ÖN BİLGİ

Bu belge

- https://www.bgasecurity.com/makale/adli-bilisim-acisindan-dos-ve-ddos-saldirilari-ve-korunma-yontemleri/

resmi adresindeki veya linkin kırık olması ihtimaline karşın alternatif olarak

- https://www.includekarabuk.com/kitaplik/indirmeDeposu/Siber_Guvenlik_Teknik_Makaleler/Teori/BaskalarinaAitMakaleler/Adli%20Bili%C5%9Fim%20A%C3%A7%C4%B1s%C4%B1ndan%20Dos%20ve%20Ddos%20Sald%C4%B1r%C4%B1lar%C4%B1%20ve%20Korunma%20Y%C3%B6ntemleri.pdf

adresindeki makaleye çalışılarak elde edilen notlarımı kapsamaktadır. Bu çıkarılan notlar belgemde alıntılar ve/veya kişisel ilavelerim mevcuttur.

1)

Bir DOS saldırısı olduğunu nereden anlarız?
- Sistemimiz açılmıyordur, çalışmıyordur. :)

DOS Saldırısının tipini nasıl anlarız?
- Tcpdump, awk, sort, uniq araçlarının birlikte kullanımıyla.

(page 4)

2)

DDOS saldırılarında aşağıdaki iki aşama tatbik edilmelidir:

- Saldırıyı Engelleme
- Saldırının kim tarafından, ne şiddette, hangi araçlar ile ve hangi yöntem kullanılarak gerçekleştirildiğinin belirlenmesi

Sadece saldırıyı engelleme dışında ikinci aşama (analiz aşaması) da tatbik edilirse aynı saldırının tekrarı durumunda nasıl engelleyebileceğimize dair bir yol haritası elde etmiş oluruz ve onu kullanarak sıkıntıyı bertaraf edebiliriz.

(Page 5)

3)

DDOS Analizi için Gerekli Teçhizatın Kurulması
DDOS analizi için DDOS saldırısı sırasında sistemin otomatik olarak devreye girip saldırıya ait delil niteliğindeki tüm paketlerin kaydedilmesi gerekir. Saldırı anında bu paketler kaydedilebilirse saldırıya ait tüm detayları elde etmek mümkün olur. Paketleri kaydetmek için tcpdump ya da Wireshark kullanılabilir.

(Page 6)

4)

Saldırı Analizinde Kullanılan Araçlar
- tcpstat
- tcpdstat
- tcpdump, wireshark
- ourmon
- argus
- urlsnarf
- snort
- aguri
- cut, grep, awk, wc gibi UNIX araçları

(Page 10)

5)

DDOS Saldırılarında Delil Toplama
DDOS saldırısına uğrayan bir firma böylesi bir durumda daha önceden paketlerin kaydedildiği bir ortamı kurmuş olması gerekir. Fakat paket kaydetme işlemi kesinlikle aktif cihazlar olan IPS, DDOS Engelleme Sistemleri, Firewall'lar tarafından yapılmamalıdır.

Benim NOT: Aktif cihazlar ile yapılmasın denmesinin nedeni bence cihazları şişirip güvenliği tepetaklak devirmesinler diyedir. Paket kaydetmek için ayrı bir sunucu kullanılabilir.

Tüm paket detayları aşağıdaki gibi kaydedilebilir:

```
> tcpdump -s0
```

(Page 13)

6)

Paket Kaydetme
Paket kaydetme konusunda linux üzerindeki en uygun seçenek tcpdump'tır. Windows üzerindeki tercih edilebilecek seçenekler ise Wireshark, windump'tır.

NOT: 10 GB'lık paket kaydedebilecek ortamlarda klasik libpcap kütüphanesi yerine alternatif kütüphaneler tercih edilmelidir.

