

BGA

**BİLGİ GÜVENLİĞİ
AKADEMİSİ**

www.bga.com.tr

DOS/DDOS Saldırıları, Savunma Yolları ve Çözüm Önerileri

Huzeyfe ÖNAL <huzeyfe.onal@bga.com.tr>

İÇERİK TABLOSU

DOS/DDOS Saldırıları	7
Genel Tanımlar.....	7
DOS/DDOS Saldırıları	7
Neden Kaynaklanır?	8
(D)DOS Çeşitleri.....	8
Sonuçlarına göre (D)DOS Çeşitleri	8
Yapılış şekline göre (D)DOS Çeşitleri	9
Synflood (D)DOS Saldırıları	9
Synflood yapıldığını nasıl anlarsınız?	10
HTTP Get Flood Saldırıları.....	10
Internet'i durdurma(DNS DOS)Saldırıları	11
Türkiye'de DOS Saldırıları Açısından Güncel Durum.....	12
Korunma Yolları ve Yöntemleri	13
Router(Yönlendirici) Seviyesinde Koruma.....	13
Güvenlik Duvarı Seviyesinde Koruma.....	14
Güvenlik duvarı üzerinde ön tanımlı ayarların değiştirilmesi	14
TCP ve UDP paketleriyle ilgili oturum bilgilerinin varsayılan değerlerinin değiştirilmesi.....	15
Saldırı Tespit ve Engelleme (NIPS) Seviyesinde Koruma.....	15
Basit bir SYNflood önlemi	16
Web Sunuculara Yönelik Koruma	16
İşletim sistemleri üzerinde basit koruma ayarları	17
Synflood saldırılarına karşı koruma.....	17
Ağınızdan DOS/DDOS yapılmasını Engelleme.....	17
SynFlood DDOS Saldırıları.....	18
Nedir?	18
SYN Flood Saldırılarının Temel Kaynağı	18
Temel TCP bilgisi.....	19
TCP Bağlantılarında Bayrak Kavramı (TCP Flags)	19
TCP Oturum başlatma	20
TCP Oturum Sonlandırma.....	22
Bağlantı sonlanması esnasında oluşan durumlar:	22
SynFlood Saldırıları Nasıl Gerçekleştirilir?	24
Backlog queue kavramı	24
Synflood matematik hesabı	25
SynFlood DDOS Saldırısı Nasıl Anlaşılır?	25
SynFlood Saldırı Testleri	26
Scapy ile basit Synflood aracı	27
Hping Kullanarak SYNflood Testleri.....	27
SYNFlood Saldırısı Esnasında Yapılacaklar	28
Ülkeye göre IP bloklama	29

Türkiye IP Blokları Nereden Edinilebilir	29
Synflood Önleme Yöntem ve Çeşitleri	29
Syncookie Nasıl çalışır?	30
Syn cookie dezavantajları.....	30
SynCache nasıl çalışır?	31
Syn Proxy nasıl çalışır?	31
Ağınızdan SYN Flood yapılmasını Engelleme	32
OpenBSD PF ile URPF kullanımı?.....	32
Snort Kullanarak SYNflood saldırılarının Belirlenmesi.....	33
İşletim Sistemlerinde SYNflood Koruması	33
Backlog Queue değerini arttırma	33
Linux için backlog queue değerini arttırma.....	34
FreeBSD backlog queue değeri arttırma.....	34
Zaman aşımı(Timeout) değerlerini düşürme	34
Syncookie aktivasyonu.....	34
Web Sunuculara Yönelik DoS/DdoS Saldırıları	36
HTTP'e Giriş	36
HTTP Nasıl Çalışır?	36
Klasik bir HTTP isteği.....	37
HTTP isteğine dönebilecek cevaplar.....	37
HTTP ve TCP ilişkisi	37
Http Pipelining	39
Web Sunuculara Yönelik DOS Saldırıları	39
Kaba kuvvet DOS/DDOS Saldırıları	40
Yazılımsal ya da tasarımsal eksikliklerden kaynaklanan DOS/DDOS Saldırıları	41
Web sunucularına yönelik performans test araçları	41
Ab(ApacheBenchmark) kullanarak yük testi	42
DOS/DDOS Saldırılarından Korunma Yöntemleri	42
Sonuç.....	43
DNS Hizmetine Yönelik DoS/DdoS Saldırıları.....	44
DNS Hakkında Temel Bilgiler	45
DNS Nedir?	45
DNS Protokol Detayı	45
Detaylı DNS başlık bilgisi incelemesi için http://www.networksorcery.com/enp/protocol/dns.htm adresinden faydalınabilir.	46
DNS Paket boyutu.....	46
DNS Kayıt Tipleri.....	47
DNS Sorgulamaları	48
DNS Sorgulamalarını Yorumlama - Dig	48

