Detecting Web Application Firewalls

pentestlab.blog/category/web-application/page/4

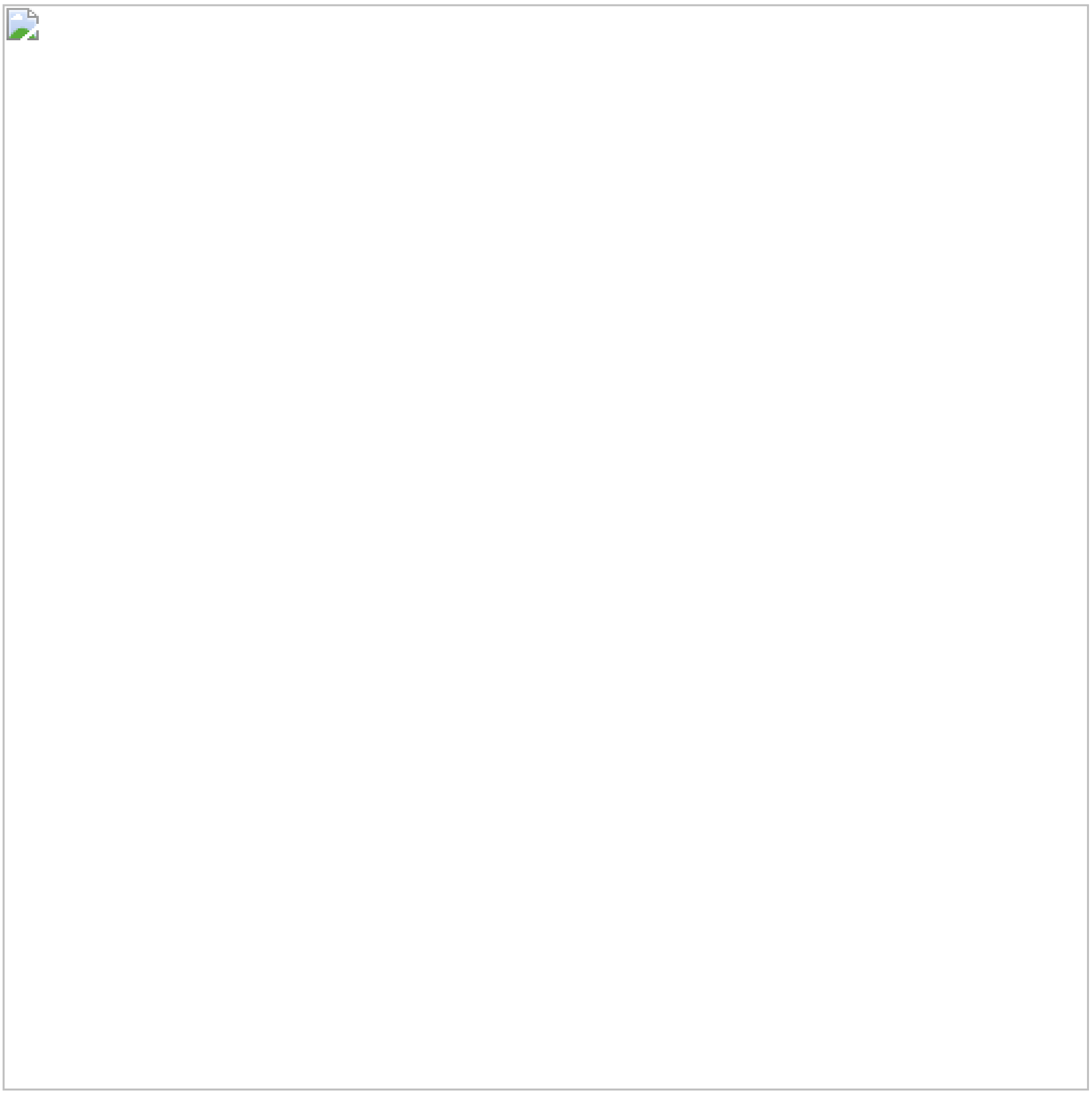
January 13, 2013

WAF – Network Topology

Web application firewalls play an important role in the security of websites as they can mitigate risks and they can offer protection against a large-scale of vulnerabilities.That is the reason that many companies in nowadays are implementing a web application firewall solution in their existing infrastructure.Of course an implementation of a WAF on its own cannot resolve the security problems that a web application might have and proper modifications must be made in order many of the attacks to be able to identified and blocked.

