Web Açıklık Tarayıcıları
Bölüm - II

Bedirhan Urgun, Ağustos 2009, WGT E-Dergi 1. Sayı

İlk bakışta ve özellikle demolarda özellikleri ile baş döndüren bu araçlar, yakından tecrübe edildikçe ve üzerinde çalıştıkları teknolojiler/zafiyet detayları öğrenildikçe, **başarmaya çalıştıkları görevin çok karmaşık ve zor olmasından**, bazı durumlarda beklenildiği gibi çalışmadıkları görülmektedir.

Tekrarlamak gerekirse, bu yazıda otomatik tarayıcıların iddia ettikleri özelliklerin üzerine yoğunlaşılmıştır. İddia etmedikleri özellikler ise başka bir makale konusu olabilir (mesela, kullanıcı profili çapraz testleri, integer overflow, v.b.).

Tarayıcıyı Doğru Yolda Tutmak

Hassas verilerini son kullanıcılardan koruma güvenlik gereksinimi ile yazılan web uygulamaları, kimlik doğrulama ve yetkilendirme adımlarını gerçeklemektedirler. Bu uygulamaları uzaktan denetleyecek tarayıcıların, bu kontrollerden geçip uygulamanın dahili arayüzlerine ulaşması ve **uygulamadan çıkmadan** ulaşabildiği bütün arayüzleri denetlemesi gerekmektedir.

Tarayıcılar doğru yolda tutunabilmek için bir çok mantıklı seçenek getirmişlerdir; giriş makro kaydedicileri, logout link belirlenmesi, v.b. Bu seçenekler denetlenen uygulamaların çoğunluğunda (ağırlıklı olarak ASP.NET/PHP/ASP) işe yaramakla birlikte, bazı durumlarda yetersiz kalmakta ve taramaların eksik kalmasına neden olmaktadırlar.

Örnek olarak, Java RichFaces JavaServer Faces için yazılmış bir bileşen kütüphanesidir. Bu kütüphane kullanılarak, AJAX teknolojisi ile kolay ve zengin uygulamalar yazmak mümkündür. ExtJS gibi, ürettiği istemci taraflı çok karmaşık kodlarla denetleyicinin sinirini hoplatan bu teknolojinin HTML element'lere (JSF'den kaynaklanan) otomatik id değeri ataması "logout link belirlenme" tekniğini bazı durumlarda kullanılmaz hale getirmekte, bazı durumlarda ise çok zorlaştırmaktadır. Örnek bir logout POST isteği aşağıda verilmiştir;

j_id_jsp_1019999124_4=j_id_jsp_1019999124_4&javax.faces.ViewState=j_id2&j_id_jsp_1019999124_4%3Aj_id_jsp_1019999124_7=j_id_jsp_1019999124_4%3Aj_id_jsp_1019999124_7

Yukarıda küçük periyotlar ile sadece viewstate değişmektedir. Diğer parametre isimleri (ve dolayısıyla değerleri) zaman zaman sunucu yeniden başlatmalarında, uygulama redeployларında ve kritik hata durumlarında değişmektedirler. Parametre isimleri (id attribute değerleri), geliştiricinin özel olarak vermediği durumlarda otomatik olarak JSF motoru tarafından üretilmektedirler.

Yukarıda,

j_id_jsp_1019999124_4: form'un id'sini
j_id_jsp_1019999124_7: logout link'inin id'si belirtmektedir

Giriş ekranlarında CAPTCHA kullanıldığından, giriş makrosu oluşturulması ve kullanılması olanaksızdır. Logout linki belirlenmesi ise yukarıda bahsedilen durumlardan ötürü (ve karmaşık olsa dahi giriş makrosu oluşturulamadığından) mümkün değildir.

Ayrıca CAPTCHA kullanımından dolayı giriş makrosunun kullanılmadığı durumlarda açıklık tarayıcının geçerli bir session cookie'sinin verilmesi ile dahili sayfalara ulaşması sağlanabilemektedir. Eğer bu session cookie'si herhangi bir nedenden dolayı geçersiz hale gelirse (bubi tuzakları, container problemleri, v.b.) tarayıcı bunun farkına varamayacağından sonuçların eksik kalmasına ve daha kötüsü sistem yöneticilerinde ve denetçilerde yanlış bir güvenlik hissine yol açacaktır.

Açıklık Bulma Kalitesi

Tarayıcılar yıllardır bilinen ve yaygın olarak exploit edilen bazı açıklıkları bulma konusunda eksik kalabilmektedirler. Bu duruma en iyi iki örnek; Reflected XSS ve Kör SQL Enjeksiyonudur.

Bariz sentaks (syntactic) özelliği ile bulunması en kolay açıklık vektörlerinden biri olan Reflected XSS, istek ve ardından cevap ikilisinin analiz edilmesi ile ortaya çıkarılır. Ancak bu durumda bile tarayıcılarda eksiklikler görülebilmektedir. Kullanılan veya el altındaki araçlar değerlendirilmek istenirse http://www.webguvenligi.org/xsstb/reflected.php uygulaması veya çözümleri ile beraber http://www.webguvenligi.org/xsstb/reflected_hints.php uygulaması kullanılabilir.

