

Anti Scanner

written by Mert SARICA | 3 February 2014

2011 yılının Şubat ayında, web sitemin, Acunetix, Netsparker ve Appscan web uygulaması güvenliği zafiyet tarama araçları ile sıkça taranmasından dolayı bu araçlar üzerinde ufak bir araştırma yapıp Script Kiddie Bezdirme Mekanizması adında bir yazı yazmıştım. Geçtiğimiz aylarda sitemin kayıtlarını incelerken yine çok sayıda Netsparker ile tarama kaydına rastladım. Ufak bir araştırma ve karşılaştırma sonucunda, geçtiğimiz 3 sene içinde sitemi taramak için kullanılan araçların başında yine Netsparker'ın (community edition) olduğunu, ikinci olarak ise Acunetix'in (ticari sürüm) olduğunu gördüm. Netsparker'ın hem ücretsiz olması hem de ticari sürümüne göre kısıtları olmasına rağmen, rakiplerine kıyasla daha tutarlı sonuçlar üretebilen etkili bir araç olması, güvenlik uzmanlarının yanı sıra niyeti bozuk arkadaşlar tarafından da tercih edilmesine neden olmaktadır. 3 sene önceye göre sitesi daha da sık taranan ve nezaketen de olsa tarayanlar tarafından ne idari ne de teknik zafiyet analiz raporu paylaşılmayan biri olarak (:)) tarayanların işini 3 sene önceye göre biraz daha zorlaştırmaya, yöntemi ve ilgili kodları sizlerle paylaşmaya karar verdim.

Sitem daha çok Netsparker ile tarandığı için ilk olarak Netsparker odaklı basit bir çözüm üretmeye karar versem de, özelleştirilebilen daha esnek bir çözümün daha fazla zafiyet tarayıcısını ve zafiyet arayan botları engellemede kullanılabileceğini düşünerek farklı çözümler üzerinde düşünmeye başladım.

İşe ilk olarak WordPress'in trafik kayıtlarını incelemekle başladım. Çoğu zafiyet tarayıcısı tarama esnasında, USER-AGENT alanları da dahil olmak üzere sunucuya gönderilen verilere imzalarını (Acunetix, Netsparker vs.) atarlar. Özellikle Netsparker gibi ücretsiz olarak dağıtılan araçlarda bu imzaların arayüz üzerinden değiştirilmesi çoğu zaman mümkün olmamaktadır dolayısıyla bu imzaya yönelik üretilebilecek basit bir çözüm, tara ve geçen öteye gidemeyen niyeti bozuk kişileri ve/veya botları bezdirmek için yeterli olacaktır. Örneğin aşağıdaki iki ekran görüntüsüne bakacak olursanız burada Netsparker'ın USER-AGENT alanında imzasına yer verdiğini görebilirsiniz.

17 December, 2013	12:51:08	85.102.160.100	English		http://\'--/style/scriptsriptnetsparker(0x000284)/script	[Page]: Home
17 December, 2013	12:51:05	85.102.160.100	English		http://\'+NSFTW+\'	[Page]: Home
17 December, 2013	12:50:46	85.102.160.100	English		http://\'+NSFTW+\'	[Page]: Home
17 December, 2013	12:50:45	85.102.160.100	English		http://ns.netsparker056650=vuln	[Page]: Home
17 December, 2013	12:50:40	85.102.160.100	English		http://ns.netsparker056650=vuln	[Page]: Home
17 December, 2013	12:50:38	85.102.160.100	English		http://netsparker.com/n	[Page]: Home
17 December, 2013	12:50:34	85.102.160.100	English		http://\'+NSFTW+\'	[Page]: Home
17 December, 2013	12:50:32	85.102.160.100	English		http://\'+NSFTW+\'	[Page]: Home
17 December, 2013	12:50:30	85.102.160.100	English		http://\'--/style/scriptsriptnetsparker(0x00028F)/script	[Page]: Home
17 December, 2013	12:50:30	85.102.160.100	English		http://netsparker.com/n	[Page]: Home
17 December, 2013	12:50:25	85.102.160.100	English		http://\'--/style/scriptsriptnetsparker(0x000284)/script	[Page]: Home
17 December, 2013	12:50:15	85.102.160.100	English		http://ns.netsparker056650=vuln	[Page]: Home
17 December, 2013	12:50:07	85.102.160.100	English		http://ns.netsparker056650=vuln	[Page]: Home

