Review Old Backup and Unreferenced Files for Sensitive Information (Hassas Bilgiler için Eski Yedekleri ve Referanslanmamış Dosyaları İnceleyin)

Summary (Özet)

Bir web sunucusundaki dosyaların çoğu doğrudan sunucunun kendisi tarafından ele alınırken, altyapı veya kimlik bilgileri hakkında önemli bilgiler edinmek için kullanılabilecek referanssız veya unutulmuş dosyaları bulmak nadir değildir.

Çoğu yaygın senaryo, değiştirilmiş dosyaların yeniden adlandırılmış eski sürümlerinin varlığını, tercih edilen dile yüklenen ve kaynak olarak indirilebilen ve basınçlı arşivler şeklinde otomatik veya manuel yedeklemeler olarak indirilebilen dahil dosyaları içerir. Yedekleme dosyaları, uygulamanın üzerinde barındırılan temel dosya sistemi tarafından otomatik olarak da oluşturulabilir, genellikle “snapshots” olarak adlandırılan bir özelliktir.

Tüm bu dosyalar, test cihazına iç işlerine, arka kapılara, idari arayüzlere ve hatta kimlik bilgilerine, idari arayüze veya veri tabanı sunucusuna bağlanmak için erişim sağlayabilir.

Savunmasızlığın önemli bir kaynağı, uygulamayla ilgisi olmayan, ancak uygulama dosyalarını düzenlemenin bir sonucu olarak veya uçuşta yedekleme kopyaları oluşturduktan sonra veya web ağacı eski dosyaları veya referanssız dosyalarda bırakarak oluşturulan dosyalarda bulunur. Yerinde düzenleme veya diğer idari işlemleri gerçekleştirmek, dosyaları düzenlerken editör tarafından otomatik olarak oluşturulan veya bir dizi dosya oluşturmak veya yönetici tarafından bir dizi dosya oluşturabilir.

Bu tür dosyaları unutmak kolaydır ve bu da başvuru için ciddi bir güvenlik tehdidi oluşturabilir. Bu, yedekleme kopyalarının orijinal dosyalardan farklı dosya uzantıları ile oluşturulabileceği için olur. A.

`.tar`, `.zip` ya da `.gz` Ürettiğimiz (ve unuttuğumuz) arşiv açıkça farklı bir uzantıya sahiptir ve aynı şey birçok editör tarafından oluşturulan otomatik kopyalarla da olur (örneğin, emacs, adı verilen bir yedek kopya oluşturur. `file~` Düzenlenirken `file`). Bir kopyasını elle yapmak aynı etkiyi yaratabilir (koplama yapmayı düşünün `file` tot için `file.old`). Uygulamanın üzerinde olduğu altta yatan dosya sistemi yapılabilir `snapshots` Bilginiz olmadan zaman içinde farklı noktalarda uygulamanızın, web üzerinden de erişilebilir, benzer ama farklı bir poz veren `backup file` Uygulamanız için stil tehdidi.

Sonuç olarak, bu faaliyetler uygulama tarafından ihtiyaç duyulmayan dosyaları oluşturur ve web sunucusu tarafından orijinal dosyadan farklı olarak ele alınabilir. Örneğin, bir kopyasını yaparsak `login.asp` adı verildi `login.asp.old` Kullanıcıların kaynak kodunu indirmelerine izin veriyoruz `login.asp`. . Bunun nedeni budur. `login.asp.old` Genellikle uzatması nedeniyle idam edilmek yerine metin veya sade olarak hizmet edilecektir. Başka bir deyişle, erişim `login.asp` Sunucu tarafı kodunun yürütülmesine neden olur `login.asp`, erişim sırasında `login.asp.old` İçeriğine neden olur `login.asp.old` (yine, yine sunucu tarafı kodu) kullanıcıya açıkça iade edilmek ve tarayıcıda görüntülenmek için. Bu, hassas bilgiler ortaya çıkabileceğinden, güvenlik riskleri oluşturabilir.

