METASPLOIT İLE WEB UYGULAMA GÜVENLİK DENETİMLERİ

Deniz Çevik, <denizcev at gmail dot com>, webguvenligi.org, 01/01/2009

Bilindiği üzere metasploit oldukça başarılı ve ücretsiz olarak dağıtılan, açık kaynak kodlu bir exploit platformudur. Değişik uygulama ve işletim sistemlerine yönelik açıkları tespit etmeyi bundan yararlanmayı sağlayan çok sayıda bileşenden oluşmaktadır. Bu bileşenler arasında WEB uygulamalarına yönelik değişik kontroller gerçekleştiren WMAP bileşeni de yer almaktadır.

scanner/http/frontpage	FrontPage Server Extensions Detection
scanner/http/frontpage_login	FrontPage Server Extensions Login Utility
scanner/http/options	HTTP Options Detection
scanner/http/version	HTTP Version Detection
scanner/http/wmap_backup_file	HTTP Backup File Scanner
scanner/http/wmap_blind_sql_query	HTTP Blind SQL Injection GET QUERY Scanner
scanner/http/wmap_brute_dirs	HTTP Directory Brute Force Scanner
scanner/http/wmap_copy_of_file	HTTP Copy File Scanner
scanner/http/wmap_dir_listing	HTTP Directory Listing Scanner
scanner/http/wmap_dir_scanner	HTTP Directory Scanner
scanner/http/wmap_file_same_name_dir	HTTP File Same Name Directory Scanner
scanner/http/wmap_files_dir	HTTP Interesting File Scanner
scanner/http/wmap_generic_email_extract	WMAP Generic Email Extractor
scanner/http/wmap_prev_dir_same_name_file	HTTP Previous Directory File Scanner
scanner/http/wmap_replace_ext	HTTP File Extension Scanner
scanner/http/wmap_sqlmap	SQLMAP SQL Injection External Module
scanner/http/wmap_ssl_vhost	HTTP SSL Certificate VHOST Detection
scanner/http/wmap_verb_auth_bypass	HTTP Verb Authentication Bypass Scanner
scanner/http/wmap_vhost_scanner	HTTP Virtual Host Brute Force Scanner
scanner/http/writable	HTTP Writable Path PUT/DELETE File Access

Bu bileşenlerden tek tek yararlanmak ortalama büyüklükte bir web sitesi için bile oldukça fazla zaman gerektirici bir uğraş olacaktır. Ancak metasploit geliştiricileri bunu düşünerek otomatik exploit çalıştırma motorunu (db_autopwn) web uygulamaları içinde düzenleyerek, metasploit, ratproxy ve sqlite3 yazılımlarını entegre şekilde çalışmasını sağlayarak, kapsamlı bir taramayı kısa sürede gerçekleştirmemize izin verecek db_wmap uygulamasını geliştirdiler. Bu yazıda db_wmap kullanarak otomatik web uygulama taramasının nasıl yapılabileceği gösterilmeye çalışılmıştır.

KURULUM

Metaspolit ile otomatik web uygulama taramasını gerçekleştirmek için sistem üzerinde Metasploit, sqlite3 ve raptproxy yazılımlarının kurulu ve çalışır durumda olması gerekmektedir. Yazıda bu ürünlerin nasıl kurulacağından çok dikkat edilmesi gereken noktalara değinilmiştir.

1- SQLITE3 yazılımının güncel olması önem taşıyor. Sistemime otomatik kurulu gelen Sqlite 3.3.3 sürümünde oldukça fazla sorunla karşılaştım. Ancak güncel sürümü 3.6.7'ye yükseltince her hangi bir problem kalmadı.

2- Ratproxy kaynak kodlarının Metasploit ile birlikte çalışabilmesi için derlenmeden önce Metasploit tarafından hazırlanan yamanın uygulanması gerekmektedir. Ratproxy yazılımı aşağıdaki linkten indirilebilir.

http://ratproxy.googlecode.com/files/ratproxy-1.51.tar.gz

Gerekli yama dosyaları ise Metasploit'in kurulu olduğu dizin altındaki /external/ratproxy/ alt dizini içinde yer almaktadır. Yamayı uygulamak için aşağıdaki komut dizin içinde iken çalıştırılabilir.

```
patch -d RAT_PROXY_Dizini < ratproxy_wmap.diff
```