Penetration testers must be aware before they start the web application engagement if there is a WAF in place as the results of their attacks can be affected.So if the penetration test is a white-box then this question should be asked in the initial meetings with the client.If it is a black-box then the penetration tester should try to investigate on his own whether or not there is a web application firewall in place.In this article we will try to examine the methods and the tools that will allow us to detect a WAF.

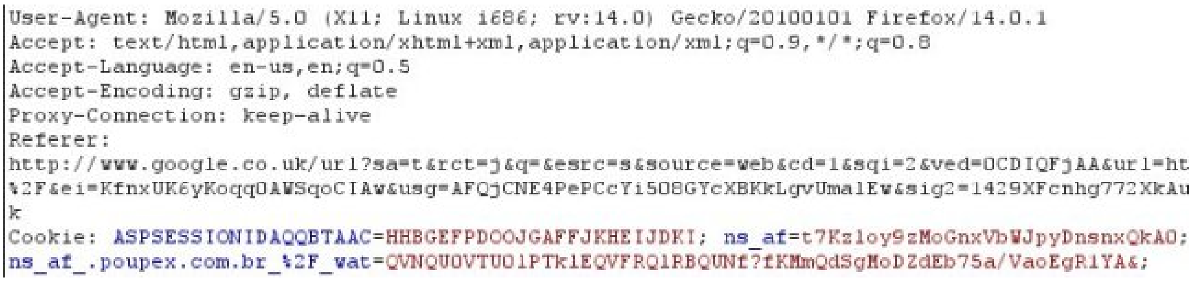
Before we start it is always good to know where a WAF is usually used on a network.Most of the times a web application firewall is between a web server and a client like the one that we can see in the next image.However there are web application firewalls that can be installed directly into the web servers.



WAF – Network Topology

Manual Discovery

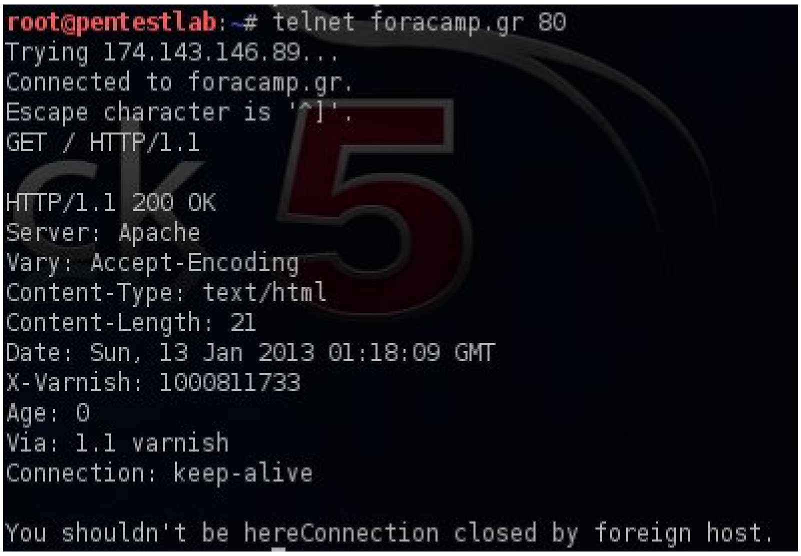
The existence of a web application firewall can be identified with a variety of ways.A good indication is by checking the cookies because some web application firewalls add their own cookie in the communication between the client and the web server.For example in the next image we can see an HTTP request where a cookie has been added by the WAF.Specifically the **ns_af** unveils that the web application firewall is a Citrix Netscaler.



```
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:14.0) Gecko/20100101 Firefox/14.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
Referer:
http://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&sqi=2&ved=0CDIQFjAA&url=ht
%2F&ei=KfnxUK6yKoqq0AWSqoCIAw&usg=AFQjCNE4PePCcYi508GYcXBKkLgvUmalEw&sig2=1429XFcnhg772XkAu
k
Cookie: ASPSESSIONIDAQQBTAAC=HHBGEFPDOOJGAFFJKHEIJDKI; ns_af=t7Kzloy9zMoGnxVbWJpyDnsnxQkA0;
ns_af_.poupex.com.br_%2F_wat=QVNQU0VTU01PTk1EQVFRQ1RBQUNf?fKMmQdSgMoDZdEb75a/VaoEgR1YA&;
```