Genel olarak, sunucu tarafı kodunu ifşa etmek kötü bir fikirdir. Sadece gereksiz yere iş mantığını ifşa etmekle kalmıyorsunuz, aynı zamanda bir saldırgana (yol adları, veri yapıları vb.) yardımcı olabilecek uygulama ile ilgili bilgileri bilmeden ortaya koyuyor olabilirsiniz. Net metinde gömülü kullanıcı adı ve şifreye sahip çok fazla komut dosyası olduğu gerçeğinden bahsetmiyorum bile (düşüncesiz ve çok tehlikeli bir uygulamadır).

Referanssız dosyaların diğer nedenleri, veri dosyaları, yapılandırma dosyaları, kayıt dosyaları, kayıt dosyaları, web sunucusu tarafından erişilebilen dosya sistemi dizinlerinde saklanmasına izin verdiklerinde tasarım veya yapılandırma seçeneklerinden kaynaklanmaktadır. Bu dosyaların normalde web üzerinden erişilebilen bir dosya sistemi alanında olması için hiçbir nedeni yoktur, çünkü yalnızca uygulama düzeyinde, uygulamanın kendisi tarafından (ve etrafta dolaşan gündelik kullanıcı tarafından değil) erişilmelidir.

Threats (Tehditler)

Eski, yedekleme ve referanssız dosyalar, bir web uygulamasının güvenliğine yönelik çeşitli tehditler sunar:

- Referanssız dosyalar, uygulamaya yönelik odaklanmış bir saldırıyı kolaylaştırabilecek hassas bilgileri açığa çıkarabilir; örneğin veritabanı kimlik bilgilerini içeren dosyaları, diğer gizli içeriğe referanslar içeren yapılandırma dosyalarını, mutlak dosya yollarını vb.içerir.
- Referanssız sayfalar, uygulamaya saldırmak için kullanılabilecek güçlü işlevsellik içerebilir; örneğin, yayınlanmış içerikten bağlantısı olmayan ancak nerede bulacağını bilen herhangi bir kullanıcı tarafından erişilebilen bir yönetim sayfası.
- Eski ve yedekleme dosyaları, daha yeni sürümlerde düzeltilmiş güvenlik açıkları içerebilir; örneğin `viewdoc.old.jsp` İçinde sabitlenmiş bir dizin geçişi güvenlik açığı içerebilir `viewdoc.jsp` Ama yine de eski versiyonu bulan herkes tarafından sömürülebilir.
- Yedekleme dosyaları sunucuda yürütmek için tasarlanmış sayfaların kaynak kodunu açıklayabilir; örneğin talep `viewdoc.bak` Kaynak kodunu iade edebilir `viewdoc.jsp` , yürütülebilir sayfaya kör taleplerde bulunarak bulunması zor olabilecek güvenlik açıkları için incelenebilir. Bu tehdit açıkça Perl, PHP, ASP, kabuk komut dosyaları, JSP vb. gibi senaryolu diller için geçerli olsa da, bir sonraki mermide sağlanan örnekte gösterildiği gibi bunlarla sınırlı değildir.
- Yedekleme arşivleri, web tabanı içindeki (hatta dışarıda) tüm dosyaların kopyalarını içerebilir. Bu, bir saldırganın, karşılıksız sayfalar, kaynak kodu, içeren dosyaları vb. dahil olmak üzere tüm uygulamayı hızlı bir şekilde numaralandırmasına izin verir. Örneğin, adı geçen bir dosyayı unutursanız `myservlets.jar.old` Servis uygulama sınıflarınızı içeren (kaldırma kopyası) içeren dosya, ayrışma ve tersine mühendislike duyarlı birçok hassas bilgiyi açığa çıkarıyorsunuz.
- Bazı durumlarda bir dosyayı kopyalamak veya düzenlemek dosya uzantısını değiştirmez, ancak dosya adını değiştirir. Bu, örneğin, dosya kopyalama işlemlerinin “sevkiyat” veya bu dizenin yerelleştirilmiş sürümleri ile önceden

yüklenmiş dosya adları oluşturduğu Windows ortamlarında olur. Dosya uzantısı değişmeden bırakıldığından, bu, yürütülebilir bir dosyanın web sunucusu tarafından düz metin olarak iade edildiği ve bu nedenle kaynak kodu açıklaması yapılmadığı bir durum değildir. Bununla birlikte, bu dosyalar da tehlikelidir, çünkü çağrıldığında, teşhis mesajının etkinleştirilmesi durumunda, bir saldırgana değerli bilgiler verebilecek uygulama hatalarını tetikleyebilecek eskimiş ve yanlış mantık içerme şansı vardır.

