4.2 BİLGİ TOPLAMA

Güvenlik değerlendirmesinde ilk faz, hedef uygulama hakkında bütün bilgileri toplamaya odaklıdır. Bilgi toplama penetrasyon testinin gerekli bir adımıdır. Bu görev bir çok değişik yol kullanılarak yapılabilir.
Herkese açık araçları (arama motorları), tarayıcıları, basit HTTP istekleri veya özel olarak hazırlanmış istekleri kullanarak, uygulamayı kullanılan sürüm ve teknolojileri açığa vuracak hata mesajlarını ele vermek suretiyle bilgi sızdırmaya zorlamak mümkündür.

Uygulama sunucuda veya web sunucudaki varsayılı kötü bir yapılandırmanın sonucu olarak, çoğu zaman, yapılandırmada veya kötü sunucu yönetimi hakkında zayıflıkları açığa çıkarabilecek uygulamadan alacağınız bir cevap ile bilgi toplamak mümkündür.

Uygulama Sürüm Analizi

Uygulama sürüm analizi, versiyon saptama adına Bilgi Toplama işleminde ilk adımdır ve koşan web sunucu tipi denetçilerin bilinen açıklıkları bulmasına ve değerlendirme süresince uygun gerçeklenmiş açıklıkları kullanmasına izin verir.

Uygulama Keşfi

Uygulama keşfi bir web/uygulama sunucuda koşan uygulamaların belirlenmesi ile ilgili bir aktivitedir.
Bu önemli bir analizdir çünkü bir çok kez uygulama arka ucuna bağlı direk bir link yoktur. Bu nedenle keşif analizi yönetici arayüzleri, dosyaların veya başka ürünlerin (test/geliştirme aşamasında veya yürütme esnasında oluşturulan ve kullanıldıktan sonra tam olarak silinmeyen betikler gibi) eski versiyonları gibi önemli detayları bulmak için faydalıdır.

Site Link Çıkarma (Spidering) ve "Google"lama

Bilgi Toplama işlemini bu aşaması değerlendirilen uygulamaya ait kaynakların taranması ve ele geçirilmesinden oluşmaktadır. Google gibi arama motorları, web uygulamasının yapısı ile ilgili konuları veya uygulamanın ürettiği hata sayfaları keşfetmek için kullanılabilirler.

Hata kodlarının analizi

Web uygulamaları, son kullanıcının görmemesi gereken hata mesajlarını bir penetrasyon testinde verebilirler. Bilgi (hata kodları gibi), denetçiyi uygulama tarafından kullanılan teknoloji ve ürünler hakkında bilgilendirir.
Bu tür hata kodlarını yetersiz hata yönetimi stratejisi sebebiyle elde etmek kolay olabilir.

Altyapı yapılandırma yönetimi testi

Altyapı ve topoloji mimarisi analizi, çoğu zaman bir web uygulama hakkında bir çok bilgi verir; kaynak kodu, izin verilen HTTP metotları, yönetici fonksiyonelliği, kimlik doğrulama metotları ve altyapısal yapılandırmalar.
Bu nedenlerden dolayı sadece web uygulama odaklı bir test kapsayıcı olamaz. Çünkü herhangi bir güvenlik değerlendirmesinde (pentest) elde edilen bilgiler daha geniş anlamda bir altyapı analizi ile elde edilebilecek bilgiler kadar kapsayıcı olamazlar.

SSL/TLS Testi

SSL ve TLS, kriptografi desteği ile bilginin gizliliğini ve kimliğini korumak adına güvenli bir iletişim sağlayan iki protokoldürler. Bu güvenlik gerçeklemelerinin önemi düşünülürse, güçlü bir şifreleme algoritmasının düzgün bir şekilde kullanılıyor olmasını kontrol etmek önemlidir.

VT Dinleyici (DB Listener) Testi

Bir veritabanının yapılandırma aşamasında, bir çok VT yöneticisi DB dinleyici bileşeninin güvenlik eksiğini dikkate almaz. Dinleyici hassas bilgileri açığa vurabileceği gibi yapılandırma ayarlarını veya koşan veritabanlarını ele verebilir.
Bu bilginin toplanması, saklanılan verinin gizliliğini, bütünlüğünü ve kullanılabilirliğini zedeleyecek işe yarar tüyolar sağlayabilir.
VT dinleyici yapılandırması üzerindeki dikkatli bir güvenlik analizi bu bilgiye ulaşmaya izin veriri. ?

Uygulama yapılandırma yönetim testi

Web uygulamaları, geliştirme aşamasında genellikle düşünülmeyen bazı bilgileri veya uygulamanın kendi yapılandırmasını gizlerler.
Bu veriler kaynak kodunda, log dosyalarında veya web sunucuların varsayılı hata kodlarında keşfedilebilir. Bu nedenle bu konuda doğru bir yaklaşım bir güvenlik değerlendirmesinde temeldir.

