Web Sunuculara Yönelik DOS/DDOS Saldırıları

Giriş

Web, siber dünyada bizleri temsil eden hayatımızın ayrılmaz bir parçası oldu. Çok kullanılıyor bu yüzden göz önünde ve hackerların dikkatini çekiyor. Sistemlerde açıklık bulamayan hackerlar çoğu zaman "ya benimsin ya hiç kimsenin" mantığıyla fazla bilgi ve beceri gerektirmeyen DDOS'a başvuruyor ve web sunucuların işlevsiz kalmasına neden oluyorlar.

Bu yazıda web'in ve buna altyapı sağlayan HTTP'nin DOS saldırıları karşısındaki durumunu inceleyeceğiz.

HTTP'e Giriş

HTTP(Hypertext Transfer Protocol) OSI modelinde uygulama katmanında yer alan iletişim protokolüdür. Günümüzde zamanımızın çoğunu geçirdiğimiz sanal dünyada en sık kullanılan protokoldür.(%96 civarında)

HTTP Nasıl Çalışır?

Http'nin istemci-sunucu mantığıyla çalışan basit bir yapısı vardır. Önce TCP bağlantısı açılır, kullanıcı istek(HTTP isteği) gönderir sunucu da buna uygun cevap döner ve TCP bağlantısı kapatılır.

İstemci(kullanıcı) tarafından gönderilen istekler birbirinden bağımsızdır ve normalde her HTTP isteği için bir TCP bağlantısı gerekir.

HTTP'nin basitliğinin yanında günümüz web sayfaları sadece http sunuculardan oluşmaz, çoğu sistemin bilgilerini tutmak için kullanılan veritabanları da artık web sayfalarının vazgeçilmez bileşeni olmuştur.

Yukarıdaki resim klasik bir web sayfasını temsil eder. Buna göre web sayfalarımız ister web sunucuyla aynı makinede olsun ister başka bir makinede olsun bir veritabanına bağımlıdır.

Web'in çalışma mantığı istek ve cevaplardan oluşur, istekler ve bunlara dönülecek cevaplar farklıdır. Bu konuda detay için HTTP RFC'si 2616 incelenebilir.

Klasik bir HTTP isteği

```
GET /docs/1.3/keepalive.html HTTP/1.1
Host: httpd.apache.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.5) Gecko/20091102
Firefox/3.5.5
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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.lifeoverip.net
```

HTTP isteğine dönebilecek cevaplar

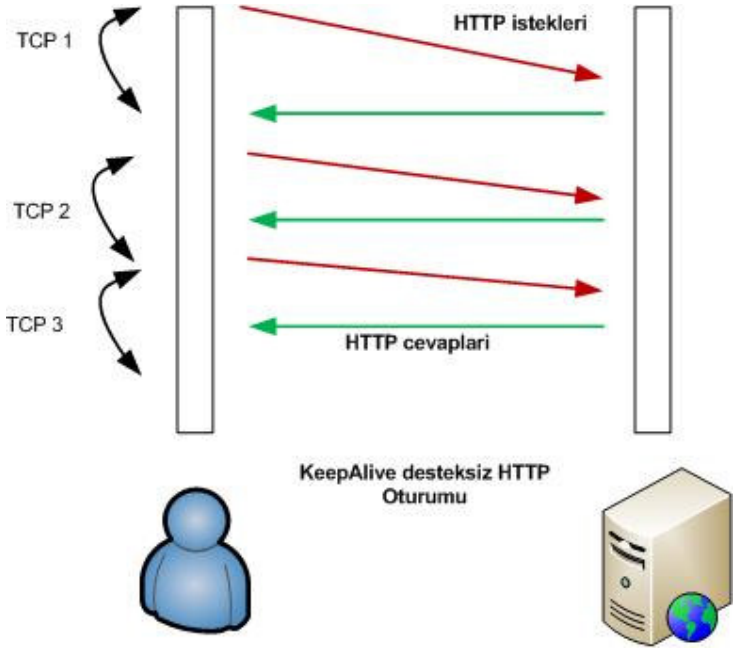
```
HTTP/1.1 200 OK
Date: Fri, 04 Dec 2009 09:09:34 GMT
Server: Apache/2.3.4 (Unix) mod_fcgid/2.3.2-dev
Content-Location: keepalive.html.en
Vary: negotiate,accept-language,accept-charset,Accept-Encoding
TCN: choice
Accept-Ranges: bytes
Content-Encoding: gzip
Content-Length: 1752
Keep-Alive: timeout=30, max=100
Connection: Keep-Alive
Content-Type: text/html
Content-Language: en
```