DNS Sorgu Çeşitleri	51
DNS Sunucu Yazılımları	55
DNS Sunucu Tipini Belirleme	55
İsteğe Göre DNS Paketi Üretmek	56
DNS Güvenlik Zafiyetleri.....	59
2011 Yılı ISC Bind Yazılımında Çıkmış Güvenlik Zafiyetleri	59
DNS Protokolünde IP Sahteciliği (IP Spoofing)	60
Kaynak Portun Rastgeleliğinin Sorgulanması	61
DNS Transaction ID Değerinin Rastgeleliğinin Sorgulanması	61
DNS ve TCP İlişkisi	62
DNS'e Yönelik DoS ve DDoS Saldırıları.....	66
Yazılım Temelli DoS Saldırıları	66
DNS Flood DoS/DDoS Saldırıları.....	67
DNS Flood DDoS Saldırıları	68
DNS Flood ve UDP Flood DDoS Saldırıları Arasındaki Farklar	68
Netstress Kullanarak DNS Flood DDoS Atağı Gerçekleştirme	70
Saldırılarda Sahte(spoofed) IP Adreslerinin Kullanımı	71
Bilinen DNS Sunucu IP Adreslerinden DNS Flood Gerçekleştirme	73
DNS Performans Ölçümü.....	74
Amplified DNS DoS Saldırıları	77
Adım Adım DNS Amplification DoS Saldırısı	77
Örnek DNS Amplified DoS Saldırısı	80
DNS Flood DDoS Saldırılarını Yakalama	80
DNS Flood DDoS Saldırılarını Engelleme.....	80
DNS Flood saldırılarını engellemek için kullanılan temel yöntemler:	80
Rate Limiting Yöntemi	80
DFAS.....	81
DNS Caching Cihazlarını Atlatma Saldırıları.....	83
DdoS Saldırı Analizi ve Adli Bilişim İncelemesi.....	85
DDoS Analizi İçin Gerekli Yapının Kurulması	85
Saldırı Analizinde Cevabı Aranılan Sorular.....	86
Alet Çantasında Bulunması Gereken Araçlar	86
DDoS Saldırı Tespit Sistemleri	86
DDoS Saldırılarındaki Delil Toplama.....	87
Paket Kaydetme.....	88
Tcpdump ile paket kaydetme	88
DDoS Saldırı Tipi Belirleme	88
TCP Bayrakları Kullanılarak Gerçekleştirilen DDoS Saldırıları	91
SYN Flood Saldırısı Analizi	91
Saldırının Şiddetini Belirleme.....	92
Saldırı Kaynağını Belirleme	93
Saldırıda Kullanılan Top 10 IP Adresi	95

HTTP GET Flood Saldırısında Kullanılan IP Adresleri	95
Saldırıda Kullanılan IP Adresleri Hangi Ülkeden?	97
Saldırı Paketlerini Pasif Snort Sisteminden Geçirme.....	97
NTP Servisi Kullanarak Gerçekleştirilen Amplification DDoS Saldırıları.....	99
Hping Nedir?	110
Hping'in kullanılacağı alanlar	110
Nasıl Edinebilirim?	110
Temel Hping Kullanımı	111
Hping versiyonu öğrenme	112
Hping Çalışma Modları.....	112
Hping ile paket gönderimi.....	112
TCP Paketleriyle Oynama	113
TCP'deki bayraklar ve hping parametreleri	114
Hping çıktısını yorumlama.....	115
Hping kullanımında port belirtimi	116
RST Bayraklı TCP paketleri oluşturmak.....	116
Aynı pakette birden fazla bayrak kullanımı	117
IP Paketleriyle Oynama	117
Hping ile spoof edilmiş paketler oluşturma(IP Spoofing).....	117
ICMP Paketleriyle Oynama.....	119
Broadcast ICMP Paketleri	121
UDP Paketleriyle Oynama	122
UDP kullanarak traceroute	122
Broadcast UDP Paketleri	123

Giriş

DOS/DDOS saldırıları internet dünyasının başlangıcından beri önemi hiç eksilmeyen bir tehdit olarak bilinmektedir. Güvenlik açıklıkları kapatılsa da TCP/IP protokolü yapısı değişmeden bu soruna kesin çözüm bulunamayacaktır.

