WEB UYGULAMASINA YÖNELİK DENETLEMELERDE LİNK KEŞFETME TEKNİKLERİ VE ARAÇLARI

Deniz Çevik, <denizcev at gmail dot com>, webguvenligi.org, 17/11/2008

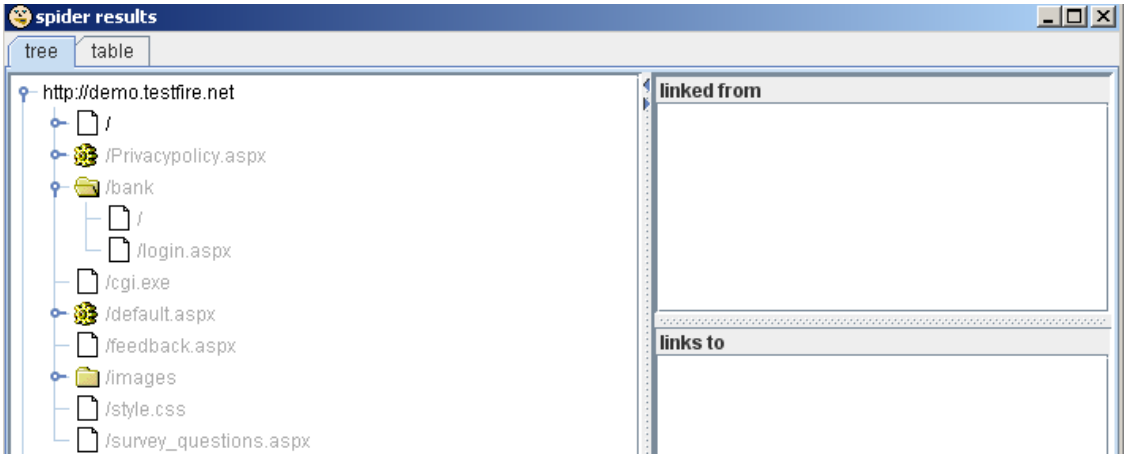
Güvenlik denetimlerinde kimi zaman sadece bir IP adresi veya FQDN verilerek, sunucu üzerindeki uygulamaların güvenlik denetimlerinin gerçekleştirilmesi istenilebilir. Bu çalışma sonucunda genellikle sistem hakkında herhangi bir bilgisi olmayan sıradan kullanıcıların neler yapabileceğinin, ne gibi bilgilere ulaşabileceğinin önceden ortaya çıkarılması hedeflenir.

Uygulama açılış sayfası bulunmayan, sunucu üzerindeki değişik dizinlerde web uygulaması barındıran, uygulamaya erişilecek linkin açıkça ortaya konmadığı sistemlerde güvenlik denetimini tam anlamı ile gerçekleştirebilmek için muhtemel zafiyet içeren linklerin, dizinlerin ve uygulamaların keşfi önem taşımaktadır. Bu amaçla kullanılabilecek yöntemlere ve araçlara aşağıda değinilmeye çalışılmıştır.

1- Otomatik Link Çıkartma (Crawling, Spidering)

Crawling, uygulama ana sayfasından başlayarak, uygulama HTML kaynak kodlarının incelenerek tüm linklerin çıkarılması işlemidir. İlk sayfadan çıkartılan linkler ziyaret edilerek, aynı inceleme yöntemi ile diğer başka linklere ulaşmak hedeflenir. İşlem her tespit edilen linkin ziyareti ve yenilerinin tespit edilmesi ile bir ağaç gibi dallanarak gider. Bunu sitenizdeki tüm sayfaları ve uygulamaları ziyaret eden bir internet gezginin oluşturduğu aktivite olarak düşünebilirsiniz.

Crawling mümkün olduğu kadar farklı uygulama ve parametre çıkartmak için kullanılabilecek en etkin yöntemlerden biridir. Link çıkartma işleminde HTML kodunda genellikle href, src, onclick, action gibi ifadeler kontrol edilir. Bunun yanında script içinde veya flash kullanılarak oluşturulmuş bir site içindeki linkleri çıkartmak için özel teknikler kullanmak gerekecektir. Web uygulama güvenliğini denetlemek için hazırlanmış pek çok ticari araç bu işlemi başarı ile gerçekleştirmektedir. Bunun yanında burpsuite, paros proxy, OWASP Webscarab gibi yazılımlar da bu amaçla kullanılabilecek ücretsiz programlardır.

