek bir çok e-ticaret sitesindeki "Beni Hatırla" opsiyonudur. Eğer bir kullanıcının cookie dosyası yakalanırsa veya kaba kuvvete maruz bırakılırsa, saldırgan bu oturum bilgisini kullanarak o kullanıcının web hesabına erişim elde eder. Bu problem bir çok kullanıcının aynı bilgisayarı kullandığı paylaşılan ortamlarda daha da tehlikelidir. Ek olarak, oturum tokenları potansiyel olarak vekil sunucularda (proxy) kaydedilir ve ara belleğe alınırlar (cache). Eğer bir saldırgan bu vekillere erişim hakkı kazanırsa, HTTP sunucularda oturum tokenları zaman aşımına uğramadığından bu bilgiler tekrar kullanılabilirler.

Saldırıya açık olup olmadığınızı anlama (yolları)

Boşta Kalma "Koruması"

- Uygulama meta-refresh veya buna benzer Javascript hileleri ile boşta kalan oturumu sonlandırmaya çalışıyor mu? Eğer böyle ise (tek başına) uygulama saldırya açıktır.

Beni Hatırla?

- Uygulama "beni hatırla" özelliğine sahip mi? Eğer öyle ise uygulama saldırya açıktır.

Hatalı boşta kalma zaman aşımı

- Uygulamaya giriş yapın
- Öğle yemeğine gidin

- Uygulamayı kullanmaya çalışın. Çalıştı mı? Eğer öyle ise uygulama saldırya açıktır.

Kendinizi Koruma Yolları

Yüksek derecede korunan uygulamalarda boşta kalma zamanını 5 dakika ile düşük riskli uygulamarda 20 dakikadan fazla olmyacak şekilde ayarlayın.

Yüksek derecede korunan uygulamalarda:

- Boşta kalma durumunu engelleyici mekanizmalar yazmayın
- "beni hatırla" özelliğini yazmayın (kullanmayın)

Oturum Tokenlarının Yeniden Oluşturulması

Oturum korsanlığı ve kaba kuvvet saldırı risklerini azaltmak için, HTTP sunucu mütemadiyen tokenları sonlandırıp yeniden üretebilir. Bu da tekrar oynama ve kaba kuvvet saldırılarının fırsat penceresini kısaltacaktır.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Uygulamanızda uzunca bir oturum açın
- Her önemli işlemden (transaction) once ve sonra Oturum ID'sini kaydedin
- Eğer oturum ID'si hiç değişmiyorsa risk altında olabilirsiniz.

Kendinizi Koruma Yolları

Bu kontrol yüksek derecede korunan siteler için uygundur. Token yeniden oluşturulması

- belirgin önem derecesine sahip işlemlerden önce
- belli bir istek sayısından sonra
- Zamanın bir fonksiyonu olarak, diyelim li 20 dakikada bir.

durumlarında gerçekleştirilmelidir.

Oturum Sahteciliği/Kaba Kuvvet Yakalama ve/veya Kilitleme

Bir çok web sitesi şifre tahminlerinde yasaklara (hesabı kilitleyebilir veya IP adresini kısıtlar) sahiptir, ancak saldırgan çoğu zaman meşru bir URL veya cookie içindeki yüzlerce veya binlerce oturum tokenını sunucunun hiç bir şikayeti olmadan deneyebilir.

Bir çok saldırı tespit sistemleri bu tip saldırılara aktif olarak bakmaz ve ayrıca penetrasyon testleri e-ticaret sistemlerinde bu zayıflığa yoğunlaşmazlar.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Oturum IDsini değiştirebileceğiniz HTTP yakalayan bir vekil kullanın
- Eğer uygulama kapsamı giriş yapılmış halde ise, uygulama çatısı hatalı ve kullanılmamalıdır.

Kendinizi Koruma Yolları

- Hiç bir zaman kullanılmayan ama belli bir token aralığını deneyen saldırganları farketmek için "bubi tuzağı" oturum tokenları kullanmayı düşünebilirsiniz.

Alınabilecek aksiyonlar:

- Kaynak IP'yi yavaşlatmak veya kısıtlayın. (bu problemlere yol açabilir çünkü her gün artan sayıda ISP giderleri kısmak için transparan vekilleri kullanmaktadır. Bu nedenle, her zaman "proxy_via" başlığını kontrol edin)
- Farkettiğinizde bir hesabı kilitleyin. (ki bu durum bir kullanıcı için potansiyel bir hizmet dışı bırakma saldırısına dönüşebilir)
- Anomali saptama teknikleri eğer kimliği doğrulanmış bir kullanıcı yetkilerini arttırmak amacı ile kendi tokenlarını değiştirdiğinde de çalışabilir.
- Bu tür korumalarda kullanmak için, mod_dosevasive ve mod_security Apache web sunucu modülleri mevcuttur. mod_dosevasive Hizmet dışı bırakma saldırılarının etkilerini azaltmak için kullanılsa bile diğer bazı amaçlar için de kullanılabilir.