Bu kitapta DOS, DDOS saldırıları hakkında genel bilgi, güncel örnekler ve çözüm yolları anlatılmıştır.

DOS/DDOS Saldırıları

Genel Tanımlar

DOS (Denial Of Service): Herhangi bir sistemi, servisi, ağı işlevini yerine getiremez hale getirmek için yapılan saldırılar

DDOS (Distributed Of Service) DOS saldırılarının organize şekilde birden fazla kaynakla yapılmasına DDOS denir.

Zombi: Ele geçirilmiş ve sahibinden habersiz şekilde çeşitli amaçlar için kullanılan bilgisayar sistemleri. Zombiler en önemli DDOS kaynaklarındanıdır.

Botnet (Robot Networks) Zombiler tarafından oluşturulan ve çeşitli amaçlarla kullanılan sanal bilgisayar ordularıdır. Zombiler botnetleri oluşturur, botnetler organize siber suçlarda sık kullanılan ara elemanlardır.

SYN: TCP başlığında bulunan bayraklardan biridir. TCP bağlantılarında ilk gönderilecek paket SYN bayrağına sahip olmalıdır. Oturumun başlatılması için gönderilir ve hedeften cevap olarak SYN-ACK bayraklı paket beklenir.

IP Spoofing: Saldırganın yakalanma riskini yok etmek için IP adresini olduğundan farklı göstermesi

DOS/DDOS Saldırıları

DOS/DDOS saldırıları günümüz internet dünyasının en temel sorunlarından biridir. İnternet'in ilk çıktığı günden beri çözülemeyen bu tehdit hâlihazırda kullanılan TCP/IP protokolüyle uzun süre çözülemeyecek kadar ciddi bir problemdir. DDOS saldırılarında ana amaç sistemi işlevsiz kılmaktır.



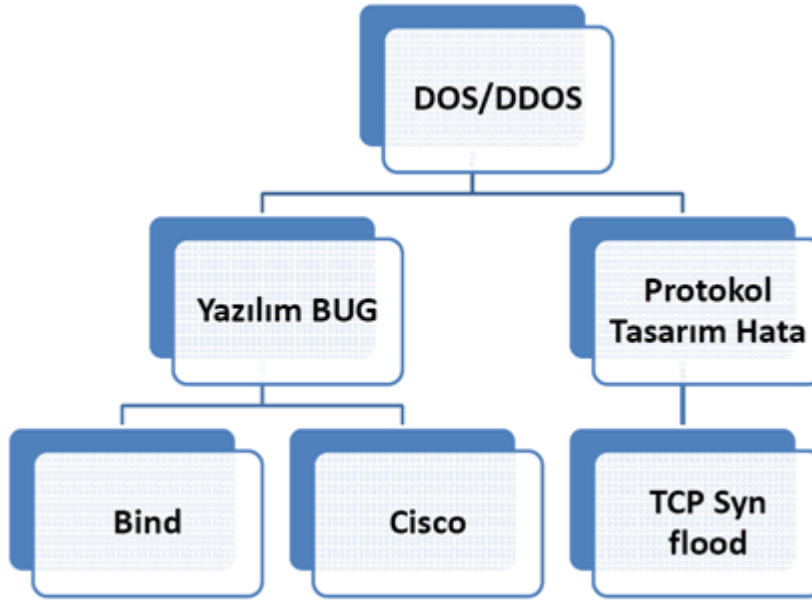
Yapılan saldırıya göre DOS'un etkileri aşağıdaki maddelerden biriyle sonuçlanabilir

- Web sayfasının ulaşılamaz olması

- Web sayfası/servislerinin isteklere geç cevap dönmesi
- Ağ performansında yavaşlama
- İşletim sistemlerinde CPU/Ram performans problemi
- Uyarı sistemlerinin çökmesi