- Günlük dosyaları, uygulama kullanıcılarının faaliyetleri hakkında hassas bilgiler içerebilir, örneğin URL parametrelerinde, oturum kimliklerinde, ziyaret edilen URL'lerde (ek referanssız içeriği ifşa edebilecek) vb. Diğer günlük dosyaları (örneğin ftp günlükler) sistem yöneticileri tarafından uygulamanın bakımı hakkında hassas bilgiler içerebilir.
- Dosya sistemi anlık görüntüleri, daha yeni sürümlerde düzeltilmiş güvenlik açıkları içeren kodun kopyalarını içerebilir. Örneğin `/.snapshot/monthly.1/view.php` İçinde sabitlenmiş bir dizin geçişi güvenlik açığı içerebilir `/view.php` Ama yine de eski versiyonu bulan herkes tarafından sömürülebilir.

Test Objectives (Test Hedefleri)

- Hassas bilgiler içerebilecek referanssız dosyaları bulun ve analiz edin.

How to Test (Nasıl Test Edilir)

Black-Box Testing (Siyah-Kutu Testi)

Referanssız dosyalar için test hem otomatik hem de manuel teknikler kullanır ve genellikle aşağıdakilerin bir kombinasyonunu içerir:

Inference from the Naming Scheme Used for Published Content (Yayınlanan İçerik için Kullanılan İsimleme Şemasından Çıkarım)

Uygulamanın tüm sayfalarını ve işlevselliğini numaralandırın. Bu, bir tarayıcı kullanılarak veya bir uygulama örümcek aracı kullanılarak manuel olarak yapılabilir. Çoğu uygulama tanınabilir bir adlandırma şeması kullanır ve özgeçmişlerini tanımlayan kelimeleri kullanarak sayfalara ve dizinlere kaynakları düzenler. Yayınlanan içerik için kullanılan adlandırma şemasından, referanssız sayfaların adını ve yerini çıkarmak genellikle mümkündür. Örneğin, bir sayfa varsa

`viewuser.asp` bulunur, sonra da arayın `edituser.asp` , `adduser.asp` ve `deleteuser.asp` . . Bir dizin varsa `/app/user` bulunur, sonra da arayın `/app/admin` ve `/app/manager` . .

Other Clues in Published Content (Yayınlanan İçerikteki Diğer İpuçları)

Birçok web uygulaması, gizli sayfaların ve işlevselliğin keşfine yol açabilecek yayınlanmış içerikte ipuçları bırakır. Bu ipuçları genellikle HTML ve JavaScript dosyalarının kaynak kodunda görünür. Yayınlanan tüm içerik için kaynak kodu, diğer sayfalar ve işlevsellik hakkındaki ipuçlarını tanımlamak için manuel olarak gözden geçirilmelidir. Örneğin:

Programcıların yorumları ve kaynak kodunun yorumlanan bölümleri gizli içeriğe atıfta bulunabilir:

```
<!-- <A HREF="uploadfile.jsp">Upload a document to the server</A> -->
<!-- Link removed while bugs in uploadfile.jsp are fixed          -->
```

JavaScript, belirli koşullar altında yalnızca kullanıcının GUI'sinde yapılan sayfa bağlantıları içerebilir:

```
var adminUser=false;
if (adminUser) menu.add (new menuItem ("Maintain users", "/admin/useradmin.j
```