Oturum Tokenlarının Aktarılması (Transmission)

Eğer oturum token ağ içinde aktarılırken yakalanırsa, bir web uygulaması tekrar oynama veya korsanlık saldırılarına kolayca maruz kalır. Durum mekanizma tokenını korumak için tipik web şifreleme teknolojileri SSLv3 ve TLSv1 protokollerini içerirler ama sadece bunlardan ibaret değildirler.

Saldırıya açık olup olmadığınızı anlama (yolları)

Casus programlar (Spyware), virüsler, ve truva atlarından (Trojans) dolayı bütün Internet tarayıcıları potansiyel olarak saldırıya açıktırlar.

Kendinizi Koruma Yolları

Oturum IDsini, IP adresi ve kullanıcının istek başlığında bulunan diğer özniteliklerinin kriptoğrafik özeti (attributes) ile ilişkilendirin. Kriptoğrafik özet değişirse ve birden fazla öznitelirkler değişirse büyük bir ihtimaldir ki bir önceden belirlenmiş oturum saldırısı gerçekleşmektedir.

Çıkış İşlemi Sırasında Oturum Tokenları

Internet ulaşım noktalarının ve palyaşılan bilgisayar ortamlarının artmasıyla oturum tokenları için yeni bir risk ortaya çıkmıştır. Bir tarayıcı oturum cookiesini sadece koştuğu süreç parçacığı (thread) sonlandığında siler. Bir çok Internet ulaşım noktası aynı tarayıcı iş sürecini korur ve kullanır.

Saldırıya açık olup olmadıgınızı anlama (yolları)

Uygulamada “Çıkış” işlemini yapın

- Bütün oturum ile alakalı olmayan değişkenlerin yok edildiğini görmek için cookieleri kontrol edin

Kendinizi Koruma Yolları

Kullanıcı çıkış işlemini gerçekleştirdiğinde

- Oturumu yok edin
- Oturum cookielerinin üzerine yazın

Oturum Korsanlığı

Saldırgan sunucu üzerinde geçerli olan bir oturum tokenı yakaladığında veya oluşturduğunda, diğer bir kullanıcıyı taklit edebilir. Oturum korsanlığı uygulamada yeterli anti-korsanlık kontrolleri bulundurduktan sonra bir parça giderilebilir. Bu kontrollerin seviyesi organizasyonunuzun risk anlayışı veya müşteri bilgilerinize bağlı olmalıdır: Örnek olarak, bir online bankacılık uygulaması sinema gösterim zamanlarını gösteren uygulamadan daha fazla dikkat gerektirir.

Korsanlığı en kolay yapılan uygulama özellikle zaman aşımı bulunmayan URL tabanlı oturum tokenları kullanandır. Paylaşılan bilgisayarlarda bud aha da tehlikelidir çünkü özellikle Internet kafelerde veya herkese açık Internet ulaşım yerlerinde ara belleki (cache) temizlemek veya gezinim geçmişini silmek nerdeyse imkansızdır.

Saldırıya açık olup olmadığınızı anlama (yolları)

Casus programlar (Spyware), virüsler, ve truva atlarından (Trojans) dolayı bütün Internet tarayıcıları potansiyel olarak saldırıya açıktırlar.

Kendinizi Koruma Yolları

- Kullanıcılara uygulamanızdan çıkmalarına yarayan bir metod sağlayın. Çıkış işlemi bütün oturum durumunu temizlemeli ve silmeli veya kalan bütün cookieleri geçersiz kılmalıdır.
- Kalıcı cookieler için bir günü geçmeyecek kısa zaman aşımları kullanın veya tercihen kalıcı cookieler kullanmayın.
- Oturum tokenlarını URL'lerde veya kolayca oynanabilecek veri giriş noktalarında saklamayın.

Oturum Doğruluğu Saldırıları

Yapılan en sık hatalardan bir tanesi kısıtlanan bir fonksiyon veya bir veri erişimi öncesi yetkilendirme kontrolü yapmamaktır. Kullanıcının oturum tokenı olması uygulamanın her yerini kullanmasına veya her veriye ulaşmasına izin vermez.