Report for 85.102.160.100

[Ban IP address](#)

Records in database: 952

Latest hit: 17 December, 2013 12:52:27

First hit: 17 December, 2013 12:44:41

User agent(s): Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/7.0)
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36

URLs Requested

Date	Time	OS	Browser	Agent	Referrer	URL Requested
17 December, 2013	12:52:27	 Windows XP	 Internet Explorer 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)		/wp-content/doxygen.conf
17 December, 2013	12:52:18	 Windows XP	 Internet Explorer 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)		/wp-content/httpd-vhosts.conf
17 December, 2013	12:52:17	 Windows XP	 Internet Explorer 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)		/wp-content/local.conf
17 December, 2013	12:52:16	 Windows XP	 Internet Explorer 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)		/wp-content/httpd.conf
17 December, 2013	12:52:16	 Windows XP	 Internet Explorer 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)	Euroforensics 2013 Adli Bilimler, Siber Güvenlik ve Gözetim Teknolojileri Konferansı ve Sergisi	
17 December, 2013	12:52:14	 Windows XP	 Internet Explorer 6	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Netsparker)		/wp-content/test.conf

Bezdirme yöntemi olarak tarayıcı tarafından web sunucusuna gönderilen her istek (request) için, rastgele değerlerden oluşan bir form ve az sayıda sahte zafiyet (dizin bilgisi ifşası, veritabanı bilgisi ifşası) oluşturan kısa ve öz bir PHP uygulaması hazırlamaya karar verdim. Buradaki amacım, rastgele değerlerden oluşan bir form oluşturan bu PHP uygulaması sayesinde tarayıcı, her gönderdiği yeni istekte, yeni bir form ve bunun bağlı olduğu yeni bir sayfa ile karşılaştığını zannederek her sayfayı, bu sayfada bulunan formu ve ilgili alanları, test edilecek sayfalar kuyruğuna alarak kısır döngüye girmesini ve/veya sistem üzerinde performans sorununa yol açmasını sağlamaktı.

```

1 <?php
2 function randString($length, $charset='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789')
3 {
4     $sstr = '';
5     $count = strlen($charset);
6     while ($length-->0) {
7         $sstr .= $charset[mt_rand(0, $count-1)];
8     }
9     return $sstr;
10 }
11 ?>
12 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
13 <html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
14 <head profile="http://gmpg.org/xfn/11">
15 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
16 <title>Hack 4 Career - http://www.mertsarica.com</title>
17 <link rel="stylesheet" href="http://www.<?php echo randString(50);?>.css" type="text/css" media="screen" />
18 <link rel="stylesheet" href="http://www.<?php echo randString(50);?>.css" type="text/css" media="screen" />
19 <link rel="alternate" type="application/rss+xml" title="Hack 4 Career RSS Feed" href="http://www.<?php echo randString(50);?>.com/feed/" />
20 <link rel="alternate" type="application/atom+xml" title="Hack 4 Career Atom Feed" href="http://www.<?php echo randString(50);?>.com/feed/atom/" />
21 <link rel="pingback" href="http://www.<?php echo randString(50);?>.com/xmlrpc.php" />
22
23 <form action="<?php echo randString(50);?>.php" method="post">
24 <p><?php echo randString(50);?>: <input type="text" name="<?php echo randString(50);?>" /></p>
25 <p><?php echo randString(50);?>: <input type="text" name="<?php echo randString(50);?>" /></p>
26 <p><input type="submit" /></p>
27 </form>
28
29 function antiscanner($antiscanner)<br \>
30 {<br \>
31     return $antiscanner;<br \>
32 }<br \>
33
34 "/usr/local/<?php echo randString(50);?>"<br \>
35
36 "c:/<?php echo randString(50);?>"<br \>
37
38 define( 'DB_NAME', 'database' );<br \>
39 define( 'DB_USER', 'www.mertsarica.com' );<br \>
40 define( 'DB_PASSWORD', 'antiscanner' );<br \>
41 define( 'DB_HOST', 'localhost' );<br \>
42 define( 'DB_CHARSET', 'utf8' );<br \>
43
44 <?php echo randString(50);?><?php echo randString(50);?>.com<br \>
45 </html>