(D)DOS Saldırıları

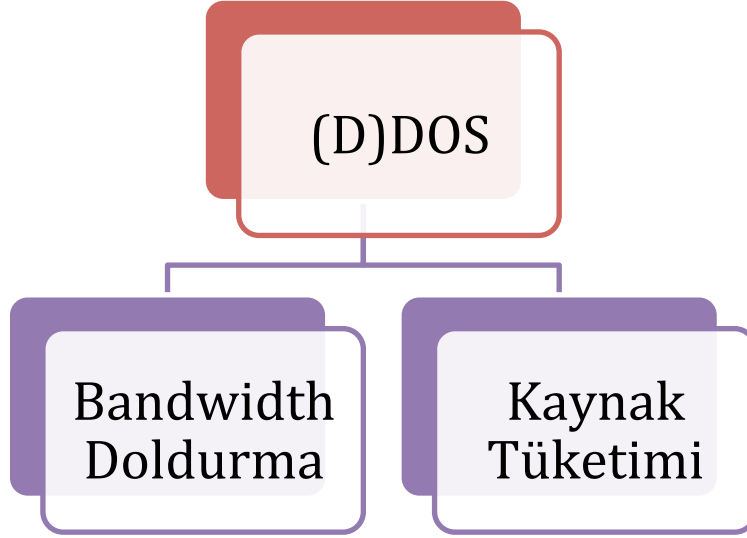
Neden Kaynaklanır?



(D)DOS Çeşitleri

Sonuçlarına göre (D)DOS Çeşitleri

DOS saldırılarında temel amaç hedefi işlevsiz kılmaktır, bunun için yapılacak ili şey vardır: hedef sistemin sahip olduğu bandwidth miktarından daha fazla trafik göndermek ya da hedef sistemin kaynaklarını diğer kullanıcıların kullanamayacağı şekilde sömürmek.



Her ne kadar DOS denildiğinde çoğu kişinin aklına ilk yöntem gelse de aslında asıl tehlikeli olan DDOS çeşidi ikinci tip olan kaynak tüketimidir.

İlk yöntemi gerçekleştirebilmek için saldırganın organize çalışması ve sağlam bir botnet'e sahip olması gerekir. Oysa ikinci yöntem olan kaynak tüketimi saldırılarında saldırgan hedef sistemin bandwidth 'in 1/10'na sahip olduğunda genellikle başarılı bir DOS saldırısı gerçekleştirebilir. Özellikle günümüz network cihazlarının DOS'a karşı sağlıklı bir koruma sağlayamamasından kaynaklanan bu sorun gelecekte de en fazla baş ağrıtan konulardan olmaya devam edecektir.

Yapılış şekline göre (D)DOS Çeşitleri

- Synflood
- Udpflood
- Ack flood
- HTTP GET flood
- DNS flood
- Teardrop
- Ping of death

Synflood (D)DOS Saldırıları

1 SYN paketi 60 byte, 50Mb bağlantısı olan biri saniyede teorik olarak 1.000.000 kadar paket gönderebilir. Bu değer günümüzde kullanılan çoğu güvenlik cihazının kapasitesinden yüksektir.

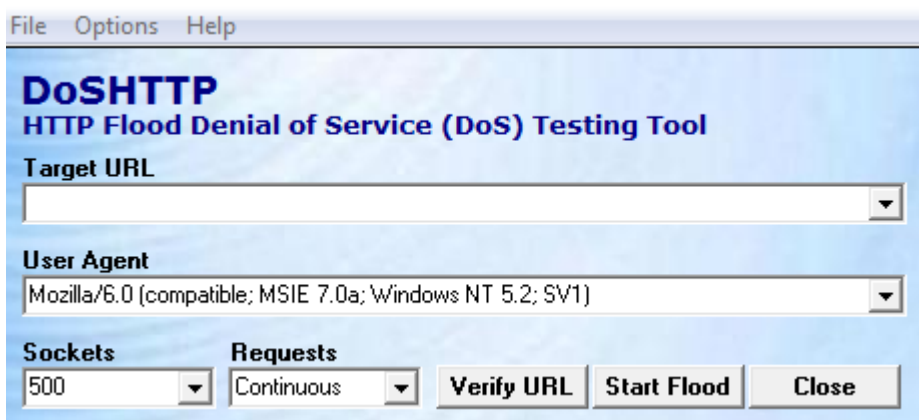
Günümüzde en sık karşılaşılan ve en etkili DOS/DDOS yöntemi SYNflood saldırıdır.