(Page 14)

7)

tcpdump ile DDOS Paket Kaydetme Komutu

```
> tcpdump -n -s0 -w ddosrecords.pcap -C 2000
```

-n : İsim – IP çözümlemesi yapmamayı sağlar.
-s0 : Alınacak data paketinin varsayılan limitleriyle kabul edilmesini sağlar. Yani bir paket maksimum 65535 byte olabilmektedir. -s0 ile o kadara kadar kabul edebilirsin, kayıtlara geçebilirsin demiş oluyoruz. 0 değerini değiştirerek byte limitini belirleyebiliriz.
-w : Gelen paketleri ekrana basmak yerine dosyaya yazdırmayı sağlar.
-C : Dosya boyutunu (limitini) elle girmeyi sağlar. Şayet -w'nin açtığı dosya -C'nin aldığı 2000'den büyükse ddosrecords.pcap dosyası kapatılır ve tcpdump'ın kendisi bir dosya açar.

(Page 15)

8)

DDOS Saldırı türünü belirlemek için önce gelen paketlerin hangi protokole ait olduğunu belirlememiz gerekir. Bunun için tcpdstat aracı kullanılabilir:

> tcpdstat -n ddos.pcap

Bu komutla gelen paketin hangi protokole ait olduğu saptandıktan sonra sıradaki işlem paketlerin nereden geldiği, yani kaynak IP'leridir. Bu IP'ler spoof edilmiş IP'lerden mi değil mi sorusuna yanıt aramaktır.

(Page 16, 19)

9)

SYN Flood'a dair izleri bulmak için aşağıdaki gibi tcpdump'ı kullanabiliriz:

SYNTAX:
> tcpdump -r okunacakDosya.pcap -n "tcp[tcpflags] & tcp-syn == tcp-syn"

Example:
> tcpdump -r ddos.pcap -n "tcp[13] & 2 != 0" // 2 != 0 olayını bilmiyorum.

-r : ddos.pcap adlı dosyayı okur. Böylece paketleri diğer parametredeki syn flag'ine göre filtreleyecektir ve ekrana sunacaktır.
-n : İsim – IP çözümlemesi yapmamayı sağlar.

(Page 20, http://docplayer.biz.tr/711443-Synflood-ddos-saldirilari.html)

10)

ACK Flood Saldırısına Dair İzleri Bulmak // ACK Bayraklı paketleri bulmak

> tcpdump -r ddos.pcap -n "tcp[13] & 16 != 0"

FIN Flood Saldırısına Dair İzleri Bulmak // FIN Bayraklı paketleri bulmak

> tcpdump -r ddos.pcap -n "tcp[13] & 1 != 0" and tcp port 80

HTTP GET Flood Saldırısına Dair İzleri Bulmak // GET komutlarına sahip paketleri bulmak

> tcpdump -r ddos.pcap -n tcp port 80 and \(tcp[20:2] = 18225 \)

(Page 21-23)

11)

Saldırının şiddetini belirleme hususunda iki kriter vardır:

- Gelen trafiğin bant genişliğini sömürme oranı
- Gelen trafiğin saniyede ilettiği paket sayısı (PPS değeri => Packet per second)

tcpstat tool'u ile trafik dosyaları üzerinde saldırının bant genişliğini ne oranda sömürdüğünü ve trafiğin PPS değerini detaylı olarak görebiliriz.

(Page 24)

12)

Saldırı analizinde saldırıda kullanılan IP adreslerinin gerçek IP'ler mi yoksa spoof edilmiş IP'ler mi olduğu rahatlıkla anlaşılabilmektedir.

Benim Not: 27 ve 28. sayfalarda spoof edilmiş IP'nin nasıl anlaşılacağı anlatılıyor ama anlamadım.

(Page 26)

13)

Gelen dos saldırılarından en çok kaynakları sömüren top 10 IP adresi aşağıdaki kabuk kodlamasıyla öğrenilebilir:

```
> tcpdump -r test.pcap -n | cut -f3 -d " " | cut -f1-4 -d "." | sort -n | uniq -c |
        awk -F " " '{print $2 "\t" $1}' | sort -rn -k 2 | head -10
```

Output:

```
reading from file test.pcap, link-type EN10MB (ethernet)
11.22.228.246   482196
11.22.243.10    62095
11.22.228.73    27515
11.22.241.138   24972
93.18.207.182   24761
11.22.28.78     13205
195.142.247.7   5041
18.89.192.37    4870
78.16.195.145   4268
78.86.3.178     4157
```

Sol sütun saldırı yapanların kaynak IP adreslerini, sağ sütun ise o IP adresinden kaç adet paket sistemize flood edildiğini göstermektedir.