Özellikle utanç verici bir gerçek hayat örneği bir çok Avusturalya şirketinin elektronik olarak çeyrek dönemlik vergi dönüşlerini gönderdikleri Avusturalya Vergilendirme Ofisi'nin GST web sitesidir. AVO kimlik doğrulama yolu olarak istemci taraflı sertifikalar kullanır. Kulağa güvenli geliyor, değil mi? Ancak, bu site ilk olarak URL'lerinde ABN (şirketler için sosyal güvenlik numarasına benzer biricik bir numara) kullanıyordu. Bu numaralar gizli değildir ve rasgele üretilmezler. Bir kullanıcı bunu çözdü ve diğer bir şirketin ABN numarasını denedi. Şaşırarak olayın çalıştığını izledi ve diğer şirketin detaylarını görebildiğini anladı. Daha sonra bütün verileri alacak bir betik yazdı ve bilgileri ve AVO'nun web sitesinde çok ciddi bir açık olduğunu ilgili şirketlerin e-mail adreslerine yolladı. 17,000'den fazla organizasyon e-mail almıştı.

Saldırıya açık olup olmadığınızı anlama (yolları)

Casus programlar (Spyware), virüsler, ve truva atlarından (Trojans) dolayı bütün Internet tarayıcıları potansiyel olarak saldırıya açıktırlar.

Kendinizi Koruma Yolları

Her zaman o anda giriş yapmış kullanıcının veriye erişime, veriyi yenilemeye veya silmeye veya bazı fonksiyonlara erişime yetkili olduğunu kontrol edin.

Oturum Denetleme Saldırıları

Bütün diğer veriler gibi, oturum değişkenin de doğru formda olup olmadığı, herhangi beklenmedik bir karakter içermediği ve geçerli olduğu kontrol edilmelidir.

Yazar uyguladığı bir penetrasyon testinde, oturum nesnelerini null byte kullanarak kesebildiğini ve oturum yönetici kodundaki bir hatadan dolayı en kısa dizginin uzunluğu ile karşılaştırdığını gördü. Ve böylece, bir karakterlik geçerli bir oturum değişkeni karşılaştırmayı geçti ve testçiye oturum yönetimini kırma şansını verdi. Diğer bir testte ise oturum yönetimi kodu her karaktere izin veriyordu.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Oturum IDsi ile oynamak için yakalayacı bir HTTP vekil kullanın.
- Eğer uygulama kapsamı giriş yapılmış halde ise, uygulama çatısı hatalı ve kullanılmamalıdır

Kendinizi Koruma Yolları

Her zaman o anda giriş yapmış kullanıcının veriye erişime, veriyi yenilemeye veya silmeye veya bazı fonksiyonlara erişime yetkili olduğunu kontrol edin. (HATA, bu söz bu bölümün koruma yolu olamaz. Koruma yolu şöyle olacaktı: Oturum bilgilerinde kullanılan formu her zaman kontrol edin ve oturum yönetimi bilinen en yeni uygulama çatılarını kullanın. Yani kendi oturum yönetiminizi yazmayın.)

Önceden Belirlenmiş Oturum Saldırıları

Bir saldırgan uygulama çatınızın özelliklerini kullanarak ya geçerli yeni bir oturum IDsi üretecek veya önceden geçerli bir oturum ID'sini belirleyerek erişim kontrollerini geçmeye çalışacaktır.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Uygulamanızın herhangi bir sayfayı ziyaret ettiğinizde yeni bir oturum ID'si üretip üretmediğini test edin.
- Uygulamanızın oturum ID'sini kalıcı olmayan cookie bölümü hariç herhangi bir yerden kabul edip etmediğini kontrol edin. Mesela, eğer PHP kullanıyorsanız cookieden geçerli bir PHPSESSIONID alın ve URL'ye aşağıdaki gibi ekleyip yollayın;

  http://www.example.com/foo.php?PHPSESSIONID=xxxxxxx
- Eğer bu çalışırsa, uygulamanız belli bir risk altındadır ancak bu risk eğer oturum ID'si sonlandıktan veya zaman aşımına uğradıktan sonra da kullanılabiliyorsa daha fazladır.

Kendinizi Koruma Yolları

Çatınızın oturum ID'sini sadece cookie değerinden kabul ettiğinden emin olun. Bu uygulama çatınızın varsayılı davranışını değiştirmenizi veya oturum yöneticisinin üzerine yazmanızı (override) gerektirebilir.

Tek bir tarayıcıyı tek bir oturuma bağlamak için oturum belirleme kontrollerini (diğer bölüme bakınız) kullanın.

Geçerli oturumun giriş yapmış oturum olduğunu varsaymayın – oturum yetkilendirme durumunu gizli tutun ve her sayfada veya her giriş noktasında yetkilendirmeyi kontrol edin.