```

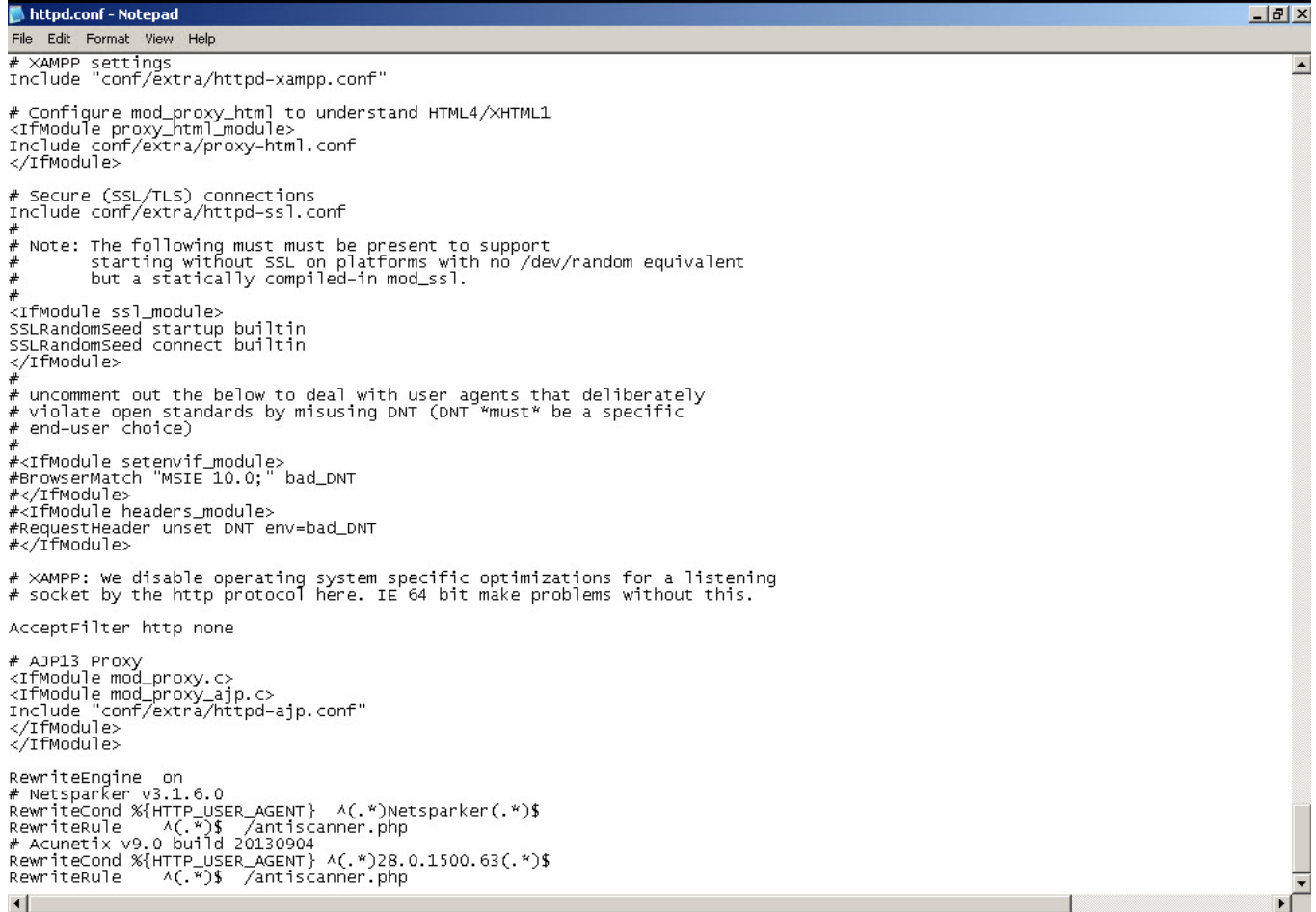
Tabii tarayıcıyı kısır döngüye sokabilmek için web sunucusu üzerinde PHP uygulaması tarafından oluşturulan her sahte form sayfasının çağrıldığında, web sunucusunun tarayıcıya geçerli (200 OK) sağlamam gerekiyordu. Bunun için sunucu üzerinde olası binlerce sayfa oluşturamayacağım için Apache'nin mod_rewrite modülünden faydalanmaya karar verdim.

mod_rewrite gelen URL isteklerini düzenli ifade kurallarına dayanarak devingen olarak dönüştürmek için bir yöntem sağlar. Böylece keyfi URL'leri kendi URL yapınızla istediğiniz şekilde eşleştirmeniz mümkün olur.