İlgili yama ratproxy yazılımına uygulandıktan sonra make komutu ile kaynak kodları derleyebilirsiniz. Eğer tüm işlemleri doğru yaptı iseniz ./ratproxy –h komutunu çalıştırdığınız zaman ekran çıktısının en altında aşağıdaki gibi bir ifade yer almalıdır.

```
4) Wmap           : -v <outdir> -b <wmap db>
```

WMAP KULLANIMI

1- Metasploit konsoluna bağlanmadan önce "svn update" komutu ile güncelleme işlemini çalıştırın.
2- ./msfconsole komutu ile metasploit konsoluna bağlanın.
3- Her türlü işlemin yazılacağı veritabanını oluşturun. Bunun için aşağıdaki komutlar kullanılabilir.

```
msf > load db_sqlite3
[*] Successfully loaded plugin: db_sqlite3
msf > db_create /root/.msf3/webapptest.db
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /root/.msf3/webapptest.db
msf > db_connect /root/.msf3/webapptest.db
[*] Successfully connected to the database
[*] File: /root/.msf3/webapptest.db
```

4- WMAP modulünü çalıştırın.

```
msf > load db_wmap
[*] =[ WMAP v0.3 - ET LoWNOISE
[*] Successfully loaded plugin: db_wmap
```

5- Farklı bir konsol penceresinden ise Metasploit ile birlikte çalışması için yamadığımız ratproxy uygulamasını aşağıdaki gibi çalıştırın.

```
# ./ratproxy -v /root/.msf3/ -b webapptest.db -a -r
ratproxy version 1.51-beta by <lcamtuf@google.com>

[!] WARNING: Running with no 'friendly' domains specified. Many cross-
domain checks will not work. Please consult the documentation for advice.

[*] Proxy configured successfully. Have fun, and please do not be evil.
[+] Accepting connections on port 8080/tcp (any source)...
```

6- Daha sonra Internet tarayıcınız üzerindeki proxy ayarını Ratproxy olacak şekilde düzenleyerek denetlemesini gerçekleştireceğini web sitesini ziyaret edin. Tüm linkleri ziyaret ettikten sonra Metasploit ekranın WMAP ile taramaları başlatabiliriz.

7- Metasploit üzerinde tarama yapılacak hedef sunucuyu seçin. Wmap_targets komutunun –r parametresi ratproxy tarafından belirlenen sistemleri yükleyecektir. Tarama yapmak istediğimiz hedef sunucuyu ise –s parametresini kullanarak seçebiliriz

```
msf > wmap_targets -r
[*] Added. 192.168.1.1 80 0
[*] Added. 192.168.1.2 80 0
msf > wmap_targets -s 1
msf > wmap_targets -p
[*]    Id. Host                                Port    SSL
[*] => 1. 192.168.1.1                          80
[*]    2. 192.168.1.2                          80
[*] Done.
```

8- wmap_website komutu ile hedef olarak seçtiğimiz sitedeki uygulamaları görebiliriz. Testler bu komut çıktısında yer alan uygulamalara ve dizinlere doğru gerçekleştirilecektir.

```
[*] Website structure
[*] 192.168.1.1:80 SSL:0
ROOT_TREE
|       images
|       +------p_main.jpg
+------default.aspx
|       testfireappl
|       |       storefront
|       |       |       testfire_files
|       |       |       +------help_graphic.gif
[*] Done.
```

9- Bu sunucuya doğru hangi wmap bileşenlerinin kullanılacağını görmek için ise wmap_run komutu –t parametresi ile birlikte kullanılabilir.

```
msf > wmap_run -t
[*] Loaded auxiliary/scanner/http/frontpage ...
[*] Loaded auxiliary/scanner/http/options ...
[*] Loaded auxiliary/scanner/http/wmap_vhost_scanner ...
[*] Loaded auxiliary/scanner/http/wmap_verb_auth_bypass ...
[*] Loaded auxiliary/scanner/http/wmap_ssl_vhost ...
[*] Loaded auxiliary/scanner/http/frontpage_login ...
[*] Loaded auxiliary/scanner/http/version ...
[*] Loaded auxiliary/scanner/http/wmap_files_dir ...
[*] Loaded auxiliary/scanner/http/wmap_brute_dirs ...
[*] Loaded auxiliary/scanner/http/writable ...
[*] Loaded auxiliary/scanner/http/wmap_dir_listing ...
[*] Loaded auxiliary/scanner/http/wmap_copy_of_file ...
[*] Loaded auxiliary/scanner/http/wmap_replace_ext ...
[*] Loaded auxiliary/scanner/http/wmap_prev_dir_same_name_file ...
[*] Loaded auxiliary/scanner/http/wmap_file_same_name_dir ...
[*] Loaded auxiliary/scanner/http/wmap_dir_scanner ...
[*] Loaded auxiliary/scanner/http/wmap_backup_file ...
[*] Loaded auxiliary/scanner/http/wmap_blind_sql_query ...
[*] Analysis completed in 2.26853394508362 seconds.
[*] Done.
msf >
```