Yapılan isteğin çeşidine göre sunucudan dönecek cevap da farklı olacaktır. Mesela istenen dosya sistem üzerinde yoksa 404 cevabı, gelen istek izni olmayan bir dosyayı istiyorsa 403 forbidden cevabı dönecektir.

HTTP ve TCP ilişkisi

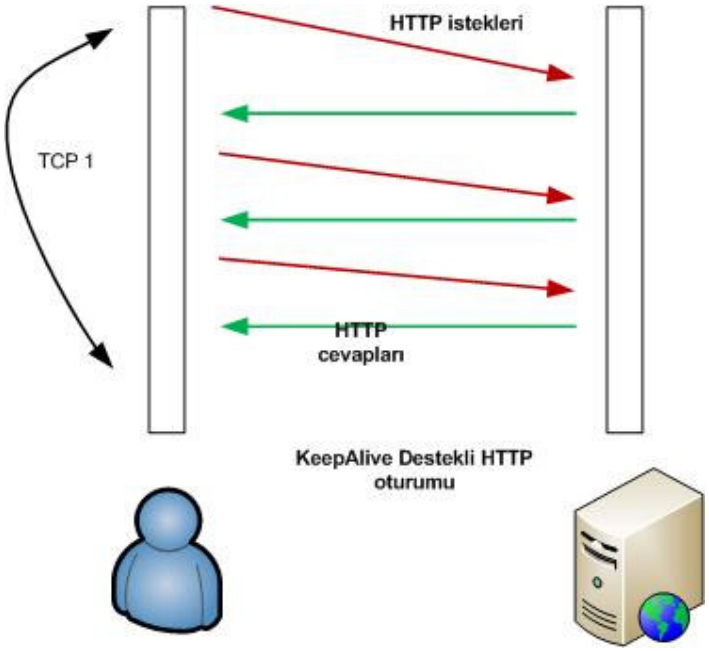
HTTP TCP kullanan bir protokoldür. Her HTTP bağlantısı öncesinde mutlaka TCP bağlantısı kurulmalıdır. Basit bir hesapla her HTTP bağlantısı için ortalama 10 adet TCP paketi gidip gelmektedir(3 adet TCP bağlantı başlangıcı, 4 adet TCP bağlantı koparılması, 2-3 adet de HTTP isteği ve buna dönecek cevap paketlerinin taşındığı TCP paketleri).

Bu da klasik HTTP kullanımında performans sorununu beraberinde getirir. Günümüzde normal bir haber portalının yüklenmesi için ortalama 40-50 HTTP GET isteği gönderilmektedir. Bunu da hesaplarsak portal sayfasının açılması için ortalama 50X10=500 TCP paketinin gidip gelmesi gerekir ki bu değer haber kullanıcıyı okumaktan vazgeçirecek kadar fazladır.



Klasik bir HTTP Oturumu

HTTP'de bu performans sorununu aşabilmek için çeşitli yöntemler geliştirilmiştir. Bunların başında HTTP KeepAlive(persistent connection) özelliği gelmektedir. HTTP Keep Alive özelliği her HTTP isteği için ayrı bir TCP bağlantısı açmak yerine bir adet TCP bağlantısı içerisinden belirli sayıda (5, 10, ..) HTTP isteğinin aktarılabilmesini sağlar.