Gerçekten esnek ve güçlü bir URL kurgulama mekanizması oluşturmak için sınırsız sayıda kural ve her kural için de sınırsız sayıda koşul destekler. URL değişiklikleri çeşitli sınamalara bağlı olabilir: sunucu değişkenleri, HTTP başlıkları, ortam değişkenleri, zaman damgaları, çeşitli biçimlerde harici veritabanı sorguları.

Tabii yanıtlanması gereken ufak bir soru daha vardı o da mod_rewrite ile tarayıcıyı kısır döngüye sokarken gerçek kullanıcının bundan nasıl etkilenmemesini sağlayabilirdim ? Bunun için yazının girişinde bahsettiğim ve tarayıcıların imzalarını kullandıkları USER-AGENT alanına yönelik bir mod_rewrite kuralı yazmaya karar verdim. Tabii Acunetix'in ticari sürümündeki (v9.0 build 20130904) varsayılan USER-AGENT imzası, Netsparker'ın (v3.1.6.0)

aksine kendi adı yerine Chrome'un USER-AGENT değerini kullanıyordu. Chrome internet tarayıcısı otomatik güncellemeye sahip olduğu ve Acunetix'in USER-AGENT alanında varsayılan olarak kullandığı bu değer, eski bir sürüme ait olduğu için dert etmeden, gönül rahatlığıyla Acunetix için de bir kural yazabileceğime karar verdim.



```
# httpd.conf - Notepad
File Edit Format View Help

# XAMPP settings
Include "conf/extra/httpd-xampp.conf"

# Configure mod_proxy_html to understand HTML4/XML
<IfModule proxy_html_module>
Include conf/extra/proxy-html.conf
</IfModule>

# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
#
# Note: The following must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
#
# Uncomment out the below to deal with user agents that deliberately
# violate open standards by misusing DNT (DNT "must" be a specific
# end-user choice)
#
<IfModule setenvif_module>
#BrowserMatch "MSIE 10.0;" bad_DNT
</IfModule>
<IfModule headers_module>
#RequestHeader unset DNT env=bad_DNT
</IfModule>

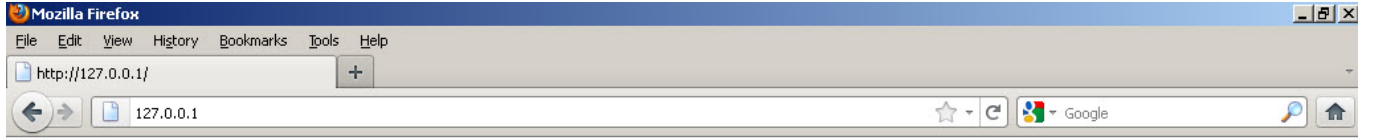
# XAMPP: We disable operating system specific optimizations for a listening
# socket by the http protocol here. IE 64 bit make problems without this.
AcceptFilter http none

# AJP13 Proxy
<IfModule mod_proxy.c>
<IfModule mod_proxy_ajp.c>
Include "conf/extra/httpd-ajp.conf"
</IfModule>
</IfModule>

RewriteEngine on
# Netsparker v3.1.6.0
RewriteCond %{HTTP_USER_AGENT} ^(.*)Netsparker(.*)$
RewriteRule ^(.*)$ /antiscanner.php
# Acunetix v9.0 build 20130904
RewriteCond %{HTTP_USER_AGENT} ^(.*)28.0.1500.63(.*)$
RewriteRule ^(.*)$ /antiscanner.php
```

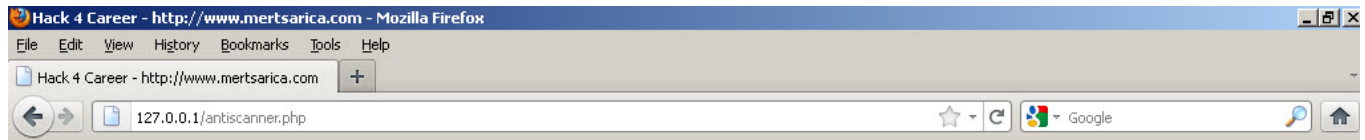
Yukardaki mod_rewrite kuralı ile USER-AGENT alanı, Netsparker veya Acunetix'in kullandığı değere eşit ise, istekleri otomatik olarak hazırladığım PHP uygulamasına (antiscanner.php) yönlendirdim.