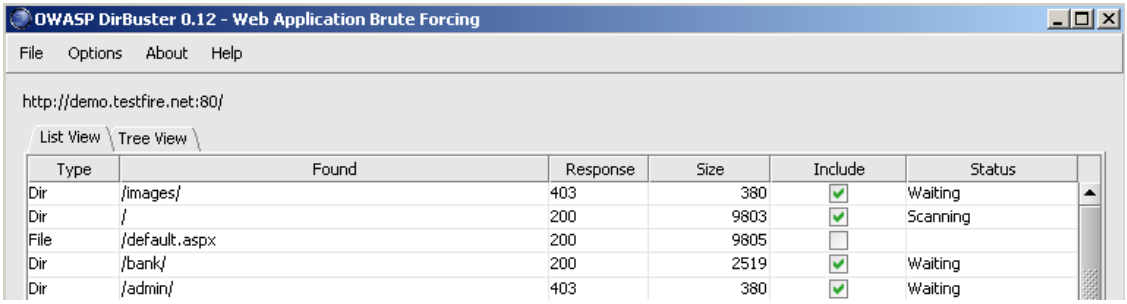


Test edilen sitenin büyük olması veya çok sayıda link barındırması durumunda crawling işleminin oldukça uzun süreceği göz ardı edilmemelidir.

2- Dizin Listeleri Kullanılması (Fuzzing, BruteForcer)

Bu yöntemde daha önceden oluşturulmuş, metin veya veritabanı dosyasında saklanan dizin ve dosya listeleri veya brute-force tekniği ile oluşturulan kelime dizileri sırayla sunucuya yollanır ve web sunucu hata mesajlarına bakılarak, yollanan dizinin veya dosyanın sistemde olup olmadığı tespit edilmeye çalışılır. Bu yöntem genellikle, crawling ile tespit edilemeyen, direkt erişim linki bulunmayan gizli dizinlerin veya uygulamaların çıkartılması amacıyla kullanılmaktadır. Ticari olarak dağıtılan web güvenlik tarayıcıları bu işlemi sadece en sık karşılaşılan dizin ve dosyalar için gerçekleştirmektedir. Bu sebeple kendi sözlüklerimizi oluşturmak ve geliştirmek bu yöntem ile daha etkili sonuçlar almamıza yardımcı olacaktır. İçinde güvenlik açıkları barındıran ve web sayfası üzerindeki linklerden ulaşılmayan uygulamalara bu yöntemle erişmek oldukça yaygın bir durumdur. Bu işlem için OWASP DirBuster, nikto, wfuzz, http-dir-enum gibi araçlar kullanılabilir. Metasploit Framework 3.2 üzerindeki bazı auxiliary modullerini kullanarak da benzer işlemler gerçekleştirmek mümkün olabilir.

scanner/http/wmap_brute_dirs	HTTP Directory Brute Force Scanner
scanner/http/wmap_dir_scanner	HTTP Directory Scanner
scanner/http/wmap_files_dir	HTTP Interesting File Scanner
scanner/http/wmap_file_same_name_dir	HTTP File Same Name Directory Scanner



Bu yöntemin en önemli dezavantajı, başarı oranının eldeki deneme listelerinin kalitesine bağlı olmasıdır. Çok büyük listeler ile kontrol yapılması durumunda bile halen keşfedilmeyen dizinler sunucu üzerinde bulunabilir. Ek olarak çok sayıda isteğin yollanması uzun zaman alabilir. Web sunucunun servis dışı kalmasına, mevcut bant genişliğinin hepsinin kullanılmasına yol açabilir. Bu sebeple optimum sonucu verecek sözcük listelerinin oluşturulması önem taşımaktadır. Bunun yanı sıra web sunucu hata mesajlarının değiştirilmesi ve her hata durumu için 200 OK veya 302 Redirect mesajı döndürülmesi gibi durumlarda hatalı sonuçlar ile karşılaşılmasına neden olabilir.

2- Alternatif, Yedek Dosyalar, Test Uygulamaları

Uygulama geliştiricilerinin yaptığı en yaygın hatalardan biri, uygulama yedeklerini, kopyalarını, uygulama eski sürümlerine ait dosyaları, test amaçlı geliştirdikleri uygulamaları web sunucu üzerindeki erişilebilir bölgelerde bırakmalarıdır. Bu durum güncel uygulamada her hangi bir güvenlik problemi olmamasına rağmen, eski sürümlerinde bulunabilecek açıklar sebebi ile sorunlara yol açabilir. Yine bu tip dosyalar uygulama kaynak kodlarına erişimi mümkün kılabilir. Test uygulamaları ise genellikle güvenlik gereksinimleri düşünülmeden tasarlandıkları için beklenen dışında sorunlara yol açabilirler. Dolayısı ile bu tip dosyaların belirlenmesi, güvenlik değerlendirmesinin tam anlamı ile gerçekleştirilebilmesi için önem taşımaktadır.