Oturum Kaba Kuvvet Saldırıları

Bazı e-ticaret siteleri oturum ID'leri için ardışık sayıları veya kolayca tahmin edilebilir algoritmaları kullanırlar. Bu sitelerde, başka birisi olmak için oturum ID'sini başka birine değiştirmek kolaydır. Genellikle, kullanıcılara sağlanan fonksiyonlar kişisel gizlilik ve sahtecilik olaylarına yol açarlar.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Bir tarayıcıda geçerli bir oturum açın
- Brutus gibi oturum kaba kuvvet aracı kullanarak diğer açtığınız oturumu ele geçeirebilip geçiremediğinizi test edin.

- Eğer Brutus oturuma devam edebiliyorsa, uygulama çatınız daha iyi bir oturum ID entropisine ihtiyaç duyuyordur.

Kendinizi Koruma Yolları

- Kriptoğrafik olarak sağlam bir token oluşturma algoritması kullanın. Kendi algoritmanızı yazmayın ve algoritmayı güvenli bir şekilde besleyin (seed). Veya sadece uygulama çatınızdaki oturum yönetme yapılarını kullanın.
- Tercihen tokenı istemciye kalıcı olmayan cookielerde veya gizli bir alanda sayfa oluşturulurken yollayın.
- Kaba kuvvet saldırılarını anlamak için "bubi tuzağı" token değerlerini kullanın.
- Aynı IP adresinden gördüğünüz biricik oturum tokenlarının sayısını limitleyin. (Mesela son 5 dakikada 20 tane)
- Kaba kuvvet saldırı fırsat penceresini kısaltmak için periyodik olarak tokenları yeniden oluşturun.
- Eğer bir kaba kuvvet saldırısını fark edebiliyorsanız, oturumu tekrar kullanılmasını engellemek için tamamen silin.

Oturum token tekrar oynaması (replay)

Oturum tekrar oynama saldırıları saldırgan oturumları kaydeder bir pozisyonda ise basittir. Saldırgan istemci ve sunucu arasındali oturumu kaydedecek ve daha sonra sunucuya saldırmak için istemci tarafını oynayacaktır. Bu saldırı sadece kimlik doğrulama mekanizması bu saldırıyı önlemek için rasgele sayılar kullanmıyorsa çalışır.

Saldırıya açık olup olmadığınızı anlama (yolları)

- Bir oturum cookiesi alın ve başka bir tarayıcıya enjekte edin
- Simultane olarak kullanmayı deneyin – başarısız olmalı
- Zaman aşımına uğradıktan sonra kullanmayı deneyin - başarısız olmalı

Kendinizi Koruma Yolları

- Bir oturumu belli bir tarayıcıya sunucu taraflı IP adresini (REMOTE_ADDR) ve eğer başlık var ise PROXY_FORWARDED_FOR bilgilerinin kriptoğrafik özetini

(hash) alarak bağlayın. İstemci tarafında sahtesi üretilebilecek başlıkları kullanmamanız ama kriptoğrafik özetini almanız gerekir. Eğer yeni kriptoğrafik özet eskisi ile uyuşmuyorsa muhtemel ile bir oturum tekrar oynaması durumu vardır.

- Oturum token zaman aşımlarını ve tokenların tekrar oynama fırsat pencelerini kısaltmak için token yeniden oluşturulmasını kullanın
- Oturum tokenlarını oluşturmak için kriptoğrafik olarak iyi beslenmiş rasgele sayı üreteçlerini kullanın.
- Oturum tokenlarını saklamak için kalıcı olmayan cookieleri veya en kötüsü form üzerindeki gizli bir alanı kullanın.
- Uygulama için çıkış fonksiyonunu gerçekleyin. Kullanıcının çıkış veya zaman aşımı işlemini gerçekleştirirken, sadece istemci taraflı cookielerin temizlendiğine değil aynı zamanda sunucu tarafındaki oturum bilgilerinin de silindiğine emin olun. Bu zaman aşımından veya çıkış işleminden sonra oturum bilgilerinin tekrar oynama saldırılarında kullanılamamasını sağlar.

İleri Okuma

- David Endler, "*Web Uygulama Oturum ID'lerinin Kaba Kuvvet Yolu ile Kırılmaları*" http://downloads.securityfocus.com/library/SessionIDs.pdf
- Ruby CGI::Session oturum dosyalarını güvensiz olarak oluşturuyor http://www.securityfocus.com/advisories/7143

Dökümanın Aslı : OWASP GUIDE 2.0.1

Çeviri : Bedirhan Urgun - urgunb@hotmail.com