Öncelikle normal kullanıcıların bu PHP uygulamasından etkilenmediğini teyit etmek için sayfayı internet tarayıcısının varsayılan USER-AGENT'ı ile çağırdığımda sayfanın normal halini görüntüleyebildim.



Hack4Career
<http://www.mertsarica.com>

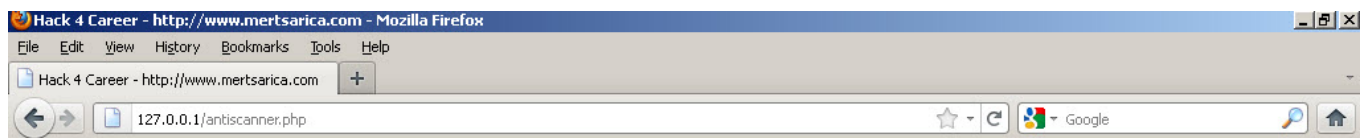
Ardından Firefox'un User Agent Switcher eklentisi ile USER-AGENT'ımı Netsparker olarak değiştirdikten sonra sayfanın her istekte farkı yanıt (form ve formun bulunduğu adres) döndüğünü doğruladım.



KbZ6Wqpk7ZZRKfM3gloW4B0B26DS2uu9tHzYLcmA8DkV8HNL2w:

aQlm7hsaOyqAdjqWtkR0ZGR7wGEORnotpi04pTAQNnHlx1u3R5:

```
function antiscanner($antiscanner)
{
return $antiscanner;
}
"/usr/local/r3DTNPv2DijGQUesnWCWsbIG1BSigU4GDa6zhqGA5bjACtIT3"
"c:/wHc216fs5oI6O5Gbjkg3464kNGXuIMVoD0ktFdV13Z9ONol5"
define( 'DB_NAME', 'database' );
define( 'DB_USER', 'www.mertsarica.com' );
define( 'DB_PASSWORD', 'antiscanner' );
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8' );
J07NrgQvoBhl0BLLMscmd1lqxvf8idM8LxkOtlGLouszHN2K YU@mabYEJPzNgBdv3yFnb3YJwVaJp8KSIgP0i5NDmRY3i8qd79TM.com
```



2N3D4A5bQqHPa2Y CZmZyuWlRfed0AnU13opboBNruC2qOr8R1:

RA40AUhw8SFaXwpbXpis7P3k2XWAGRuFTNbS7NychQVFmeNHC:

```
function antiscanner($antiscanner)
{
return $antiscanner;
}
"/usr/local/aF79V6teJWnFYZRvR9G2uw7GL2G5OXIaiRGOqr9aDTbIT1DkN"
"c:/ha4CtMkL4KoaOkTbUBEr6iqvO8avJslhc6mVlyneQ7FCyetw2V"
define( 'DB_NAME', 'database' );
define( 'DB_USER', 'www.mertsarica.com' );
define( 'DB_PASSWORD', 'antiscanner' );
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8' );
ZEw33U17Rr3qIpBMXa3KM6vd0AxWjcdUQp7Z3j9S8iOCgRKmqn@A94ZFLjWkOpHMmbepM06siEbMffCi0bH5siTrFh2000eercb7.com
```

Sıra Netsparker ve Acunetix ile test yapmaya geldiğinde, Netsparker Community Edition sürümünün, başladığı taramayı 4 saat sonunda hala bitiremediğini ve

artan bellek kullanımı nedeniyle işletim sistemi üzerinde bellek sorununa yol açtığını gördüm.

The screenshot displays the Netsparker 3.1.6.0 interface during a scan. The 'Issues' panel on the right lists the following items:

- Auto Complete Enabled
- Forbidden Resource
- E-mail Address Disclosure
- [Possible] Internal Path Disclosure (Windows)
- [Possible] Internal Path Disclosure (*nix)

A yellow warning box is visible, stating: "Windows - Virtual Memory Minimum Too Low. Your system is low on virtual memory. Windows is increasing the size of your virtual memory paging file. During this process, memory requests for some applications may be denied. For more information, see Help."