Yukarıdaki kabuk kodunu anlamlandıralım. Normalde çıktıya yansıyan ifade orijinalde pcap dosyasında şöyle bir şey:

….....11.22.228.246….....
….....11.22.243.10….....
….....11.22.228.73….....
….....11.22.241.138….....
….....93.18.207.182….....
….....11.22.28.78….....
….....195.142.247.7….....
….....18.89.192.37….....
….....78.16.195.145….....
….....78.86.3.178….....

Aşağıdaki kod ile

cut -f3 -d " "

yukarıdaki aşikar görünen blok komple alınır. Bu alınan kısım aşağıdaki koda verilerek

cut -f1-4 -d "."

noktaya göre sütun ayrımına gidilir ve 1. sütundan 4. sütuna kadarki kısım alınır. Bu alınan kısım aşağıdaki koda verilerek

sort -n

numerik sırada satırlar dizilir. Sıralanan kayıtlar aşağıdaki koda verilerek

uniq -c

kayıtlar unique'leştirilir ve unique olan her bir kayıt kaç defa önceden tekrar etmişse sayısı prefix olarak satırların başına eklenir. Ardından bu yeni satır dizisi aşağıdaki koda verilerek

awk -F " " '{print $2 "\t" $1}'

boşluk ayraç olarak kullanılır ve ayrılan iki parçadan prefix sağa, ip adresi sola konur. Aralarına da bir tab boşluk konur. Daha sonra bu satır dizisini aşağıdaki koda verip

sort -rn -k 2

-r parametresi ile satırları tersden düze ve -n parmetresi ile de numeric olarak tersden düze sıralama işlemini yaptırırız. -k parametresi ile de var olan sıralamanın üzerine ikinci field üzerinden tekrar numerik sıralama yaparız. Yani çift sıralama yapılıyor (Benim Not: Ne gerek var çift sıralamaya bilmiyorum). Daha sonra sıralanmış en son kayıtlar aşağıdaki koda verilerek

head -10

ilk 10 kayıt çekilir ve ekrana yansıtılır:

```
reading from file test.pcap, link-type EN10MB (ethernet)
11.22.228.246   482196
11.22.243.10     62095
11.22.228.73     27515
11.22.241.138    24972
93.18.207.182    24761
11.22.28.78      13205
195.142.247.7     5041
18.89.192.37      4870
78.16.195.145     4268
78.86.3.178       4157
```

Yani özetle *cut -f3 -d " "* ile pcap dosyasının ortalarında yer alan IP bloğunu çekmiş bulunmaktayız. Sonra *cut -f1-4 -d "."* ile IP'nin ip formatına uygunluğunu teyit ederek tekrar IP bloğunu çekmiş bulunmaktayız. *sort -n* ile seçtiğimiz tüm satırları (IP bloklarını) numeric olarak sıralıyoruz. Sonra *uniq -c* ile tüm satırlardaki unique IP adreslerini seçip tekrar sayılarını sol sütuna, o kadar tekrar eden ilgili IP adresini de sağ sütuna koyarız. Bu sütunları yer değiştirmek için *awk -F " " '{print $2 "\t" $1}'* kodunu kullandık. Ardından *sort -rn -k 2* ile anlam veremediğim bir çift sıralama yapıldı. En sonunda da *head -10* ile sıralanan IP blokları + Tekrar Sayısı satırlarından ilk 10'u çekilir ve ekrana basılır.