WAF Discovery Via Cookies

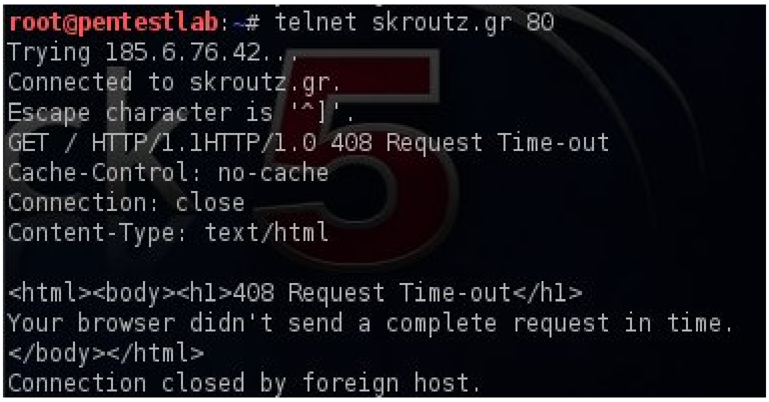
Another method is through the HTTP headers as many WAF products allow the header to be rewritten and they can also make the web server to produce different HTTP responses from the common ones.For example as we can see and from the image below the web server respond to our request with a message *You shouldn't be here* and unveiled that is Varnish.



```
root@pentestlab:~# telnet foracamp.gr 80
Trying 174.143.146.89...
Connected to foracamp.gr.
Escape character is '^]'.
GET / HTTP/1.1

HTTP/1.1 200 OK
Server: Apache
Vary: Accept-Encoding
Content-Type: text/html
Content-Length: 21
Date: Sun, 13 Jan 2013 01:18:09 GMT
X-Varnish: 1000811733
Age: 0
Via: 1.1 varnish
Connection: keep-alive

You shouldn't be hereConnection closed by foreign host.
```

Indication of WAF via HTTP response

Additionally a web application firewall presence can be identified in cases where you are trying to send a request and the session is expiring very quickly like the example in the next image.

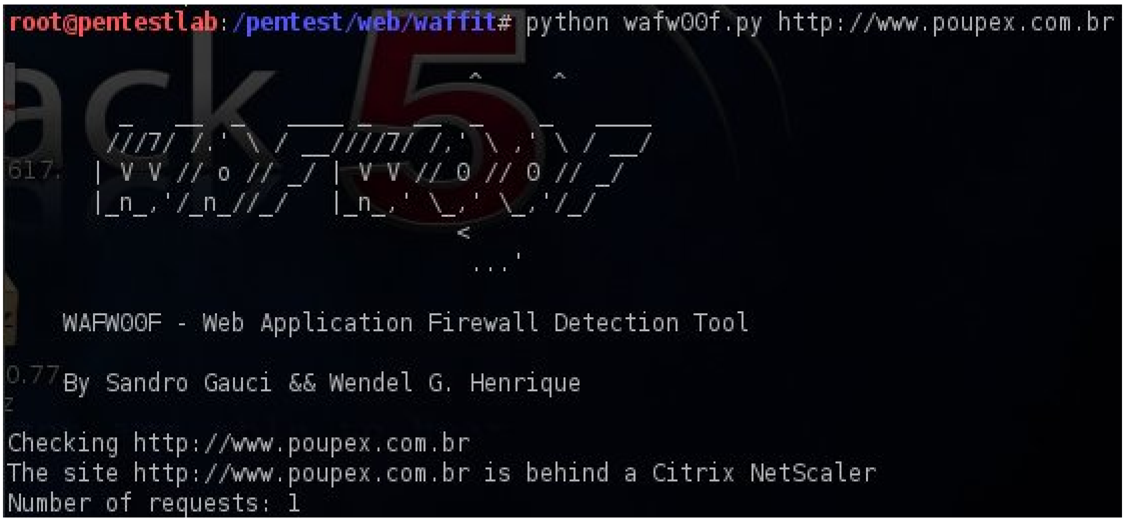


```
root@pentestlab:~# telnet skroutz.gr 80
Trying 185.6.76.42...
Connected to skroutz.gr.
Escape character is '^]'.
GET / HTTP/1.1HTTP/1.0 408 Request Time-out
Cache-Control: no-cache
Connection: close
Content-Type: text/html

<html><body><h1>408 Request Time-out</h1>
Your browser didn't send a complete request in time.
</body></html>
Connection closed by foreign host.
```

