Enumerate Infrastructure and Application Admin Interfaces (Altyapı ve Uygulama Yönetici Arayüzlerini Listeleme)

Summary (Özet)

Yönetici arayüzleri, belirli kullanıcıların sitede ayrıcalıklı faaliyetler üstlenmesine izin vermek için uygulamada veya uygulama sunucusunda mevcut olabilir. Bu ayrıcalıklı işlevselliğe yetkisiz veya standart bir kullanıcı tarafından nasıl erişilebileceğini ve nasıl erişilebileceğini ortaya çıkarmak için testler yapılmalıdır.

Bir uygulama, ayrıcalıklı bir kullanıcının sitenin nasıl çalıştığı konusunda değişiklik yapabilecek işlevselliğe erişmesini sağlamak için bir yönetici arayüzü gerektirebilir. Bu tür değişiklikler şunları içerebilir:

- kullanıcı hesabı temini
- Site tasarımı ve düzeni
- Veri manipülasyonu
- yapılandırma değişiklikleri

Birçok durumda, bu tür arayüzler onları yetkisiz erişimden korumak için yeterli kontrole sahip değildir. Test, bu yönetici arayüzlerini keşfetmeyi ve ayrıcalıklı kullanıcılar için tasarlanan işlevselliğe erişmeyi amaçlamaktadır.

Test Objectives (Test Hedefleri)

- Gizli yönetici arayüzlerini ve işlevselliğini belirleyin.

How to Test (Nasıl Test Edilir)

Black-Box Testing (Siyah-Kutu Testi)

Aşağıdaki bölüm, idari arayüzlerin varlığını test etmek için kullanılabilecek vektörleri açıklar. Bu teknikler ayrıca ayrıcalıklılığın tırmanması da dahil olmak

üzere ilgili konuları test etmek için kullanılabilir ve bu kılavuzda başka bir yerde tanımlanmıştır (örneğin, yetkisizlik şeması ve Güvensiz Doğrudan Nesne Referansları Testi için Testler daha ayrıntılı olarak.

- Yönetmenlik ve dosya numaralandırması. Bir idari arayüz mevcut olabilir, ancak test cihazına görünür bir şekilde mevcut değildir. İdari arayüzün yolunu tahmin etmeye çalışmak, talep etmek kadar basit olabilir: */admin veya /admintrator vb. veya* bazı senaryolarda Google çukurları kullanılarak saniyeler içinde ortaya çıkabilir.
- Sunucu içeriğinin kaba kuvvetlendirilmesini gerçekleştirmek için birçok araç vardır, daha fazla bilgi için aşağıdaki araçlar bölümüne bakın. Bir testçinin de yönetim sayfasının dosya adını tespit etmesi gerekebilir. Belirlenen sayfaya zorla taramak arayüze erişim sağlayabilir.
- Kaynak kodunda yorumlar ve bağlantılar. Birçok site, tüm site kullanıcıları için yüklenen ortak kod kullanır. Müşteriye gönderilen tüm kaynakları inceleyerek, yönetici işlevselliğine bağlantılar
  keşfedilebilir ve araştırılmalıdır.
- Sunucu ve uygulama dokümantasyonu inceleme. Uygulama sunucusu veya uygulaması varsayılan yapılandırmasında dağıtılırsa, yapılandırmada veya yardımcı dokümantasyonda açıklanan bilgileri kullanarak yönetim arayüzüne erişmek mümkün olabilir. Bir idari arayüz bulunursa ve kimlik bilgilerine ihtiyaç duyulursa varsayılan şifre listelerine danışılmalıdır.
- Kamuya açık bilgiler. WordPress gibi birçok uygulama varsayılan idari arayüzlere sahiptir.
- Alternatif sunucu bağlantı noktası. Yönetim arayüzleri ana uygulamadan farklı bir bağlantı noktasında görülebilir. Örneğin, Apache Tomcat'ın Yönetim arayüzü genellikle 8080 numaralı portda görülebilir.
- Parametre kurcalama. Yönetici işlevselliğini etkinleştirmek için bir GET veya POST parametresi veya bir çerez değişkeni gerekebilir. Bunun ipuçları, aşağıdakiler gibi gizli alanların varlığını içerir:

  `<input type="hidden" name="admin" value="no">`

Ya da bir kurabiyede:

`Cookie: session_cookie; useradmin=0`

Bir idari arayüz keşfedildikten sonra, yukarıdaki tekniklerin bir kombinasyonu kimlik doğrulamayı atlamaya çalışmak için kullanılabilir. Bu başarısız olursa, testçi kaba kuvvet saldırısı girişiminde bulunmak isteyebilir. Böyle bir durumda, test cihazı, bu tür bir işlevsellik mevcutsa, idari hesap kilitleme potansiyelinin farkında olmalıdır.

Gray-Box Testing (Gri Kutu Testi)

Sertleştirmeyi sağlamak için sunucu ve uygulama bileşenlerinin daha ayrıntılı bir incelemesi yapılmalıdır (yani yönetici sayfaları IP filtreleme veya diğer kontroller yoluyla herkes tarafından erişilebilir değildir) ve uygulanabilir olduğunda, tüm bileşenlerin varsayılan kimlik bilgilerini veya yapılandırmaları kullanmadığını doğrulamak. Yetkilendirme ve kimlik doğrulama modelinin normal kullanıcılar ve site yöneticileri arasındaki görevlerin net bir şekilde çözülmesini sağlamak için kaynak kodu gözden geçirilmelidir. Bu tür bileşenlerin çizimi ile bu tür paylaşılan işlevlerin bu tür paylaşılan işlevsellikten sızması ile bilgi sızıntısı arasında net bir ayrım yapılmasını sağlamak için normal ve yönetici kullanıcıları arasında paylaşılan kullanıcı arayüzü işlevleri gözden geçirilmelidir.

Her web çerçevesinin kendi yönetici varsayılan sayfaları veya yolu olabilir. Örneğin

WebSphere:

```
/admin
/admin-authz.xml
/admin.conf
/admin.passwd
/admin/*
/admin/logon.jsp
/admin/secure/logon.jsp
```

PHP:

```
/phpinfo
/phpmyadmin/
/phpMyAdmin/
/mysqladmin/
```

```
/MySQLadmin
/MySQLAdmin
/login.php
/logon.php
/xmlrpc.php
/dbadmin
```

FrontPage:

```
/admin.dll
/admin.exe
/administrators.pwd
/author.dll
/author.exe
/author.log
/authors.pwd
/cgi-bin
```

WebLogic:

```
/AdminCaptureRootCA
/AdminClients
/AdminConnections
/AdminEvents
/AdminJDBC
/AdminLicense
/AdminMain
/AdminProps
/AdminRealm
/AdminThreads
```

WordPress:

```
wp-admin/
wp-admin/about.php
wp-admin/admin-ajax.php
```

```
wp-admin/admin-db.php
wp-admin/admin-footer.php
wp-admin/admin-functions.php
wp-admin/admin-header.php
```

Tools (Araçlar)

- OWASP ZAP - Forced Browse, OWASP'nin önceki DirBuster projesinin şu anda devam eden bir kullanımıdır.
- THC-HYDRA, form tabanlı HTTP kimlik doğrulaması da dahil olmak üzere birçok arayüzün kabaca zorlanmasına izin veren bir araçtır.
- Kaba bir güç, iyi bir sözlük kullandığında, örneğin netsparker sözlük kullandığında çok daha iyidir.

References (Referanslar)

- Sirk: Varsayılan Şifre listesi
- FuzzDB, kaba kuvvet tarama yönetme yöneticisi giriş yolu yapmak için kullanılabilir
- Ortak yönetici veya hata ayıklama parametreleri