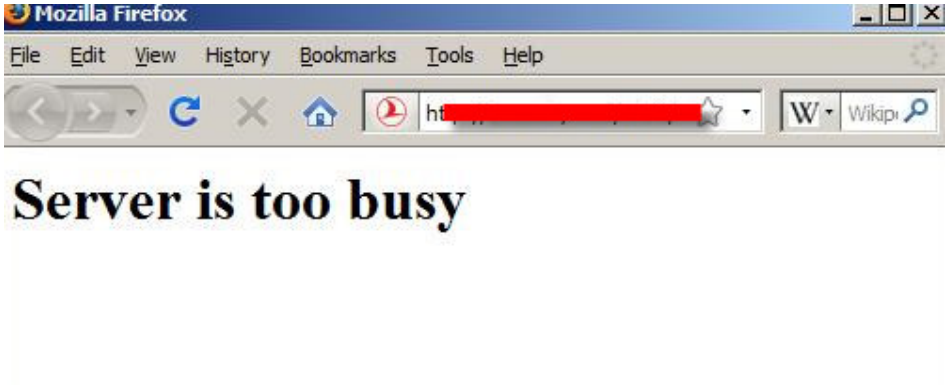


Http Pipelining

pipelining http isteklerinin eş zamanlı olarak gönderilmesi işlemidir, genellikle Keep Alive kavramıyla karıştırılır fakat birbirlerinden farklı kavramlardır. Klasik http bağlantılarında önce istek gönderilir, cevap beklenir, sonra tekrar istek gönderilir, cevabı beklenir. Pipelining kullanıldığında cevapları beklemeksizin birden fazla http isteği eş zamanlı olarak gönderilebilir bu arada istenirse KeepAlive özelliği kullanılarak her istek için ek bir TCP bağlantısı açılmaz.

Web Sunuculara Yönelik DOS Saldırıları

Web sunucularına yönelik DOS/DDOS saldırılarında amaç sayfanın işlevsiz kalması ve o sayfa üzerinden verilen hizmetlerin kesintiye uğratılmasıdır. DOS/DDOS'a maruz kalan web sunucularında çalışan web sayfalarında genelde aşağıdakine benzer bir hata ile karşılaşılır, eğer saldırı yoğunluğu yüksekse sayfa hiç gelmeyebilir.



Web sunuculara yönelik yapılacak DOS saldırıları temelde iki türden oluşur;

- kaba kuvvet saldırıları(Flood)
- tasarımsal/yazılımsal eksikliklerden kaynaklanan zaafiyetler

Kaba kuvvet DOS/DDOS Saldırıları

Bu tip saldırılarda sunucu üzerinde ne çalıştığına bakılmaksızın eş zamanlı olarak binlerce istek gönderilir ve sunucunun kapasitesi zorlanır. Literatürde adı "GET Flood", "POST Flood" olarak geçen bu saldırılar iki şekilde yapılabilir.

Bir kişi ya da birden fazla kişinin anlaşarak belli bir hedefe eş zamanlı yüzlerce, binlerce istek gönderir ya da bu işi hazır kölelere(zombie) devredilerek etki gücü çok daha yüksek Dos saldırıları gerçekleştirilir.

İlk yöntemde bir iki kişi ne yapabilir diye düşünülebilir fakat orta ölçekli çoğu şirketin web sayfası tek bir kişinin oluşturacağı eşzamanlı yüzlerce isteğe karşı uzun süre dayanamayacaktır. Güzel olan şu ki bu tip saldırıların gerçekleştirilmesi ne kadar kolaysa engellemesi de o kadar kolaydır(güvenlik duvarları/IPS'lerin rate limiting özelliği vs)

İkinci yöntem yani Zombi orduları(BotNet'ler) aracılığıyla yapılan HTTP Flood saldırıları ise binlerce farklı kaynaktan gelen HTTP istekleriyle gerçekleştirilir. Gelen bağlantıların kaynağı dünyanın farklı yerlerinden farklı ip subnetlerinden gelebileceği için network seviyesinde bir koruma ya da rate limiting bir işe yaramayacaktır.