The 'Windows Task Manager' window is open, showing the following processes:

Image Name	User Name	CPU	Mem Usage
Netsparker.exe	Administrator	81	216,492 K
firefox.exe	Administrator	08	74,184 K
NetsparkerHelper.exe	Administrator	08	52,600 K
chrome.exe	Administrator	00	36,892 K
rubyw.exe	SYSTEM	00	32,908 K
chrome.exe	Administrator	00	28,740 K
rubyw.exe	SYSTEM	00	21,828 K
chrome.exe	Administrator	00	18,784 K
javaw.exe	Administrator	00	12,788 K
httpd.exe	Administrator	05	12,128 K
explorer.exe	Administrator	00	11,616 K
svchost.exe	SYSTEM	00	11,048 K
mcshield.exe	SYSTEM	00	9,276 K
chrome.exe	Administrator	00	8,676 K
httpd.exe	Administrator	00	7,428 K
vmtoolsd.exe	Administrator	02	6,580 K
vmtoolsd.exe	SYSTEM	00	4,392 K
xampp-control.exe	Administrator	00	2,968 K
winlogon.exe	SYSTEM	00	2,864 K

The 'Issues' panel also shows the following information:

- Auto Complete Enabled
- Forbidden Resource
- E-mail Address Disclosure
- [Possible] Internal Path Disclosure (*nix)
- [Possible] Internal Path Disclosure (Windows)

The 'Settings' panel on the right shows the 'Custom Cookies' section.

Acunetix ile yapmış olduğum taramada ise bellek sorunu ile karşılaşmamış olsam da taramanın 2 saat sonunda hala bitemediğini gördüm.

The screenshot displays the Acunetix Web Vulnerability Scanner (Consultant Edition) interface. The main window shows a scan in progress for the target URL `http://127.0.0.1:80/`. The scan results are categorized into Web Alerts (153), Knowledge Base, and Site Structure. The Web Alerts section lists several issues, including Slow HTTP Denial of Service Attack (1), HTML form without CSRF protection (150), Clickjacking: X-Frame-Options header missing (1), and TRACE method is enabled (1). The Site Structure section shows a list of URLs, all of which are marked as OK. The Status section indicates that the scan is in progress. The Target information section shows the target URL as `http://127.0.0.1:80/`. The Alerts summary section shows 153 alerts. The Response time history section shows a graph of response times over time. The Statistics section shows 20902 requests. The Progress section shows a progress bar at 25.27%.

Tools Explorer

- Web Vulnerability Scanner
- Web Scanner
- Tools
 - Site Crawler
 - Target Finder
 - Subdomain Scanner
 - Blind SQL Injector
 - HTTP Editor
 - HTTP Sniffer
 - HTTP Fuzzer
 - Authentication Tester
 - Compare Results
- Web Services
 - Web Services Scanner
 - Web Services Editor
- Configuration
 - Application Settings
 - Scan Settings
 - Scanning Profiles
- General
 - Program Updates
 - Version Information
 - Licensing
 - Support Center
 - Purchase
 - User Manual
 - AcuSensor

Scan Results

Scan Thread 1 (`http://127.0.0.1:80/`)

Web Alerts (153)

- Slow HTTP Denial of Service Attack (1)
- HTML form without CSRF protection (150)
- Clickjacking: X-Frame-Options header missing (1)
- TRACE method is enabled (1)

Knowledge Base

Site Structure

Status

Responsive: true
Web server banner: Apache
Operating system: Unknown
Web server: Apache
Technologies: Apache

Target information `http://127.0.0.1:80/`

Alerts summary 153 alerts

Scan time: 26 minutes, 41 seconds
Number of requests: 20902
Average response time: 1,51 milliseconds
Scan iteration: 1