(page 29)

14)

HTTP GET Flood saldırılarında IP Spoofing yapmak mümkün değildir. Çünkü HTTP protokolü TCP üzerinde koşar. Yani TCP 3 yollu el sıkışma yapılmadan HTTP çalışmaz. O yüzden HTTP GET Flood saldırılarında kaynak IP adresini tespit etmek %99 mümkündür.

Benim Not: %1'lik IP adresi tespit edilemez payı muhtemelen her ihtimale binaen konulmuş olmalı. İleride ne olacağı belli olmaz sonuçta....

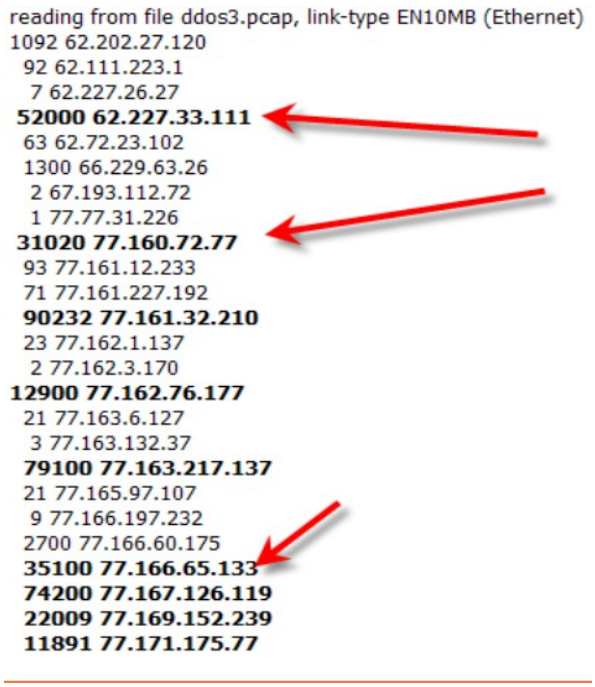
(Page 30)

15)

HTTP GET Flood yapılan sistem üzerinde saldırı izlerini tcpdump ile bulalım:

```
> tcpdump -n -r ddos.pcap tcp port 80 and \(tcp[20:2] = 18225 \) | sort -k3 -n | cut -f3 -d " "
        | cut -f1,2,3,4 -d "." | sort -n | uniq -c
```

Output:



-n : IP Adres - Domain çözümlemesi yapmamayı sağlar.
-r : Dosyadan okuma yapmayı sağlar.

Kalınlaştırılmış ve aynı zamanda kırmızı oklarla gösterilmiş satırların ilk sütununda dikkat edersen sistemimize fazla (aşırı) sayıda paket geldiği görülmekte. Diğer satırlarda ise sistemimize gayet düşük sayıda paket geldiği görülmekte. Dolayısıyla aşırı paket sayılarından sistemimize paket hücumu yapıldığını, yani saldırı yapıldığını anlayabiliriz.

Benim Not: tcp[20:2] = 18225 kısmındaki rakamlar, yani tcpflag değeri tcp header syntax'ı kullanılarak bit bazında yapılan matematiksel işlemler sonucu elde ediliyor. tcpdump'ın man sayfasında bu tür hesaplamalara dair temel bir örnek izah edilerek açıklanmış.

16)

Wireshark'ın display filter özelliği kullanılarak gelen paketleri daraltabilmekteyiz. Ayrıca paketlerin paket başlıklarına ve payload'larına bakabildiğimiz için boş olan HTTP Get isteklerini görebiliriz. Eğer HTTP GET paketi boş geliyorsa o paketin bir tarayıcıdan gelmediğini, pek de akıllı olmayan otomatize bir tool'dan geldiğini tespit edebiliriz. Yani saldırıya maruz kaldığımızı çözebiliriz.

(page 35)

17)

User Agent Nedir?