Yazılımsal ya da tasarımsal eksikliklerden kaynaklanan DOS/DDOS Saldırıları

Tasarımsal zaafiyetler protokol düzenlenirken detaylı düşünülmemiş ya da kolaylık olsun diye esnek bırakılmış bazı özelliklerin kötüye kullanılmasıdır.

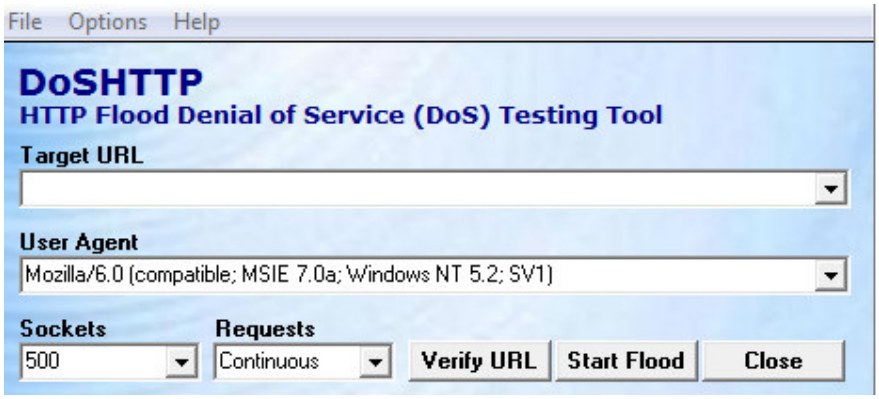
Tasarımsal zaafiyetlerden kaynaklanan DOS saldırılarına en iyi örnek geçtiğimiz aylarda yayınlanan Slowloris aracıdır. Bu araçla tek bir sistem üzerinden Apache HTTP sunucu yazılımını kullanan sistemler rahatlıkla devre dışı bırakılabilir.

Benzeri şekilde Captcha kullanılmayan formlarda ciddi DOS saldırılarına yol açabilir. Mesela form üzerinden alınan bilgiler bir mail sunucu aracığılıyla gönderiliyorsa saldırgan olmayan binlerce e-posta adresine bu form üzerinden istek gönderip sunucunun mail sistemini kilitleyebilir.

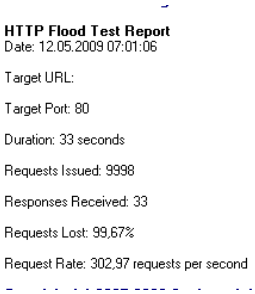
Zaman zaman da web sunucu yazılımını kullanan ve web sayfalarını dinamik olarak çalıştırmaya yarayan bileşenlerde çeşitli zaafiyetler çıkmaktadır. Mesela yine geçtiğimiz günlerde yayınlanan bir PHP zaafiyeti (PHP "multipart/form-data" denial of service)ni kullanılarak web sunucu rahatlıkla işlevsiz bırakabilir. Bu tip zaafiyetler klasik sınır koruma araçlarıyla kapatılamayacak kadar karmaşıktır. Yazılımları güncel tutma, yapılandırma dosyalarını iyi bilme en iyi çözümdür.

Web sunucularına yönelik performans test araçları

DDOS korumasında web sunucularımızı olası bir DOS/DDOS'a maruz kalmadan test edip gerekli ayarlamaları, önlemleri almak yapılacak ilk işlemdir. Bunun için saldırganların hangi araçları nasıl kullanacağını bilmek işe yarayacaktır. Zira günümüzde saldırgan olarak konumlandırdığımız kişilerin eskisi gibi uzman seviyesinde bilgi sahibi olmasına gerek kalmamıştır, aşağıda ekran görüntüsünden görüleceği gibi sadece hedef ip/host girilerek etkili bir dos saldırısı başlatılabilir.