Alternatif, yedek veya test amaçlı oluşturulan dosyaları tespit etmek için genellikle mevcut uygulama dosya isimlerinin önüne ve arkasına eklemler yapmak, sık kullanılan uygulama isimlerini denemek gibi yöntemler kullanılır. Örneğin index.asp uygulaması için aşağıdaki gibi alternatif isimleri oluşturulabilir.

index.asp	index.asp_	index.asp~	eski/index.asp
index.asp.bak	index1.asp	index.old	tmp/index.asp
index.asp.yed	index2.asp	test.asp	index.asp.tmp
index.asp.yedek	index_1.asp	deneme.asp	index.tmp
index.asp.old	index.as_	test123.asp	
index.asp.old.1	_index.asp	index~1.asp	
index.asp.old.2	copy of index.asp	old/index.asp	

http-dir-enum, OWASP DirBuster gibi araçların yanı sıra Metasploit Framework 3.2 ile birlikte gelen aşağıdaki auxiliary modulleri ile de yukarıda anlatılan işlemler gerçekleştirilebilir.

scanner/http/wmap_backup_file	HTTP Backup File Scanner
scanner/http/wmap_prev_dir_same_name_file	HTTP Previous Directory File Scanner
scanner/http/wmap_replace_ext	HTTP File Extension Scanner

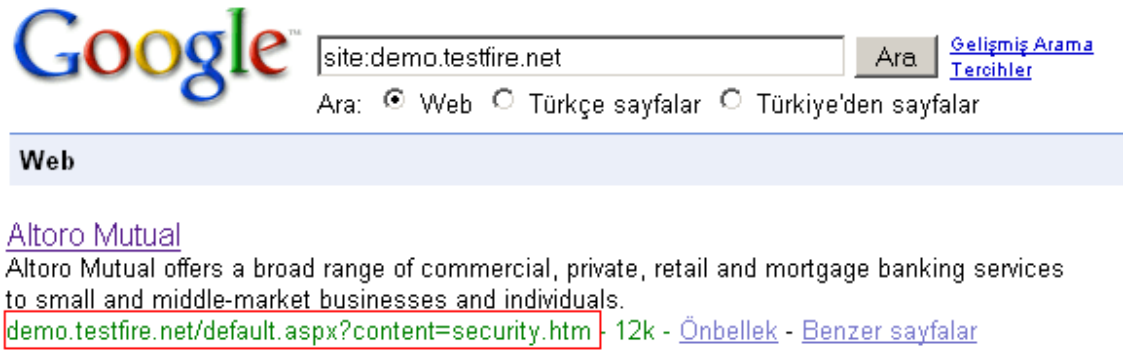
3- Benzer Sistemleri Kurarak Bilgi Toplama

Test edilen web sunucu yazılımının aynı sürümünü kendi sisteminize kurarak, kullandığı dizin yapısı, taşıdığı uygulamalar, yapılandırma bilgileri hakkında bilgi toplamak mümkün olacaktır. Bu sayede test edilen sistem üzerinde bulunabilecek muhtemel dizinler öğrenilebilir. Benzer yöntem açık kaynak kodlu uygulamalar kullanıldığının tespit edilmesi durumunda da kullanılabilir. Öğrenilen dizin ve uygulama bilgilerinden oluşturulan listeler sırayla sunucuya yollanarak, test edilen sistem üzerinde olup olmadığı veya direkt erişilmesi durumunda detaylı bilgiler açığa çıkaran hatalar alınıp alınmadığı kontrol edilebilir.

4- Arama Motorlarının Kullanılması

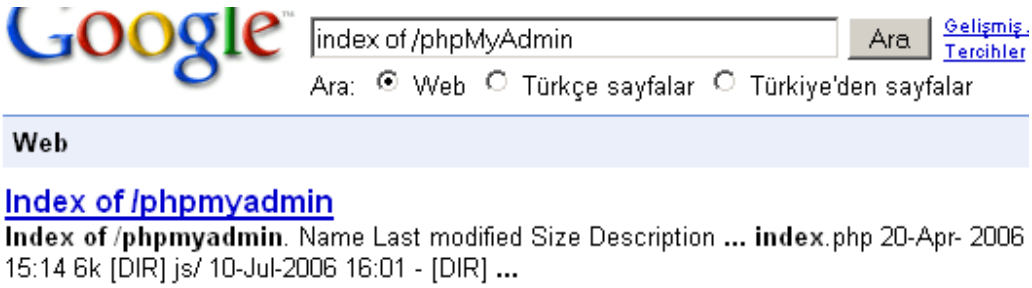
Arama motorları aracılığı ile kayıt altına alınan sayfalar sayesinde sunucu üzerinde bulunan dizin, link, uygulama adı gibi bilgilere erişmek mümkün olabilir. Özellikle sadece IP adresinin bilindiği ve ana sayfası bulunmayan sistemlerde IP adresinin veya sunucu adının sorgulanması neticesinde elde edilen sonuçlar, istenen bilgilere erişmemize yardımcı olacaktır. IP adresinin verilmesi ile gerçekleştirilen sorgular, uygulamaların yanı sıra o IP üzerinde çalışan ve sanal sunucu olarak tanımlanmış diğer web siteleri varsa bunların da ortaya çıkarılmasını sağlayacaktır.