WAF – Session Expired

Automated Discovery

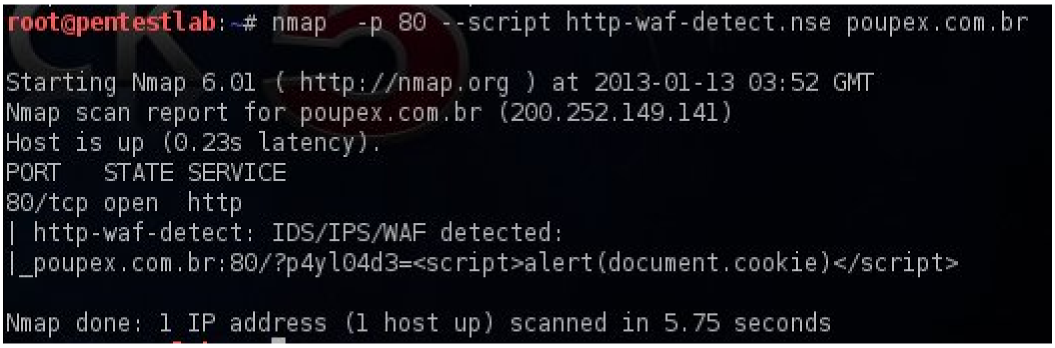
The most well-known tool that can detect and fingerprint web application firewalls is the WAFW00F.The usage of this tool is very simple and can discover a variety of WAF products.The next image is showing the successful detection of a Citrix Netscaler firewall that protects the website.



```
root@pentestlab:/pentest/web/waffit# python wafw00f.py http://www.poupex.com.br

                                 ^     ^
        _   __  _   ____ _   __  _    _   ____
       ///7/ /.' \ / __////7/ /,' \ ,' \ / __/
      | V V // o // _/ | V V // 0 // 0 // _/
      |_n_,'/_n_//_/   |_n_,' \_,' \_,'/_/
                                <
                                 ...'

    WAFW00F - Web Application Firewall Detection Tool

    By Sandro Gauci && Wendel G. Henrique

Checking http://www.poupex.com.br
The site http://www.poupex.com.br is behind a Citrix NetScaler
Number of requests: 1
```