Basit bir adsl hattından yapılan deneme ve sonuçları



Bu yazıda saldırganların kullanıkları araçların isimlerini verilmeyecektir, aynı işi yapan ve çeşitli kısıtlamalara sahip olan bazı araçların isimlerini vermekteyse bir sakınca görmüyorum. Benim kendi sistemlerimi test etmek istediğimde kullandığım temel araçlar:

ab, siege, http_load, hping ve curl. Bu araçlarla istediğiniz türde HTTP DOS saldırı testleri gerçekleştirip sunucunuzun durumunu inceleyebilirsiniz.

Ab(ApacheBenchmark) kullanarak yük testi

labs-lifeoverip#ab -c 100 -p post1.txt -T application/x-www-form-urlencoded -n 10000 -r -q http://blog.lifeoverip.net/searcme.php

ya da keepalive(-k parametresi kullanılarak) istekler göndererek TCP (layer 4) seviyesinde rate limitin kullanan sistemler atlatılabilir.

DOS/DDOS Saldırılarından Korunma Yöntemleri

Koruma konusunda bilinmesi gereken en temel kanun yapılan saldırının şiddeti (kullanılan bandwidth) sizin sahip olduğunuzdan yüksekse hiçbir şey yapamayacağınızdır. Bu durumda iş hizmet aldığınız telekom firmasına düşer.

Web sunuculara yönelik DOS/DDOS saldırılarından korunma diğer DOS/DDOS yöntemlerine göre daha zordur(synflood, udp flood, smurf vs).

Diğer saldırı yöntemleri genelde L4(TCP/UDP/ICMP) seviyesinde gerçekleştiği için ağ koruma cihazları(Router, Firewall, NIPS)tarafından belirli oranda engellenebilir fakat HTTP üzerinden yapılan DOS saldırılarında istekler normal kullanıcılardan geliyormuş gibi gözüktüğü için ağ güvenlik cihazları etkisiz kalmaktadır. Yinede web sunucular ve önlerine koyulacak ağ güvenlik cihazları iyi yapılandırılabilirse bu tip saldırılardan büyük oranda korunulabilir.

- Kullanılan web sunucu yazılımı konfigürasyonunda yapılacak performans iyileştirmeleri
- İstekleri daha rahat karşılayacak ve gerektiğinde belleğe alabilecek sistemler kullanılmalı Loadbalancer, reverseProxy kullanımı(Nginx gibi)
- Firewall/IPS ile belirli bir kaynaktan gelebilecek max. İstek/paket sayısı sınırlandırılmalı(rate limiting kullanımı)
- Saldırı anında loglar incelenerek saldırıya özel bir veri alanı belirlenebilirse(User-Agent, Refererer vs) IPS üzerinden özel imzalar yazılarak bu veri alanına sahip paketler engellenebilir fakat bunun normal kullanıcıları da etkileyeceği bilinmesi gereken nir konudur.
- Web sunucunun desteklediği dos koruma modülleri kullanılabilir(Apache Mod_dosevasive)

Iyi yapılandırılmış bu modülle orta düzey DOS saldırılarının çoğu rahatlıkla kesilebilir. Fakat kullanırken dikkat edilmesi gereken bazı önemli noktalar vardır. Mesela DOS yaptığı şüphelenilen kullanıcılara HTTP 403 cevabı dönmek yerine doğrudan saldırı yapanları iptables(APF) ile bloklatmak sunucuyu gereksiz yere yormayacaktır.

Sonuç

Siber dünyada gün geçtikçe Web'in önemi artacak ve HTTP'e yönelik DOS/DDOS saldırıları da ciddi artışlar gösterecektir. DOS/DDOS saldırılarını engellemeye yönelik atılacak en sağlıklı adım kullanılan sistemleri iyi bilmek ve DOS/DDOS saldırısı kapınızı çalmadan kendinizi test etmek ya da ettirmektir. Bu konuda hazır çözüm sunan ticari ürünlerin onu yapılandıran kadar işlevsel olacağı unutulmamalıdır.