İstemcilerin web sunucularına gönderdiği HTTP Request paketlerinde yer alan bir HTTP Request header'ı olan User Agent sunucuların kendisine ziyaret eden istemcilere uygun formatta içerik sunabilmeleri için kullanılmaktadır. Yani örneğin telefondan bir web sitesine bağlandığımızda web sitesi yazılımı kendisine gelen talepteki User Agent'a bakarak talebin telefondan geldiğini görür ve sitenin mobil versiyonunu cevap olarak döner. User Agent'ın bu yararlı kullanımı bazı sakıncalar da doğurmaktadır. Örneğin değer olarak istemcinin tarayıcı türünü ve işletim sistemi türünü taşıdığından ve sunucuya gönderdiğinden istemcinin gizliliğini ifşa etmektedir. Aşağıda bir User Agent header'ının taşıdığı veri örneği görmektesiniz:

User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko

NOT: User Agent header'ı HTTP Request'te vardır, fakat HTTP Response'ta yoktur.

https://en.wikipedia.org/wiki/User_agent
https://www.httpwatch.com/httpgallery/headers/

18)

Port vs. Socket

* The port is for routing to the “correct” socket on the computer. That is, one port can redirect a coming connection to different sockets in system (Note: Correct socket means nearly correct “process”).
* A socket represents a single connection between two applications in different two machine.
* A port represents an endpoint or "channel" for network communications.
* While a port consist of just a number, a socket consists of three things:

- An IP address
- A Transport Layer protocol (tcp, udp,...)
- A port number

For example:

1030 is a port.
(10.1.1.2 , TCP , port 1030) is a socket.

Benim Not:

```
                                                        ---------- socket 1 (process)
[router]----------~network~------------Computer------[port]---------- socket 2 (process)
                                                        --------- socket 3 (process)
```

Port sanal bir kapı, socket ise sanal bir alt kapı.

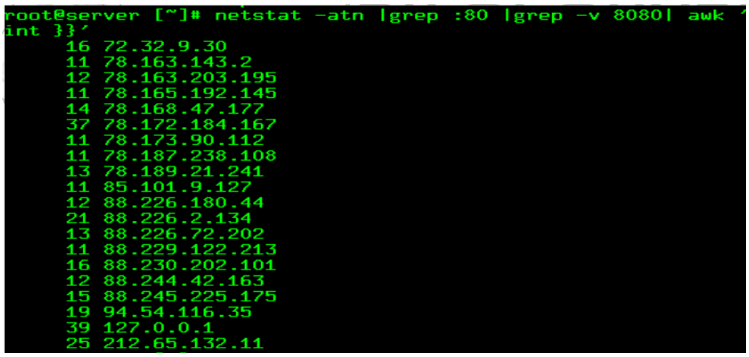
http://programmers.stackexchange.com/questions/171734/difference-between-a-socket-and-a-port

http://stackoverflow.com/questions/152457/what-is-the-difference-between-a-port-and-a-socket
18)

netstat ile linux sistemimize gelen bağlantılardan bir dos atağı yaşayıp yaşamadığımızı analiz etme:

> netstat -atn | grep ":80" | grep -v ":8080" | awk '{print $5}' | awk -F: '{print $1}' | sort -n
| uniq -c | awk '{if ($1 > 10) {print}}'

Output:



netstat
==========
-a parametresi : display all sockets
- t parametresi : only tcp sockets
- n parametresi : don't resolve IP addreses to domain names
==========

Şimdi kullandığımız kabuk kodunu sırasıyla inceleyelim:

netstat -atn // TCP socketlerini sırala.

grep ":80" // TCP socketlerinden 80.portu dinleyenleri çek.

NOT: :80 string'inin yer aldığı satırlar çekilir fakat, :8080 stringinin yer aldığı satırlar da istenmeden çekilmiş olur.

grep -v ":8080" // TCP socketlerinden :8080 string'ini içermeyenleri çeker. Yani :80'ler
// seçilir ve :8080'ler elenir.

awk '{print $5}' // Var olan satırlardan 5. kolonu çeker (varsayılan ayraç bir karakterlik
// boşluktur). Böylece [IP Numarası**:**Port Numarası] kayıtları çekilmiş olur.