http://search.msn.com/results.aspx?q=ip:65.61.137.117
http://www.google.com.tr/search?hl=tr&q=sdemo.testfire.net
http://www.google.com.tr/search?hl=tr&q=site:demo.testfire.net



Arama motorlarından tespit edilen dizin içeriğinde ne gibi dosyalarının bulunabileceğinin belirlenmesinde de yararlanılabilir. Örneğin dizin arama çalışmalarında phpMyAdmin dizini tespit edildi ve bu dizin içindeki uygulama linkleri belirlenmek isteniyor. Bu durumda ücretsiz olarak

dağıtılan bu yazılımı kendi sistemimize kurarak ne gibi dosyalar barındırdığını, dizin yapısını öğrenebileceğimiz gibi aynı uygulamayı kendi sistemine yüklemiş fakat dizin içerik listeleme özelliğini kapatmamış olan, arama motorları tarafından kayıt altına alınmış sitelerden de yararlanabiliriz. Bu işlem için "index of /phpMyAdmin" gibi bir arama ifadesi google aracılığı ile kullanılabilir.



Benzer işlemleri otomatik olarak gerçekleştiren için seat ve maltego gibi araçlardan yararlanılabilir.

5- HTML Kodlarındaki Açıklama Alanları

Uygulama isminin veya bulunduğu yolun değişmesi gibi durumlarda genellikle eski koda kolay dönüş açısından açıklama alanı haline getirilmesi sıklıkla kullanılan bir yöntemdir. Açıklama alanlarında yer alan linklerin ve dizinlerin takibi bize güvenlik açığı taşıyabilecek uygulamalara erişme olanağı tanıyabilir.

```
<!--<a href=eski/login.aspx  title="Login Page" img src="img/a1.jpg"
width="3px" height="6px" title="Login Page" style="margin:0;" /></a>-->
```

Bir firefox bileşeni olan webdeveloper gibi araçlarla açıklama alanlarının kontrol edilmesi, uygulama üzerinde direkt linki bulunmayan sayfa, uygulama ve dizinleri ortaya çıkarma açısından faydalı olacaktır.

6- Robots.txt Dosyası Yardımı İle Link Tespit Etme

Arama motorlarının indeksleme robotları bir web sitesine ulaştıklarında ilk olarak kök dizinde bulunan robots.txt dosyasını ararlar. Bu dosya, robotlara hangi sayfaların indekslenip hangilerinin indekslenmeyeceğini anlatmak için yerleştirilir. Bununla birlikte detaylı olarak hazırlanmış robots.txt dosyası web sunucu üzerindeki dosya yapısı hakkında da detaylı bilgi sunabilir.

User-agent: *

Disallow: /admin/
Disallow: /sourcecodes/
Disallow: /secret.php

Link keşfi gerçekleştirilirken sunucu üzerinde robots.txt dosyasının varlığının kontrol edilmesinin iyi bir fikir olduğunu söylemek yanlış olmayacaktır.

7- Arşiv Sitelerinin Kullanılması

Link tespit işleminde www.archive.org gibi, web sunucularının değişik tarihlerdeki hallerini barındıran sitelerden de yararlanılabilir. Arşiv siteleri aracılığı ile halen sunucu üzerinde bulunabilecek eski uygulamalar erişmek mümkün olabilir.

8- Sunucu ve Uygulama Hata Mesajları

Sunucu ve uygulama hata mesajları çoğu zaman gerektiğinden fazla bilginin açığa çıkmasına neden olmaktadır. Bu bilgiler arasında include edilmiş diğer uygulamalara ait yol bilgiler bulunabilir. Bu bilgilerden yararlanarak yeni sayfalar keşfetmek, include dosyalarının içeriğine ulaşmak mümkün olabilir.

Fatal error: Call to a member function get_cellmap() on a non-object in **/home/benj/projects/dompdf-0.5/include/table_cell_frame_reflower.cls.php** on line **64**