Dosya uzantısı yönetimi

Bir web sunucuda veya uygulamada dosya uzantılarını gözleyerek, hedef uygulamayı oluşturan teknolojileri ve bazen uygulamaya bağlı diğer sistemler belirlenebilir (mesela sunucu tarafında jsp ve asp uzantıları).

Eski, yedek ve referans verilmeyen dosyalar

Herkes tarafından indirilebilen ve okunabilen sunucudaki gereksiz dosyalar (eski, yedek veya ismi değiştirilmiş dosyalar) büyük bilgi kaynağıdırlar. Bu dosyaların varlığını kontrol etmek gereklidir çünkü kaynak kodun bi kısmını, kurulum yollarını veya uygulamalar/veritabanları için şifreleri içerebilirler.

4.2.1 WEB UYGULAMA SÜRÜM ANALİZİ İÇİN TEST

KISA ÖZET

Web sunucu sürüm analizi değerlendirici için kritik bir görevdir. Sürümü ve koşan sunucu tipini bilmek denetçiye bilinen açıklıkları bulmasına ve uygun olanlarını kullanmasına izin verir.

KONU AÇIKLAMASI

Günümüzde bir çok farklı web sunucu versiyonu ve sağlayıcısı vardır. Değerlendirdiğiniz web sunucunun tipinin bilinmesi test sürecinde çok önemli derecede yardımcı olur ve testin yönünü değiştirir. Bu bilgi web sunucuya belirli komutların gönderilmesi ve cevabın analiz edilmesi ile elde edilir. Çünkü web sunucu yazılımının her versiyonu bu komutlara değişik cevap verebilir.

Web sunucu yazılımının her versiyonunun belirli komutlara nasıl cevap verdiğini bilmesi ve bunu bir web sunucu sürüm analizi veritabanında tutması ile, denetleyici bu komutları web sunucuya gönderir ve cevapları veritabanındaki imzalar ile karşılaştırır. Lütfen dikkat edin web sunucuyu kesin bir şekilde belirlemek genellikle bir çok komut gerektirir. Çünkü aynı komut için değişik web sunucuları benzer cevaplar dönebilirler. Ancak nadiren, iki farklı sürüm bütün HTTP komutlarına aynı cevabı döner. Yani, bir çok komut göndererek, tahminizi güçlendirirsiniz.

KARA KUTU TESTİ VE ÖRNEK

Bir web sunucusunu belirlemenin en kolay ve temel yolu HTTP cevabının başlığındaki *Server* alanına bakmaktır. Denemelerimiz için netcat kullanacağız. Aşağıdaki HTTP İstek-Cevabı düşünün;

```
$ nc 202.41.76.251 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 16 Jun 2003 02:53:29 GMT
Server: Apache/1.3.3 (Unix)  (Red Hat/Linux)
Last-Modified: Wed, 07 Oct 1998 11:18:14 GMT
ETag: "1813-49b-361b4df6"
Accept-Ranges: bytes
Content-Length: 1179
Connection: close
Content-Type: text/html

$
```

Server alanından sunucunun Apache 1.3.3 olduğunu ve bir Linux işletim sistemi üzerinde koştuğunu anlıyoruz. Üç HTTP cevap başlığı örnekleri aşağıda gösterilmektedir:

Apache 1.3.23 sunucusundan:

```
HTTP/1.1 200 OK
Date: Sun, 15 Jun 2003 17:10: 49 GMT
Server: Apache/1.3.23
Last-Modified: Thu, 27 Feb 2003 03:48: 19 GMT
ETag: 32417-c4-3e5d8a83
Accept-Ranges: bytes
Content-Length: 196
Connection: close
Content-Type: text/HTML
```

Microsoft IIS 5.0 sunucusundan:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Expires: Yours, 17 Jun 2003 01:41: 33 GMT
Date: Mon, 16 Jun 2003 01:41: 33 GMT
Content-Type: text/HTML
Accept-Ranges: bytes
Last-Modified: Wed, 28 May 2003 15:32: 21 GMT
ETag: b0aac0542e25c31: 89d
Content-Length: 7369
```

Netscape Enterprise 4.1 sunucusundan:

```
HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Mon, 16 Jun 2003 06:19: 04 GMT
Content-type: text/HTML
Last-modified: Wed, 31 Jul 2002 15:37: 56 GMT
Content-length: 57
Accept-ranges: bytes
Connection: close
```

Ama bu test metodolojisi çok iyi değildir. Bir web sitesinin kullandığı sunucu başlığını değiştirecek bir çok teknik vardır. Mesela aşağıdaki cevabı alabilirdik:

```
403 HTTP/1.1
Forbidden Date: Mon, 16 Jun 2003 02:41: 27 GMT
Server: Unknown-Webserver/1.0
Connection: close
Content-Type: text/HTML;
```


```
charset=iso-8859-1
```

Bu durumda cevaptaki Server alanı değiştirilmiştir: Koşan sunucunun hangi tip olduğunu bilemeyiz.

Protokol davranışı

İyileştirilmiş denetleme tekniği, piyasadaki bir çok web sunucusunun değişik karakterlerini dikkate alır. Kullanılan web sunucunun tipini anlamaya yarayan bazı metodolojileri listeleyeceğiz.

HTTP başlık alanı sıralaması

İlk metot cevaptaki bir çok başlığın sıralamasını izlemektir. Her web sunucusunun kendine has bir başlık sıralaması vardır. Örnek olarak aşağıdaki cevapları düşünüyoruz:

Apache 1.3.23 sunucusundan cevap:

```
$ nc apache.example.com 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 15 Jun 2003 17:10: 49 GMT
Server: Apache/1.3.23
Last-Modified: Thu, 27 Feb 2003 03:48: 19 GMT
ETag: 32417-c4-3e5d8a83
Accept-Ranges: bytes
Content-Length: 196
Connection: close
Content-Type: text/HTML
```

IIS 5.0 sunucusundan cevap:

```
$ nc iis.example.com 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: http://iis.example.com/Default.htm
Date: Fri, 01 Jan 1999 20:13: 52 GMT
Content-Type: text/HTML
Accept-Ranges: bytes
Last-Modified: Fri, 01 Jan 1999 20:13: 52 GMT
ETag: W/e0d362a4c335be1: ae1
Content-Length: 133
```

Netscape Enterprise 4.1 sunucusundan cevap:

```
$ nc netscape.example.com 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Netscape-Enterprise/4.1
Date: Mon, 16 Jun 2003 06:01: 40 GMT
Content-type: text/HTML
Last-modified: Wed, 31 Jul 2002 15:37: 56 GMT
Content-length: 57
Accept-ranges: bytes
Connection: close
```

Apache, Netscape Enterprise ve IIS arasında, *Date* başlığının sıralamasının ve *Server* başlığının farklı olduğunu fark edebiliriz.

Değiştirilmiş istek testi

Çalıştırılacak diğer bir faydalı test, değiştirilmiş istekleri veya olmayan sayfa isteklerini sunucuya göndermektir. Aşağıdaki cevabı ele alalım:

Apache 1.3.23 cevabı:

```
$ nc apache.example.com 80
GET / HTTP/3.0

HTTP/1.1 400 Bad Request
Date: Sun, 15 Jun 2003 17:12: 37 GMT
Server: Apache/1.3.23
Connection: close
Transfer: chunked
Content-Type: text/HTML; charset=iso-8859-1
```

IIS 5.0 cevabı:

```
$ nc iis.example.com 80
GET / HTTP/3.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: http://iis.example.com/Default.htm
Date: Fri, 01 Jan 1999 20:14: 02 GMT
Content-Type: text/HTML
Accept-Ranges: bytes
Last-Modified: Fri, 01 Jan 1999 20:14: 02 GMT
ETag: W/e0d362a4c335be1: ae1
Content-Length: 133
```

Netscape Enterprise 4.1 cevabı:

```
$ nc netscape.example.com 80
GET / HTTP/3.0

HTTP/1.1 505 HTTP Version Not Supported
Server: Netscape-Enterprise/4.1
Date: Mon, 16 Jun 2003 06:04: 04 GMT
Content-length: 140
Content-type: text/HTML
Connection: close
```

Bütün sunucuların farklı cevapladıklarını görüyoruz. Cevap aynı zamanda sunucuların versiyonlarında da değişiyor. Aynı durum olmayan sayfalara istek gönderdiğimizde de açığa çıkar. Aşağıdaki cevapları ele alın:

Apache 1.3.23 cevabı:

```
$ nc apache.example.com 80
GET / JUNK/1.0

HTTP/1.1 200 OK
Date: Sun, 15 Jun 2003 17:17: 47 GMT
Server: Apache/1.3.23
Last-Modified: Thu, 27 Feb 2003 03:48: 19 GMT
ETag: 32417-c4-3e5d8a83
Accept-Ranges: bytes
Content-Length: 196
Connection: close
Content-Type: text/HTML
```

IIS 5.0 cevabı:

```
$ nc iis.example.com 80
GET / JUNK/1.0

HTTP/1.1 400 Bad Request
```

```
Server: Microsoft-IIS/5.0
Date: Fri, 01 Jan 1999 20:14: 34 GMT
Content-Type: text/HTML
Content-Length: 87
```

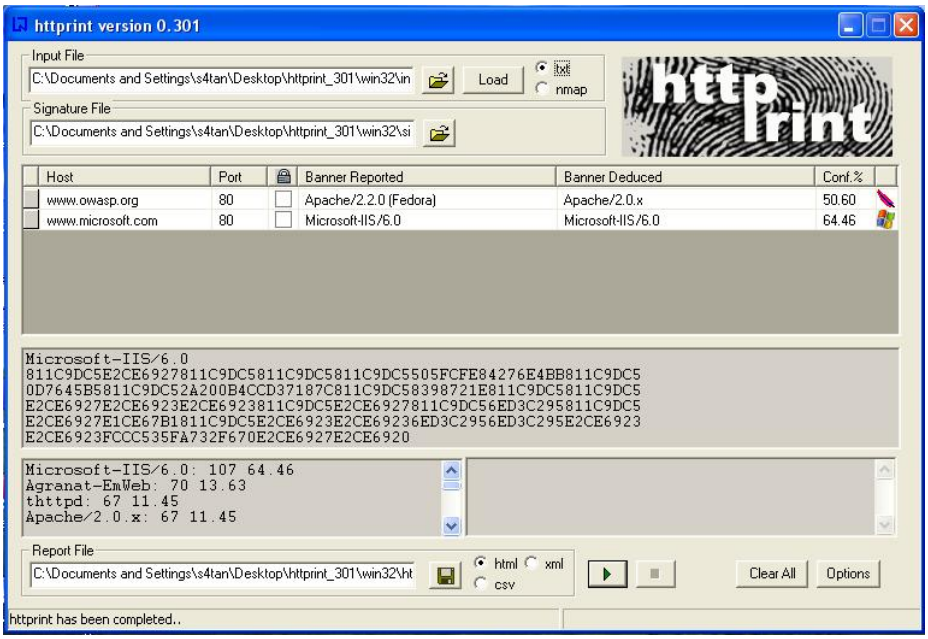
Netscape Enterprise 4.1 cevabı:

```
$ nc netscape.example.com 80
GET / JUNK/1.0

<HTML><HEAD><TITLE>Bad request</TITLE></HEAD>
<BODY><H1>Bad request</H1>
Your browser sent to query this server could not understand.
</BODY></HTML>
```

Otomatikleştirilmiş Test

Denetlemeyi gerçekleştirmek için kullanılan testler değişebilir. Bu testleri otomatize edecek bir araç ""httprint""dir. Bir imza veritabanı sayesinde kullanılan web sunucunun versiyonunu ve tipini anlayabilir. Bu aracın bir örneği aşağıda gösterilmektedir:



KAYNAKLAR

Makaleler

- Saumil Shah: "An Introduction to HTTP fingerprinting" - http://net-square.com/httprint/httprint_paper.html

Araçlar

- httprint - http://net-square.com/httprint/index.shtml

4.2.2 UYGULAMA KEŞFİ

KISA ÖZET

Web uygulaması açıklık değerlendirmesinde en önemli adımlardan biri, bir sunucuda hangş uygulamaların koştuğunun bulunmasıdır.
Bir çok uygulamanın uzaktan kontrol ve/veya veri hırsızlığı uygulanabilecek bilinen açıklıkları ve bilinen saldırı stratejileri vardır.

Buna ek olarak, bir çok uygulamalar web sunucularındaki ana uygulama/site üzerinden direk referans verilmeden koşarlar: bu yanlış yapılındırılan veya "sadece dahili" olarak kullanıldığına inanılarak yenilenmemiş, dahili ve/veya harici web siteleri için geçerlidir. Dahası, bir çok uygulamalar yönetici arayüzleri için bilinen yolları kullanılırlar ve bu arayüzler tahmin etme veya kaba kuvvet saldırılarına uğrayabilirler.

KONU AÇIKLAMASI

Sanal web sunucuların artması ile, IP adresi ve web sunucusu arasındaki geleneksel 1:1 tipi ilişkisi eski önemini kaybetmektedirler. Sembolik isimlerinin (alan ismi) aynı IP adresine çözüldüğü bir çok web sitesini / uygulamasını görmek nadir karşılaşılan bir durum değildir (ve bu senaryo sadece servis veren yerler için sınırlı değildir, aynı zamanda kurumsal yerlerde de görülebilirler).

Bir güvenlik uzmanı olarak size değerlendirmek için hedef olarak bir IP adresi kümesi veya sadece bir IP adresi verilir. Başka bir bilgi yoktur. Bunun bir penetrasyon test tipine daha çok benzediği tartışılabiilr, ama her durumda bu görev belirlenen hedefte ulaşılabilen bütün web uygulamalarının (ve muhtemelen diğer şeylerin) değerlendirmeleri umulur. Problem, verilen IP adresleri 80inci port üzerinde http servisi verebilir ama eğer servislere IP adresi vererek ulaşırsanız "Bu adreste web sunucu yapılandırması yoktur" veya buna benzer bir hata alırsınız. Ama bu sistem ilgili olmayan sembolik (DNS) isimlerde bir çok web uygulamayı saklıyor olabilir. Tabi ki, analizinizin bütünlüğü uygulamaları test edip etmediğiniz ile derinden alakalıdır, çünkü uygulamaların hepsinin veya sadece bazılarının farkında olmazsınız. Bazen hedef spesifikasyonu daha zengindir, belki bir IP adresi listesi ve ilgili DNS isimlerini hedef olarak alabilirsiniz. Ancak, bu liste tam olmayan bilgileri içerebilir, yani bazı sembolik isimleri içermeyebilir ve müşterinin bundan haberi bile olmayabilir (bu genelde büyük organizasyonlarda geçerlidir.).
Değerlendirmenin kapsamını etkileyen diğer bir konu, başka hiçbir yerde belirtilmeyen ve bilinmeyen URL'lerde yayınlanan web uygulamalarıdır (http://www.example.com/ilgiç-bir-URL). Bu bir hatadan (yanlış yapılandırmalar) veya bilerek (mesela, yayınlanmayan yönetim arabirimleri) olabilir.
Bu konuyu çözmek için, web uygulama keşfi yapmak mecburidir.

KARA KUTU TESTİ VE ÖRNEK

Web uygulama keşfi
Web uygulama keşfi verilen bir altyapı üzerindeki web uygulamalarını belirlemek üzerine bir süreçtir. Altyapı genellikle bir IP adresi kümesi (belki bir net bloğu şeklinde) olarak belirtilir, ama aynı zamanda bir DNS sembolik isim kümesi olarak da verilebilir, veya ikisinin ortası.
Bu bilgi ister klasik penetrasyon testi olsun veya uygulama odakli bir test olsun, değerlendirme öncesi verilir. Her iki durumda da, eğer değerlendirme kuralları tersini söylemiyorsa, değerlendirme en kapsamlı olan testi yapmalıdır, yani ilk olarak verilen hedefteki bütün uygulamaları bulmalıdır. Takip eden örnekte, bu hedefi gerçeklemek için kullanılabilen bir kaç

tekniği analiz edeceğiz.
Aşağıdaki tekniklerden bazıları, DNS ve web tabanlı ters IP arama servisleri olarak adlandırılan Internet taraflı web sunuculara ve arama motorlarını kullanmaya yöneliktir. Örnekler gizlilik yüzünden özel IP adreslerini (192.168.1.100 gibi) kullanmaktadır.

Verilen bir DNS ismine (veya IP adresine) kaç tane uygulama düştüğüne etki eden iki factor vardır.

1. Farklı temel (base) URL
Bir web uygulama için belli giriş noktası www.example.com'dır. Yani, bu kısaltma notasyonu ile web uygulamasının http://www.example.com/ adresinden geldiğini düşünürüz (aynı durum https için de geçerlidir). Ancak, bu en yaygın durum olmasına rağmen, uygulamanın "/" dizini ile başlamasını zorlayıcı bir durum yoktur.
Örnek olarak, aynı sembolik isim (alan adı) üç uygulama ile ilişkilendirilebilirdi, mesela;