Detection of WAF with wafwoof

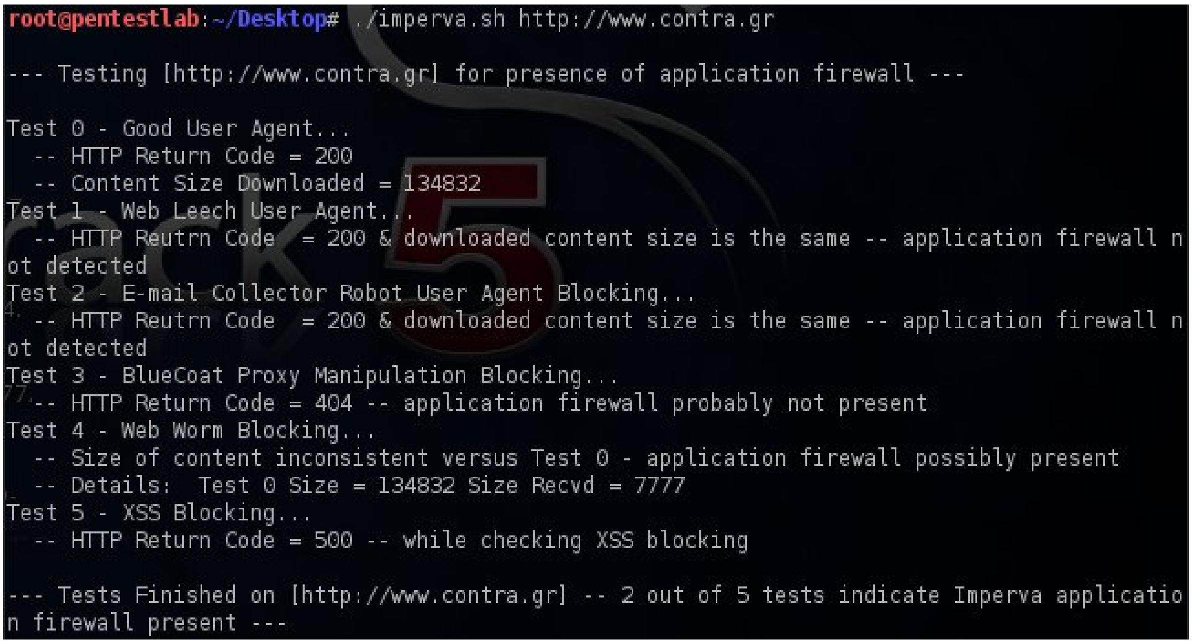
Nmap also can be used for this purpose as it contains a script that can detect a web application firewall.Specifically we run the script against the same website as above and the results were the following:



```
root@pentestlab:~# nmap  -p 80 --script http-waf-detect.nse poupex.com.br

Starting Nmap 6.01 ( http://nmap.org ) at 2013-01-13 03:52 GMT
Nmap scan report for poupex.com.br (200.252.149.141)
Host is up (0.23s latency).
PORT   STATE SERVICE
80/tcp open  http
| http-waf-detect: IDS/IPS/WAF detected:
|_poupex.com.br:80/?p4yl04d3=<script>alert(document.cookie)</script>

Nmap done: 1 IP address (1 host up) scanned in 5.75 seconds
```

WAF detection via Nmap

Finally there is a script that is capable only to detect Imperva WAF installations.



```
root@pentestlab:~/Desktop# ./imperva.sh http://www.contra.gr

--- Testing [http://www.contra.gr] for presence of application firewall ---

Test 0 - Good User Agent...
  -- HTTP Return Code = 200
  -- Content Size Downloaded = 134832
Test 1 - Web Leech User Agent...
  -- HTTP Reutrn Code  = 200 & downloaded content size is the same -- application firewall n
ot detected
Test 2 - E-mail Collector Robot User Agent Blocking...
  -- HTTP Reutrn Code  = 200 & downloaded content size is the same -- application firewall n
ot detected
Test 3 - BlueCoat Proxy Manipulation Blocking...
  -- HTTP Return Code = 404 -- application firewall probably not present
Test 4 - Web Worm Blocking...
  -- Size of content inconsistent versus Test 0 - application firewall possibly present
  -- Details:  Test 0 Size = 134832 Size Recvd = 7777
Test 5 - XSS Blocking...
  -- HTTP Return Code = 500 -- while checking XSS blocking

--- Tests Finished on [http://www.contra.gr] -- 2 out of 5 tests indicate Imperva applicatio
n firewall present ---
```

Detection of Imperva WAF

Conclusion

In this article we examined some methods and tools for detection of web application firewalls.This is an important process that must be done in every web application penetration test during the information gathering stage in order to ensure that the results from the attacks that will performed are accurate.Also by having the knowledge that a WAF is in place the penetration tester can try different techniques in order to bypass the protections and to exploit any weaknesses in the web application.