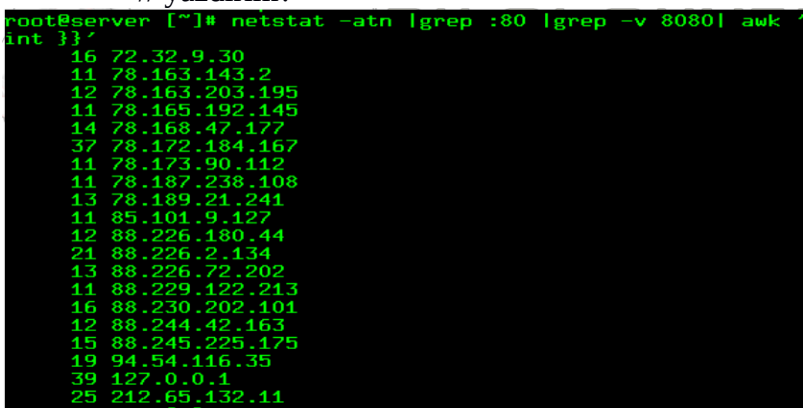
awk -F: '{print $1}' // İki nokta üst üsteye göre böler ve 1. kolonu çeker. Böylece IP'yi port
// numarasından ayırarak stdout'a verir).

sort -n // Gelen IP verisi içeren satırlar numeric olarak sıralanır.

uniq -c // IP numaraları unique'leştirilir ve tekrar etme sayıları başlarına konur.

awk '{if ($1 > 10) {print}}' // Önceki koddan gelen Tekrar Sayısı + IP şeklindeki satırlardan tekrar
// sayıları 10'dan büyük olanlar ekrana Tekrar Sayısı + IP şeklinde

// yazdırılır.



netstat aracılığıyla ve birkaç filtreleme işlemi sonrası IP 'lerden gelen tcp bağlantı sayılarını görmüş oluruz. Bir anormal bağlantı sayısı (tekrar sayısı) gördüğümüzde olası bir flood saldırısı altında olduğumuzu düşünebiliriz. (örn; SYN Flood, FIN Flood, XMAS Flood, vs...)

http://www.staff.science.uu.nl/~oostr102/docs/nawk/nawk_23.html
(Page 34)

19)

İstemci tarayıcısındaki User Agent parametresini manipule edebilir ya da boş bir veriyle doldurabilir ve o şekilde web sitelerine bilgisini ifşa etmeden girebilir. Ancak böylesi atraksiyonlara girmeyen istemciler göz önüne alındığında, yani normal koşullarda istemciden gelen HTTP Request'in User Agent header'ı dolu gelecektir. User Agent'ın dolu geleceğine dair bir kanaatte olduğumuzu varsayarsak eğer sistemimize gelen HTTP Request paketlerinden birinin User Agent header'ı Wireshark'tan baktığımızda boş görünüyorsa bir HTTP flood saldırısı ile karşı karşıyayız demektir. Çünkü User Agent header'ın boş gelmesinin nedeni gelen HTTP Request paketinin manuel olarak (yani pek de akıllı olmayan bir tool ile) oluşturulduğunu bize söyler. Boş olduğunu görerek paketin tarayıcıdan gelmediğini anlarız. Böylece saldırıda olduğumuzu düşünebiliriz. Lakin istemci tarayıcısının User Agent parametresini örneğin boşaltmışsa, yani güvenlik önlemi almışsa o kullanıcıdan gelen paketteki User Agent header'ı boş gelecektir ve biz HTTP Flood saldırısı yapılıyor sanacağız. Esasında o paket güvenlik tedbiri almış bir kullanıcının normal bir web sitesi ziyaretinden başka bir şey olmayacaktır. Dolayısıyla false positive bir durumla karşı karşıya kalmış olacağız. Yanlış yere alarma geçeceğiz.

Benim NOT: BGA'nın 35. sayfasında söylediği kıt cümleleri internetten açarak bu paragrafı oluşturdum.

http://www.howtogeek.com/113439/how-to-change-your-browsers-user-agent-without-installing-any-extensions/

(Page 35)