Ek olarak, en ciddi saldırı vektörlerinden biri olan ve üzerine bir çok araştırma (hem akademik ve akademik olmayan) yapılan Kör SQL enjeksiyonunun, otomatik tarayıcılar tarafından bulunma yetenekleri hem çok kısıtlı ve hem de hataya açıktır. Bulma algoritmaları belirli ve üzerinde çok düşünülmüş olmasına rağmen her zaman bulunması mümkün olmayan bu açıklığın ortaya çıkarılmasında yine de otomatik tarayıcılara büyük iş düşmektedir. Kullanılan veya el altındaki araçlar değerlendirilmek istenirse http://www.webguvenligi.org/projeler/sqlibench projesi kullanılabilir.

Hatalı Pozitifler (False Positives)

"False positives", güvenlik taramalarının kaçınılmaz bir şekilde otomatize edilmesi ile denetçilerin hayatına giren bir terimdir. Temel olarak, tarayıcılar tarafından bulunan açıklıkların aslında var olmadığı anlamına gelir.

Tarayıcıların, bazı yaygın uygulamaları ve dosyaları tanımlaları veya bu uygulamalar/dosyalar yolu ile çıkan false positive'leri suppress edebilme seçeneğini sunmaları bilgi kirliliğinin azaltılması bakımından çok önemlidir.

Örnek olarak, açıklık tarayıcıların artık yaygın olarak kullanılan/bulunan jQuery ve Apache Manual dosyalarını otomatik olarak tanımaları ve bu dosyalar hakkında false positive (dizin gezinimi, os komut çalıştırma, v.b.) üretmemeleri gerekmektedir.

Esneklik

Tarayıcılar çalışırken, ayarlarında bazı değişiklikler yapılması gerekebilmektedir (yapılan istek sayısının azaltılması/arttırılması, kullanılan session id'nin yenilenmesi, false pozitif vermeye başlayan URL bölümlerinin atlatılması, v.b.). Tarama zamanı, bu ayar değişikliklerinin etkin olması beklenirken, etkin olmayacağı durumların kullanıcıya belirtilmesi bilgilendirici olacaktır.

Aynı doğrultuda bir açıklık tarayıcıda mutlaka Pause/Beklet özelliğinin bulunması ve bu özelliğin çalışması gerekmektedir.

Raporlama

Tarayıcıların çoğu özelleştirilebilen pdf/html raporlar üretebilmektedirler. Yine bu tarayıcılar çoğunlukla ilişkisel veritabanlarını kullandıkları için, özel araçlar tarafından kurumsal gereksinimlere göre daha granuler ve parse edilebilen XML raporları üretebilmelidir. Enterprise boyutlarda üretilen bir çok açıklık sonuç raporunun tek bir havuzda toplanabilmesi, ölçülebilirlik ve izlenebilirlik açısından son derece önemlidir. Bir çok tarayıcının (network, audit, web) raporlarının konsolide edilebilmesi bu açıdan hayati önem taşımaktadır.

Web Servisler ve İstemci Taraflı Teknolojilerin Desteği

Farklı bir çok web servis teknolojisi ile üretilen uygulamaların otomatik tarayıcı tarafından algılanabilir ve denetlenebilir olması gereklidir. Bu teknolojilerden bazıları aşağıda listelenmiştir;

JBossWS
JBossWS JAX-WS
JAX-RPC
JAX-WS
Axis 1.x
Axis 2
Apache CFX
XFire 1.x
Oracle Proxy
XmlBeans
JAXB 2.0
.NET 2.0
Gsoap

Web üzerinden servis edilen istemci taraflı teknolojilerdeki açıklık vektörlerinin de denetlenebilmesi otomatik tarayıcıların özelliklerinden biri olmalıdır. Bu teknolojilerden en önemlisi Adobe Flash'tır.

Ürün Spesifik Uygulama Açıklıkları

Bu özellik web uygulama tarayıcılarında olması gereken en önemli özelliklerden biridir. Otomatik veya manuel olarak üçüncü parti uygulamalardaki bilinen ve yayınlanmış web açıklıklarının kontrol edilmesi ve bulunması %100 mümkün değildir. Yeterli kaynak bulunmaması bu durumdaki en büyük nedendir.

Tarayıcı bir otomatik update mekanizması ile neredeyse her gün yenisi çıkan bu tür zafiyetlerin kurallarını öğrenmeli ve uygulamalıdır.

İstek Sayısı İnce Ayarı ve İstek İzleme

Tarayıcı tarafından üretilen istek sayısının ince ayarının yapılabilmesi (network tarayıcılarındaki dakikadaki paket sayısı gibi web tarayıcılarındaki dakikadaki istek sayısı) bazı durumlarda çok önemli hale gelmektedir; IPS, uygulama bubi tuzakları, uygulamanın saturasyon eşik değerleri, v.b.

Bu özelliğin yanında tarayıcı tarafından üretilen isteklerin mutlaka kaydedilme ve tarama anında veya daha sonra detaylı olarak izlenme özelliğinin tarayıcılar tarafından gerçeklenmiş olması gerekmektedir. Oturumun geçersiz olmaya başladığı, hatalı negatiflerin nedeninin bulunması gibi kritik durumlarda bu izleme yetisi denetleyiciye önemli bilgiler sunacaktır. Bu özellik çok yer alabileceğinden opsiyonel hale getirilebilir.

Sonuç

Yazıda web açıklık tarayıcılarının kısa geldiği bazı durumlardan ve özelliklerinden bahsedilmektedir. Bu özelliklerin bazılarının gerçeklenebilmesi karşılık geldikleri problemlerin doğaları gereği zordur. Bu araçları network tarayıcıları ile kıyaslamak bu açıdan doğru olmayacaktır ama çok değişkenlik gösteren web ormanında çok kaliteli bir tarayıcının çıkması biraz daha zaman alacak gibi gözükmektedir.