Response time history

Statistics 20902 requests

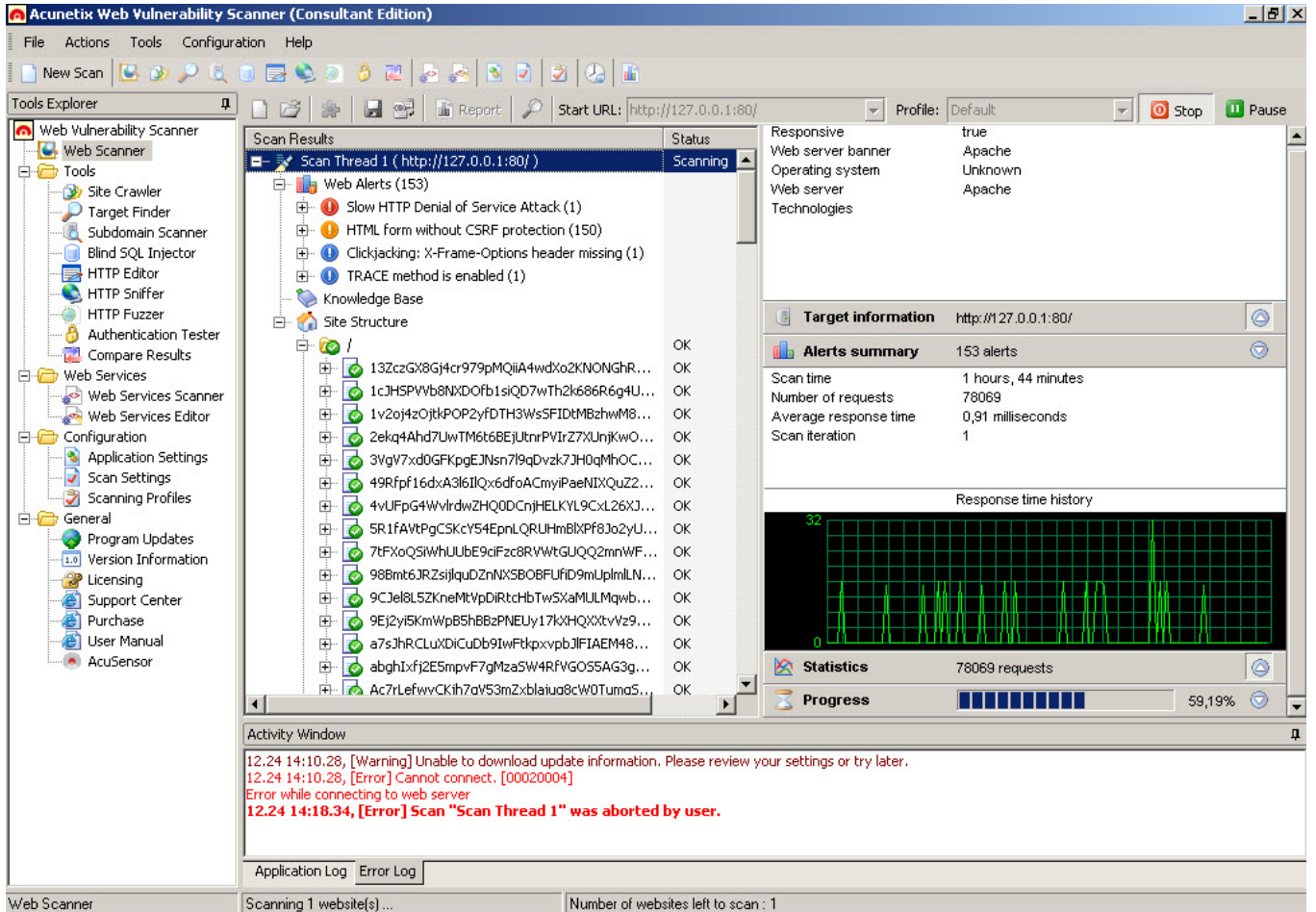
Progress 25,27%

Activity Window

12.24 14:10.28, [Warning] Unable to download update information. Please review your settings or try later.
12.24 14:10.28, [Error] Cannot connect. [00020004]
Error while connecting to web server
12.24 14:18.34, [Error] Scan "Scan Thread 1" was aborted by user.

Application Log **Error Log**

Web Scanner Scanning 1 website(s) ... Number of websites left to scan : 1



Kıssadan hisse, mod_rewrite ve ufak bir PHP uygulaması ile script kiddieler'in taramalarını yavaşlatacak bir yöntem geliştirmiş oldum. Evet baktığınız zaman bu yöntemin atlatılması çok zor değil ancak ilave kontroller uygulayarak kedi fare oyunundaki yerinizi alabilirsiniz :)

Örnek PHP uygulamasına ve mod_rewrite kuralı içeren httpd.conf dosyasını buradan indirebilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.