HTML sayfaları, SUMBIT öğesini devre dışı bırakarak gizlenmiş FORM içerebilir:

```
<form action="forgotPassword.jsp" method="post">
   <input type="hidden" name="userID" value="123">
   <!-- <input type="submit" value="Forgot Password"> -->
</form>
```

Referanssız dizinlerle ilgili bir diğer ipucu kaynağı da `/robots.txt` Web robotlarına talimatlar vermek için kullanılan dosya:

```
User-agent: *
Disallow: /Admin
Disallow: /uploads
```

```
Disallow: /backup
Disallow: /~jbloggs
Disallow: /include
```

Blind Guessing (Kör Tahmin)

En basit haliyle, bu, sunucuda var olan dosyaları ve dizinleri tahmin etmek için bir istek motoru aracılığıyla ortak dosya adlarının bir listesini çalıştırmayı içerir. Aşağıdaki netcat sarma senaryosu standinden bir kelime listesi okuyacak ve temel bir tahmin saldırısı gerçekleştirecektir:

```
#!/bin/bash

server=example.org
port=80

while read url
do
echo -ne "$url\t"
echo -e "GET /$url HTTP/1.0\nHost: $server\n" | netcat $server $port | head -1
done | tee outputfile
```

Sunucuya bağlı olarak, daha hızlı sonuçlar için GET BAŞLIK ile değiştirilebilir. Belirtilen çıkış dosyası “ilginç” yanıt kodları için tutulabilir. 200 (KİM) yanıt kodu genellikle geçerli bir kaynağın bulunduğunu belirtir (sözcün 200 kodu kullanarak özel bir “bulut bulunmama” sayfası sunmaması şartıyla). Ancak aynı zamanda 301 (Moved), 302 (Bitiş), 401 (Yetkisiz), 403 (Yasaksız) ve 500 (İç Hata) için de dikkat edin, bu da daha fazla araştırmaya değer kaynakları veya dizinleri gösterebilir.

Temel tahmin saldırısı, web tabanına ve ayrıca diğer numaralandırma teknikleri ile tanımlanan tüm dizinlere karşı yürütülmelidir. Daha ileri / etkili tahmin saldırıları aşağıdaki gibi yapılabilir:

- Uygulamanın bilinen alanlarında kullanımdaki dosya uzantılarını tanımlayın (örneğin jsp, aspx, html) ve bu uzantıların her birine eklenen temel bir kelime

listesi kullanın (veya kaynak izin verirse daha uzun bir ortak uzantı listesi kullanın).

- Diğer numaralandırma teknikleri ile tanımlanan her dosya için, bu dosyadan türetilen özel bir kelime listesi oluşturun. Ortak dosya uzantılarının bir listesini alın (@, bak, txt, src, dev, eski, orki, kopya, tmp, swp, vb.) dahil olmak üzere alın ve her uzantıyı gerçek dosya adının uzatılmasından önce, sonra ve yerine kullanın.

Not: Windows dosya kopyalama işlemleri, bu dizenin “veya yerelleştirilmiş sürümlerinin kopyalanması” ile önceden yüklenmiş dosya adları oluşturur, bu nedenle dosya uzantılarını değiştirmezler. “Dosyaların kopyası genellikle erişildiğinde kaynak kodunu açıklamazken, çağrıldığında hataya neden olmaları durumunda değerli bilgiler verebilirler.

Information Obtained Through Server Vulnerabilities and Misconfiguration (Sunucu Güvenlik Açıkları ve Yanlış Yapılandırma Yoluyla Elde Edilen Bilgiler)

Yanlış yapılandırılmış bir sunucunun referanssız sayfaları açıklayabileceği en belirgin yol, dizin listelemeden geçer. Tüm numaralandırılmış dizinlerden bir dizin listesi sağlayan herhangi bir şeyi tanımlamasını isteyin.

Bireysel web sunucularında, bir saldırganın referanssız içerik numaralandırmasına izin veren çok sayıda güvenlik açığı bulunmuştur, örneğin:

- Apache ? M = D dizin listeleme güvenlik açığı.
- Çeşitli IIS komut dosyası kaynak açıklama güvenlik açıkları.
- IIS WebDAV dizinin güvenlik açıklarını listeler.