```
http://www.example.com/url1
http://www.example.com/url2
http://www.example.com/url3
```

Bu durumda, http://www.example.com/ URLsi mantıklı bir sayfaya denk düşmeyecektir ve üç uygulama eğer nasıl ulaşılacağını (url1, url2 veya url3) göstermezsek gizli duruma düşeceklerdir. Genellikle uygulamaları bu şekilde yayınlamanın gereği yoktur, tabi uygulamalarınıza standard bir şekilde ulaşılmasını istemiyorsanız ve kullanıcılarınıza uygulamalara nasıl ulaşacakları hakkında bilgi göndermeye hazırlıklıysanız. Bu durum, bu uygulamaların gizli olduğu anlamına gelmez ama var olmaları ve kaynakları açık bir şekilde belirtilmedikleri anlamına gelir.

2. Standard olmayan portlar
Her nekadar Web uygulamaları genellikle 80 (http) ve 443 (https) portlarında koşsalar da, bu port numaraları çok önemli değildir. Aslında, web uygulamaları herhangi TCP portu ile ilişkilendirilebilirler ve şu şekilde port numarası bildirerek referans edilebilirler: http[s]://www.example.com:port/. Mesela, `http://www.example.com:20000`
Bir IP adresinin kaç web uygulaması ile ilişkili olduğunu etkileyen başka bir faktör vardır.

3. Sanal sunucular (hosts)
DNS, bir IP adresini birden fazla sembolik isim ile ilişkilendirmemize izin verir. Mesela, *192.168.1.100* IP adresi *www.example.com, helpdesk.example.com, webmail.example.com* DNS isimleri ile ilişkilendirilebilir (aslında, DNS isimlerinin aynı alan adına da ait olmaları gerekmez). Bu 1:N ilişkisi sanal sunucular olarak adlandırılan teknoloji kullanılarak farklı içeriklerin yayınlanmasını sağlar. İstediğimiz sanal sunucuyu belirleyen bilgi HTTP 1.1 *Host:* başlığı içindedir [1]. *helpdesk.example.com* ve *webmail.example.com* isimlerinden haberdar değilsek, *www.example.com* alan adında yaşayan uygulama dışında başka bir uygulama olabileceği ile ilgili şüphe duymayız.

1inci konu ile ilgili yaklaşımlar – standard olmayan URLler
Standard isimleri olmayan web uygulamalarının varlığını kesin olarak kestirmenin hiç bir yolu yoktur. Standard olmadıkları için, bu web uygulamalarını ortaya çıkarmak için sihirli bir reçete yoktur. Ancak, bu yolda bize yardım edebilecek bir kaç kriter kullanabiliriz.

İlk olarak, eğer web sunucusu yanlış yapılandırılmış ve dizin gezmeye (directory browsing) izin vermekteyse, bu uygulamaları bulmak mümkün olabilir. Açıklık tarayıcıları bu durumda yardımcı olabilirler.
İkinci olarak, bu uygulamalar diğer web sayfalarından referans verilmiş olabilirler; mesela, önceden linkleri çıkarılmış ve web arama motorları tarafından indekslenmiş olabilirler. *www.example.com'da* Bu şekilde "gizlenmiş" web uygulamalarından şüpheleniyorsak, site operatörünü kullanarak ile bir miktar googlelama yapabilir ve "site: www.example.com" isteğinin

sonuçlarını analiz edebiliriz. Dönen URL'ler arasından standard olarak gösterilmeyen uygulamaya giden bir link olabilir. Diğer bir seçenek, yayınlanmamış uygulamalar için aday olabilecek URL'leri istek olarak göndermek olabilir. Mesela, bir web tabanlı e-posta ön arayüzü https://www.example.com/webmail ile ulaşılabilir. Aynı durum standard URL'ler ile yayınlanan yönetim arayüzleri için geçerlidir (mesela bir Tomcat yönetici arayüzü) ve bu URL'ye başka bir yerde referans verilmemiş olabilir. Öyleyse, sözlük tipi aramalar (veya "akıllı tahminler") bazı sonuçlar doğurabilir. Bu açıdan açıklık tarayıcıları yardımcı olabilirler.

2inci konu ile ilgili yaklaşımlar – standard olmayan portlar

Standard olmayan portlar üzerindeki web uygulamalarının varlığını kontrol etmek kolaydır. Nmap gibi bir port tarayıcının [2] –sV opsiyonu ile servis tanımlaması mümkündür ve farklı portlarda http[s] servislerini bulabilir. Tek yapılması gereken bütün 64k TCP port adres uzayının taranmasıdır.

Mesela aşağıdaki komut, TCP connect taraması kullanarak, *192.168.1.100* IP adresindeki bütün açık portlara bakacak ve üzerlerinde hangi servislerin koştuğunu saptamaya çalışacaktır (sadece gerekli opsiyonlar gösterilmiştir, nmap bir çok seçenek ile çalışabilmektedir).

```
nmap –P0 –sT –sV –p1-65535 192.168.1.100
```

Çıktıyı incelemek ve http veya SSL ile ilgili servislerin varlığı için araştırmak yeterlidir. Mesela, bir önceki komutun çıktısı şunun gibidir;

Interesting ports on 192.168.1.100:

(The 65527 ports scanned but not shown below are in state: closed)

```
PORT      STATE SERVICE     VERSION
22/tcp    open  ssh         OpenSSH 3.5p1 (protocol 1.99)
80/tcp    open  http        Apache httpd 2.0.40 ((Red Hat Linux))
443/tcp   open  ssl         OpenSSL
901/tcp   open  http        Samba SWAT administration server
1241/tcp  open  ssl         Nessus security scanner
3690/tcp  open  unknown
8000/tcp  open  http-alt?
8080/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
```

Bu örnekten, 80inci port üzerinde bir Apache web sunucusu koştuğunu görüyoruz.

443üncü port üzerinde bir https sunucusu olduğu gözüküyor (ama bu kontrol edilmeli; bir tarayıcı ile https://192.168.1.100 ziyaret edilerek mesela).

901inci portta Samba SWAT web arayüzü var

1241inci port üzerindeki servis https değil, ama SSL ile kullanılan bir Nessus servisi

3690 portu bilinmeyen bir servis koşuyor (nmap, sadelik için kaldırılmış, parmak izini geri veriyor. Ayrıca nmap bunu parmak izi veritabanına nasıl göndereceğiniz konusunda bilgiler veriyor, tabi gerçekten hangi servis olduğunu biliyorsanız.)

8000inci port üzerinde belli olmayan diğer bir servis; muhtemelen http olabilir, çünkü bu port üzerinde http servislerini görmek nadir karşılaşılan bir olay değil: Şöyle bi bakalım:

```
$ telnet 192.168.10.100 8000
Trying 192.168.1.100...
Connected to 192.168.1.100.
Escape character is '^]'.
GET / HTTP/1.0
```

```
HTTP/1.0 200 OK
pragma: no-cache
Content-Type: text/html
Server: MX4J-HTTPD/1.0
expires: now
Cache-Control: no-cache

<html>
...
```

Bu servisin gerçekten bir HTTP sunucusu olduğunu gösteriyor. Alternatif olarak, URL'yi bir web tarayıcı ile ziyaret edebilirdik; veya yukarıdakini HTTP etkileşimini taklit edebilecek GET veya HEAD Perl komutlarını kullanabilirdik.

```
Apache Tomcat running on port 8080
```

Aynı görev, açıklı tarama araçları ile de yapılabilir – ama önce tarayıcınızın standard olmayan portlarda çalışan http[s] servisini bulabileceğinden emin olun. Örnek olarak, Nessus [3] bu servisleri farklı portlar üzerinde bulabilir ve bilinen web sunucu açıklıkları ve aynı zamanda https servislerinin SSL yapılandırmaları ile ilgili bir kaç testi üzerlerinde uygulayabilir. Daha önce de belirtildiği gibi, Nessus başka türlü fark edilemeyecek popüler uygulamaları/web arayüzlerini bulabilmektedir (mesela Tomcat'ın web arayüzünü).

3üncü konu ile ilgili yaklaşımlar – sanal sunucular

Verilen bir *x.y.z.t* IP adresine bağlı DNS isimlerini bulmak için kullanılabilen bir takım teknikler vardır.

DNS zone transferleri

DNS sunucuları tarafından zone transferlerine genel olarak izin verilmediğinden bu tekniğin günümüzde kısıtlı kullanımı vardır. Ama denemekde fayda vardır.
İlk olarak, *x.y.z.t* hizmetini veren isim sunucularını bulmamız gerekmektedir. Eğer *x.y.z.t* için bir sembolik isim biliniyorsa (diyelim ki www.example.com olsun), isim sunucuları DNS NS kayıtlarını sorgulayarak nslookup, host gibi araçlar yardımı ile bulunabilir.
Eğer *x.y.z.t* için hiç sembolik isim bilinmiyorsa, ama hedef tanımınız en azından bir tane sembolik isim içeriyorsa, aynı süreci uygulayabilir ve o isme ait isim sunucusunu sorgulayabilirsiniz (*x.y.z.t'nin de* bu isim sunucusu tarafından servis edildiğini umarak). Mesela, eğer hedefiniz *x.y.z.t* IP adresinden ve *mail.example.com*'den oluşuyorsa, isim sunucularını *example.com* alanı için bulun.

Örnek: host kullanarak www.owasp.org isim sunucularını belirlemek

```
$ host -t ns www.owasp.org
www.owasp.org is an alias for owasp.org.
owasp.org name server ns1.secure.net.
owasp.org name server ns2.secure.net.
$
```

Daha sonra, *example.com* alan adı için isim sunucularına bir zone transfer isteği yollanabilir; eğer şanslı iseniz bu alan adı için bir isim listesi geri alırsınız. Buliste bilinen *www.example.com* ve bilinmeyen *helpdesk.example.com* ve *webmail.example.com* adreslerini içerir. Zone transferi sonucu dönen bütün isimleri kontrol edin ve denetlenen hedef ile ilgili olanları belirleyin.

owasp.org için isim sunucularının birinden bir zone transferi istemek

```
$ host -l www.owasp.org ns1.secure.net
```

```
Using domain server:
Name: ns1.secure.net
Address: 192.220.124.10#53
Aliases:

Host www.owasp.org not found: 5(REFUSED)
; Transfer failed.
-bash-2.05b$
```

Ters DNS sorguları

Bu süreç bir önceki ile benzerdir ama ters DNS kayıtlarına (PTR) dayanır. Bir zone transferi istemek yerine, kayıt tipini PTR'a eşitlemeyi ve verilen IP adresine bir sorgu göndermeyi deneyin. Eğer şanslı iseniz, bir DNS ismi kaydı alabilirsiniz. Bu teknik IP'den sembolik isim ilişkisinin varlığına dayanır ve kesin değildir.

Web-tabanlı DNS aramaları

Bu tür bir arama, DNS zone transferine benzer, ama DNS üzerinde isim araması yapmaya izin veren web tabanlı servisleri kullanır. Bu tür bir servis *Netcraft Search DNS* servisidir ve http://searchdns.netcraft.com/?host adresinde bulunabilir. İstediğiniz bir alana ait olan bir isim listesini sorgulayabilirsiniz. Daha sonra bulduğunuz isimlerin hedefinize ait olup olmadığını control edersiniz.

Ters-IP servisleri

Ters-IP servisleri ters DNS sorgularına benzerler, isim sunucusu yerine web tabanlı bir uygulamayı kullanacağınız farkı ile. Bu şekilde bir çok servis bulunmaktadır. Tam olmayan sonuçları cevap döndükleri için, kapsamlı bir analiz için birden fazla servisi kullanmak daha iyidir.

Ters IP Domain tools: http://www.domaintools.com/reverse-ip/
(bedava kayıt gerektirmektedir)

MSN arama: http://search.msn.com
syntax: "ip:x.x.x.x" (without the quotes)

Webhosting bilgisi: http://whois.webhosting.info/
syntax: http://whois.webhosting.info/x.x.x.x

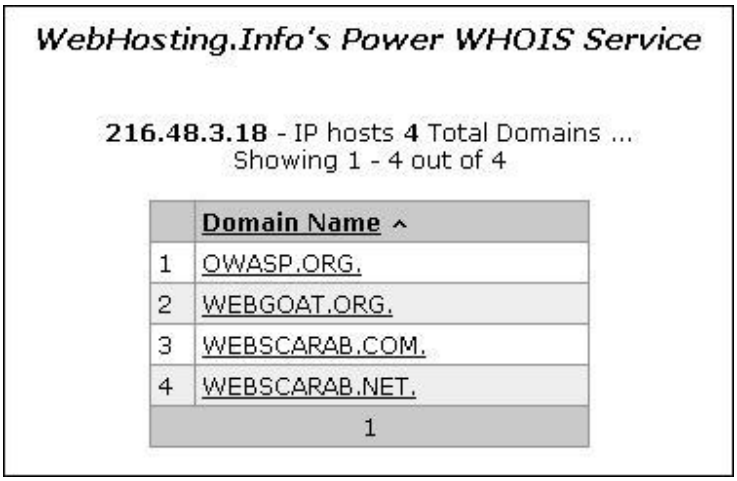
DNSstuff: http://www.dnsstuff.com/
(birçok servis vardır)

http://net-square.com/msnpawn/index.shtml
(Alan adı ve IP adresileri üzerinde birden çok sorgular, kurulum gerekmektedir)

tomDNS: http://www.tomdns.net/
(bu yazının yazılma aşamasında bazı servisler hala özeldi)

SEOlogs.com: http://www.seologs.com/ip-domains.html
(ters ip/alan adı arama)

Aşağıdaki örnek yukarıdaki ters IP servislerinden birine 216.48.3.18, www.owasp.org'un IP adresi, yapılmış sorgunun sonucunu göstermektedir. Ek olarak aynı adrese ilişkisi olan üçtü tane sembolik isim açığa çıkarılmıştır.



Googlelama

Önceki tekniklerle elde edebileceğiniz en fazla bilgiyi elde ettikten sonar, analizinizi arttırmak ve iyileştirmek için arama motorlarına güvenebilirsiniz. Bu hedefinize veya belli olmayan URLler ile ulaşılabilen uygulamalara ait başka sembolik isimleri elde etmenize yol açabilir.

Mesela, *www.owasp.org* ile alakalı önceki örnekleri göz önünde bulundurarak, Google ve diğer arama motorlarını, yeni bulunan *webgoat.org*, *webscarab.com*, *webscarab.net* alan adları ile ilgili bilgi edinmel için arama yaparak kullanabilirdiniz. Googlelama teknikleri Site link açığa çıkarma ve googlelama bölümünde açıklanmaktadır.

GRİ KUTU TESTİ VE ÖRNEK

Uygulanabilir değil. Ne kadar bilgi ile başlarsanız başlayın, metodoloji Kara Kutu testi bölümündeki ile aynıdır.

KAYNAKLAR

Makaleler

- [1] RFC 2616 – Hypertext Transfer Protocol – HTTP 1.1

Araçlar

- DNS arama araçları *nslookup*, *dig* veya benzerleri gibi.
- Port tarayıcıları (nmap gibi, http://www.insecure.org) ve açıklık tarayıcıları (Nessus gibi: http://www.nessus.org; wikto: http://www.sensepost.com/research/wikto/).
- Arama motorları (Google, ve diğerleri).
- DNS'e yönelik we b tabanlı arama servisleri: yazıya bakın.
- Nmap - http://www.insecure.org
- Nessus Açıklık Tarayıcı - http://www.nessus.org

Dökümanın Aslı : OWASP TESTING GUIDE v2

Çeviri : Bedirhan Urgun – urgunb@hotmail.com