Map Execution Paths Through Application (Uygulama Üzerinden Yürütme Yollarını Eşleme)

Summary (Özet)

Güvenlik testine başlamadan önce, uygulamanın yapısını anlamak çok önemlidir. Uygulamanın düzeni hakkında kapsamlı bir anlayış olmadan, iyice test edilmesi pek olası değildir.

Test Objectives (Test Hedefleri)

- Hedef uygulamayı haritalayın ve temel iş akışlarını anlayın.

How to Test (Nasıl Test Edilir)

Kara kutu testinde tüm kod tabanını test etmek son derece zordur. Sadece testçinin uygulamadaki kod yolları hakkında hiçbir fikri olmadığı için değil, aynı zamanda yapsalar bile, tüm kod yollarını test etmek çok zaman alıcı olacaktır. Bunu uzlaştırmanın bir yolu, hangi kod yollarının keşfedildiğini ve test edildiğini belgelemektir.

Kod kapsamının testine ve ölçümüne yaklaşmanın birkaç yolu vardır:

- **Yol** - her bir yol, her bir karar yolu için kombinatörel ve sınır değeri analizi testi içeren bir uygulama aracılığıyla test edin. Bu yaklaşım titizlik sunarken, test edilebilir yolların sayısı her karar şubesinde katlanarak artar.
- **Veri Akışı (veya Taint Analizi)** - değişkenlerin atamasını harici etkileşim (normalde kullanıcılar) yoluyla test eder. Bir uygulama boyunca verilerin akışının, dönüşümünü ve kullanımını haritalamaya odaklanır.

- **Irk** - aynı verileri manipüle eden uygulamanın birden fazla eşzamanlı örneğini test eder.

Hangi yöntemin kullanıldığına ilişkin takas ve her yöntemin hangi derece kullanıldığından başvuru sahibi ile müzakere edilmelidir. Uygulama sahibine özellikle hangi işlevlerden veya kod bölümleri hakkında hangi işlevlere veya kod bölümlerine ulaşılabileceğini ve bu kod segmentlerine nasıl ulaşılabileceğini sormak da dahil olmak üzere daha basit yaklaşımlar da benimsenebilir.

Uygulama sahibine kod kapsamını göstermek için, test cihazı bir elektronik tablo ile başlayabilir ve uygulamayı örtbas ederek keşfedilen tüm bağlantıları belgeleyebilir (ya manuel veya otomatik olarak). Daha sonra test cihazı uygulamadaki karar noktalarına daha yakından bakabilir ve kaç önemli kod yolunun keşfedildiğini araştırabilir. Bunlar daha sonra elektronik tabloda, keşfedilen yolların URL'leri, düzyazı ve ekran görüntüsü açıklamalarıyla belgelenmelidir.

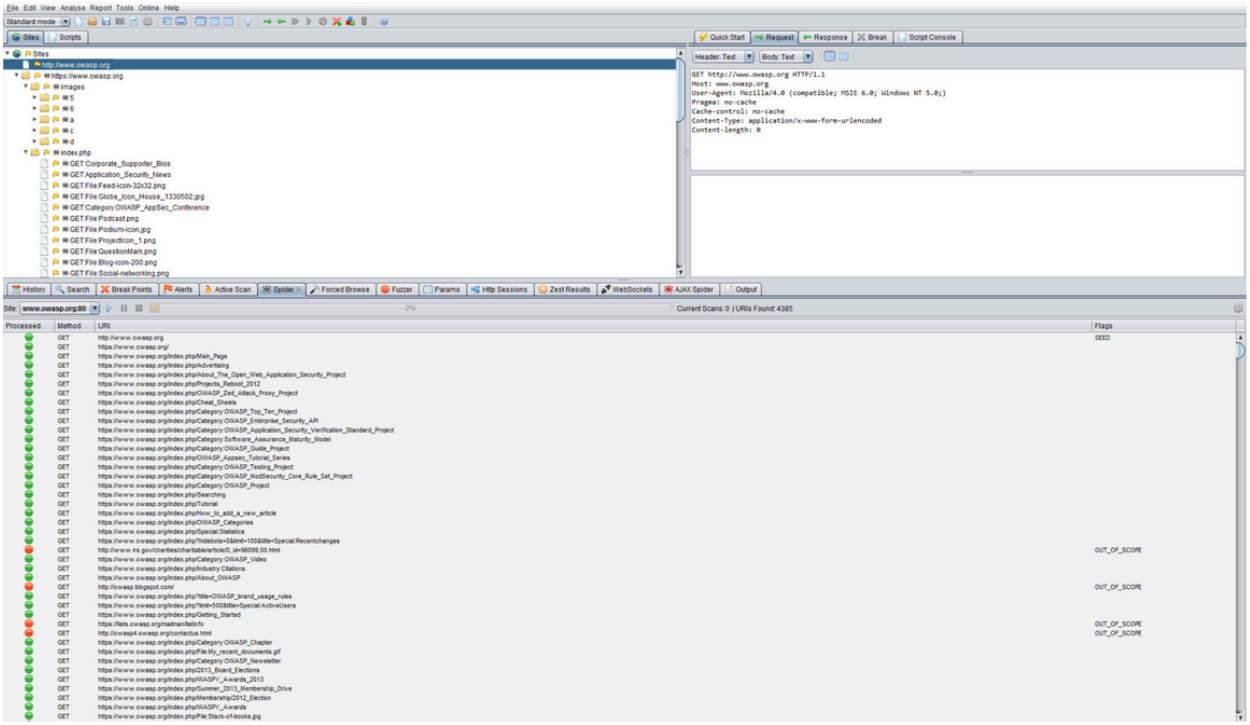
Code Review (Kod İncelemesi)

Uygulama sahibi için yeterli kod kapsamının sağlanması, test için gri kutu ve beyaz kutu yaklaşımı ile çok daha kolaydır. Test cihazı tarafından talep edilen ve onaylanan bilgiler, kod kapsamı için minimum gerekliliklerin karşılanmasını sağlayacaktır.

Birçok modern Dinamik Uygulama Güvenliği Testi (DAS) aracı, bir web sunucusu temsilcisinin kullanımını kolaylaştırır veya web uygulama kapsamı özelliklerini izlemek için üçüncü taraf bir temsilci ile eşleştirilebilir.

Automatic Spidering (Otomatik Örümcekleme)

Otomatik örümcek, belirli bir web sitesinde yeni kaynakları (URL'ler) otomatik olarak keşfetmek için kullanılan bir araçtır. Örümcek'in nasıl başlatıldığına bağlı olan tohumlar olarak adlandırılan ziyaret edilecek URL'lerin bir listesi ile başlar. Çok sayıda Örümcekleme aracı olsa da, aşağıdaki örnek Zed Attack Proxy'yi (ZAP) kullanır:



Şekil 4.1.7-1: Zed Saldırısı Vekale Ekranı

ZAP, testçinin ihtiyaçlarına göre kaldıraçlı olabilen çeşitli otomatik örümcekleme seçenekleri sunar:

- Örümcek
- AJAX Örümceği
- OpenAPI Desteği

Tools (Araçlar)

- Zed Saldırı Proxy (ZAP)
- Elektronik tablo yazılımı listesi
- Diyagramlama yazılımı

References (Referanslar)

- Kod Kapsamı