Use of Publicly Available Information (Kamuya Açık Bilgi Kullanımı)

Uygulamanın içinden referans alınmayan İnternet'e bakan web uygulamalarında sayfalar ve işlevsellik diğer kamu alanlı kaynaklarından referans alınabilir. Bu referansların çeşitli kaynakları vardır:

- Eskiden referans alınan sayfalar hala İnternet arama motorlarının arşivlerinde görünebilir. Örneğin, `1998results.asp` Artık bir şirketin web sitesinden

bağlanamayabilir, ancak sunucuda ve arama motoru veritabanlarında kalabilir. Bu eski komut dosyası, tüm siteyi tehlikeye atmak için kullanılabilecek güvenlik açıkları içerebilir. The (İngilizce) `site:` Google arama operatörü, örneğin aşağıdakiler gibi yalnızca seçim alanına karşı bir sorgu yürütmek için kullanılabilir: `site:www.example.com` . . Arama motorlarını bu şekilde kullanmak, yararlı bulabileceğiniz ve
içinde açıklanan geniş bir dizi teknikliğe yol açmıştır.
`Google Hacking` Bu kılavuzun bölümü. Test becerilerinizi Google üzerinden geliştirmek için kontrol edin. Yedekleme dosyalarının diğer dosyalar tarafından referans alınması muhtemel değildir ve bu nedenle Google tarafından dizine eklenmemiş olabilir, ancak kaşılabilir dizinlerde bulunurlarsa, arama motoru bunları biliyor olabilir.

- Buna ek olarak, Google ve Yahoo, robotları tarafından bulunan sayfaların önbelleğe alınmış sürümlerini saklar. Olsa bile `1998results.asp` Hedef sunucudan kaldırıldı, çıktısının bir sürümü hala bu arama motorları tarafından saklanabilir. Önbelleğe alınmış sürüm, sunucuda hala kalan ek gizli içeriğe referanslar veya ipuçları içerebilir.
- Bir hedef uygulama içinden referans alınmayan içerik, üçüncü taraf web siteleri tarafından bağlanabilir. Örneğin, üçüncü taraf tüccarlar adına çevrimiçi ödemeleri işleyen bir uygulama, (normalde) yalnızca müşterilerinin web sitelerindeki bağlantıları takip ederek bulunabilecek çeşitli ısmarlama işlevsellik içerebilir.

Filename Filter Bypass (Dosya adı Filter Bypass)

İnkar listesi filtreleri düzenli ifadelere dayandığından, bazen geliştiricinin beklemediği şekillerde çalıştığı belirsiz OS dosya adı genişletme özelliklerinden yararlanabilirsiniz. Test cihazı bazen dosya adlarının uygulama, web sunucusu ve altta yatan işletim sistemi tarafından ayrıştırıldığı ve dosya adı sözleşmeleri ile farklılıkları kullanabilir.

Örnek: Windows 8.3 dosya adı genişletme `c:\\program files` Olur `C:\\PROGRA\~1`

- Uyumsuz karakterleri kaldırın
- Alanları alt göze çarpacak şekilde dönüştürün
- Temel isminin ilk altı karakterini alın

- Ekleyin `~<digit>` Aynı altı ilk karakteri kullanarak isimlerle dosyaları ayırt etmek için kullanılan
- Bu sözleşme ilk 3 dosyadan sonra değişir
- Üç karaktere açılan dosya uzantısı
- Tüm karakterleri üst sıralarda yapın

Gray-Box Testing (Gri-Kutu Testi)