10- Taramayı wmap_run komutunu –e parametresi ile çalıştırarak başlatabiliriz.

```
msf > wmap_run -e
[*] Using ALL wmap enabled modules.
```

```
[*] Launching auxiliary/scanner/http/frontpage WMAP_SERVER against
192.168.1.1:80
[*] http://192.168.1.1:80 is running Microsoft-IIS/6.0
[*] FrontPage not found on http://192.168.1.1:80 [404 Not Found]
[*] Launching auxiliary/scanner/http/options WMAP_SERVER against
192.168.1.1:80
[*] 192.168.1.1 allows OPTIONS, TRACE, GET, HEAD, POST methods
[*] Launching auxiliary/scanner/http/wmap_vhost_scanner WMAP_SERVER
against 192.168.1.1:80
[*]  >> Exception during launch from
auxiliary/scanner/http/wmap_vhost_scanner: The following options failed
to validate: DOMAIN.
[*] Launching auxiliary/scanner/http/wmap_verb_auth_bypass WMAP_SERVER
against 192.168.1.1:80
[*] 192.168.1.1 No requires authentication. / 200
[*] Launching auxiliary/scanner/http/wmap_ssl_vhost WMAP_SERVER against
192.168.1.1:80
[*] Error: 192.168.1.1 unknown protocol
[*] Launching auxiliary/scanner/http/frontpage_login WMAP_SERVER against
192.168.1.1:80
[*] http://192.168.1.1:80/ may not support FrontPage Server Extensions
[*] Launching auxiliary/scanner/http/version WMAP_SERVER against
192.168.1.1:80
[*] 192.168.1.1 is running Microsoft-IIS/6.0 ( Powered by ASP.NET )
[*] Launching auxiliary/scanner/http/wmap_files_dir WMAP_DIR / against
192.168.1.1:80...
[*] NOT Found http://192.168.1.1:80/0.aspx
[*] NOT Found http://192.168.1.1:80/00.aspx
[*] NOT Found http://192.168.1.1:80/01.aspx
[*] NOT Found http://192.168.1.1:80/02.aspx
[*] NOT Found http://192.168.1.1:80/03.aspx

------------ KES -----------------
```

11- Tarama sonuçlarını görmek için ise wmap_reports komutu kullanılabilir.

```
msf > wmap_reports -p
[*] Id. Created                         Target (host,port,ssl)
1.  Tue Dec 30 18:39:43 +0200 2008      192.168.1.1,80,0
[*] Done.
msf > wmap_reports -s 1
WMAP REPORT: 192.168.1.1,80,0 Metasploit WMAP Report [Tue Dec 30 18:39:43
+0200 2008]
WEB_SERVER OPTIONS: OPTIONS, TRACE, GET, HEAD, POST  [Tue Dec 30 18:39:45
+0200 2008]
WEB_SERVER TYPE: Microsoft-IIS/6.0 ( Powered by ASP.NET )  [Tue Dec 30
18:39:48 +0200 2008]
FILE NAME: /bug.aspx File /bug.aspx found. [Tue Dec 30 18:41:14 +0200 2008]
```

Görüldüğü üzere Metasploit kullanarak web uygulama güvenlik denetimlerinde yapılan kontrollerin bir bölümünü gerçekleştirmek rahatlıkla mümkün olmaktadır. Ek olarak uygulamanın açık kaynaklı olması nedeni ile kendi wmap bileşenlerinizi yazarak bu kontrollerin daha da geniş bir alanı kapsamasına yardımcı olabilirsiniz.