Eski ve yedekleme dosyalarına karşı gri kutu testi yapmak, web uygulama altyapısının web sunucusu(lar) tarafından sunulan web dizinleri setine ait dizinlerde bulunan dosyaların incelenmesini gerektirir. Teorik olarak muayene iyice elle yapılmalıdır. Bununla birlikte, çoğu
durumda dosyaların veya yedekleme dosyalarının kopyaları aynı adlandırma sözleşmelerini kullanarak oluşturulma eğiliminde olduğundan, arama kolayca yazılabilir. Örneğin, editörler yedek kopyaları tanınabilir bir uzantı veya sonla adlandırarak geride bırakırlar ve insanlar dosyaları
bir ile geride bırakma eğilimindedir.
`.old` veya benzer öngörülebilir uzantılar. İyi bir strateji, periyodik olarak, bunları kopya veya yedekleme dosyaları olarak tanımlaması muhtemel
uzantıları olan dosyaları kontrol eden bir arka plan işi planlamak ve manuel kontrollerin yanı sıra daha uzun bir süre içinde yapılmasıdır.

Remediation (Düzeltme)

Etkili bir koruma stratejisini garanti etmek için, test, aşağıdakiler gibi tehlikeli uygulamaları açıkça yasaklayan bir güvenlik politikası ile birleştirilmelidir:

- Web sunucusunda veya uygulama sunucusu dosya sistemlerinde dosyaların düzenlenmesi. Bu özellikle kötü bir alışkanlıktır, çünkü editörler tarafından yedekleme veya geçici dosyalar oluşturması muhtemeldir. Bunun büyük organizasyonlarda bile ne sıklıkta yapıldığını görmek şaşırtıcıdır. Bir üretim sistemindeki dosyaları kesinlikle düzenlemeniz gerekiyorsa, açıkça amaçlanmayan hiçbir şeyi geride bırakmadığınızdan emin olun ve bunu kendi riskinizle yaptığınızı düşünün.
- Web sunucusu tarafından açığa çıkan dosya sistemlerinde gerçekleştirilen diğer etkinlikleri, spot yönetim faaliyetleri gibi dikkatlice kontrol edin. Örneğin,

ara sıra birkaç dizinin anlık görüntüsünü almanız gerekiyorsa (bir üretim sisteminde yapmamalısınız), önce onları fermuarlamak için cazip olabilirsiniz. O arşiv dosyalarını geride bırakmamaya dikkat edin.

- Uygun yapılandırma yönetimi politikaları, eskimiş ve referanssız dosyaları önlemeye yardımcı olmalıdır.
- Uygulamalar, web sunucusu tarafından sunulan web dizin ağaçlarının altında saklanan dosyaları oluşturmamak (veya güvenmemek) şekilde tasarlanmalıdır. Veri dosyaları, kayıt dosyaları, yapılandırma dosyaları vb. bilgi açıklaması olasılığına karşı koymak için web sunucusu tarafından erişilemeyen dizinlerde saklanmalıdır (web dizini izinleri yazılmasına izin verirse veri değişikliğinden bahsetmeyin).
- Dosya sistemi anlık görüntülerine, belge kökü bu teknolojiyi kullanan bir dosya sistemindeyse web üzerinden erişilebilir olmamalıdır. Bu tür dizinlere erişimi reddetmek için web sunucunuzu yapılandır, örneğin Apache bir konum yönergesi altında, bu şekilde kullanılmalıdır:

```
<Location ~ ".snapshot">
   Order deny,allow
   Deny from all
</Location>
```

Tools (Araçlar)

Güvenlik açığı değerlendirme araçları, standart isimlere ([admin", “test”, “backup” vb.) sahip web dizinlerini tespit etmek için kontrolleri içerme ve indekslemeye izin veren herhangi bir web dizinini bildirmeyi içerir. Herhangi bir dizin listesi alamazsanız, muhtemel yedekleme uzantılarını kontrol etmeyi denemelisiniz. Örneğini kontrol edin

- Nessus
- Nikto2

Web örümcek aletleri

- wget

- Windows için web
- Sam Spade'in
- Spike proxy bir web sitesi tarayıcı işlevi içerir
- Xenu
- Curl

Bunlardan bazıları standart Linux dağıtımlarına da dahildir. Web geliştirme araçları genellikle kırık bağlantıları ve referanssız dosyaları tanımlamak için tesisleri içerir.