Synflood yapıldığını nasıl anlarsınız?

Netstat -an -p tcp komutunu çalıştırdığınızda fazla sayıda SYN_RECEIVED satırı görüyorsanız muhtemelen bir Synflood saldırı yapıyor demektir.

```
[root@mail ~]# netstat -an -p tcp|grep SYN_RECV|more
tcp4      0      0 10.10.10.2.80      10.10.10.4.38399    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38397    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38395    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38393    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38391    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38389    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38387    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38385    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38383    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38381    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38379    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38377    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38375    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38373    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38371    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38369    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38367    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38365    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38363    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38361    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38359    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38357    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38355    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38353    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38351    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38349    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38347    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38345    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38343    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38341    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38143    SYN_RECV
tcp4      0      0 10.10.10.2.80      10.10.10.4.38141    SYN_RECV
```

HTTP Get Flood Saldırıları



İnternet'i durdurma(DNS DOS)Saldırıları

İnternet'in çalışması için gerekli temel protokollerden biri DNS (isim çözme) protokolüdür. DNS 'in çalışmadığı bir internet, levhaları ve yönlendirmeleri olmayan bir yol gibidir. Yolu daha önceden bilmiyorsanız hedefinize ulaşmanız çok zordur.

DNS protokolü ve dns sunucu yazılımlarında geçtiğimiz yıllarda çeşitli güvenlik açıklıkları çıktı. Bu açıklıkların bazıları doğrudan dns sunucu yazılımını çalışamaz hale getirme amaçlı DOS açıklıklarıdır. Özellikle internette en fazla kullanılan DNS sunucu yazılımı olan Bind'in bu açıdan geçmişi pek parlak değildir.

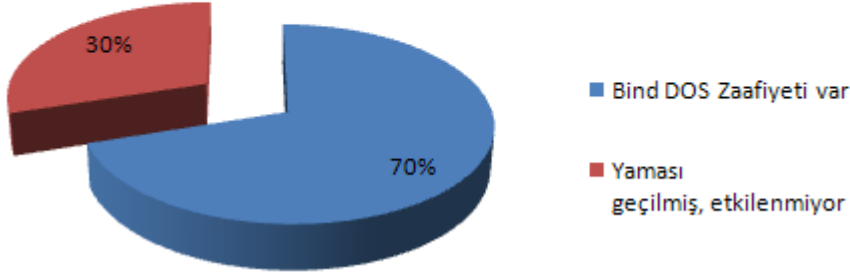
DNS sunucular eğer dikkatli yapılandırılmadıysa gönderilecek rastgele milyonlarca dns isteğiyle zor durumda bırakılabilir. Bunun için internette çeşitli araçlar mevcuttur.

DNS sunucunun loglama özelliği, eş zamanlı alabileceği dns istek sayısı, gereksiz rekursif sorgulamalara açık olması, gereksiz özelliklerinin açık olması (edns vs) vs hep DOS'a karşı sistemleri zor durumda bırakan sebeplerdir.

DNS sunucularda çıkan DOS etkili zafiyetlere en etkili örnek olarak 2009 yılı Bind açıklığı gösterilebilir. Hatırlayacak olursak 2009 yılında Bind DNS yazılımında çıkan açıklık tek bir paketle Bind DNS çalıştıran sunucuların çalışmasını durdurabiliyor. DNS paketleri udp tabanlı olduğu için kaynak ip adresi de rahatlıkla gizlenebilir ve saldırganın belirlenmesi imkânsız hale gelir.

Türkiye'de yaptığımız araştırmada sunucuların %70'nin bu açıklığa karşı korumasız durumda olduğu ortaya çıkmıştır. Kötü bir senaryo ile ciddi bir saldırgan Türkiye internet trafiğini beş dakika gibi kısa bir sürede büyük oranda işlevsiz kılabilir. Siber güvenlik üzerine çalışan ciddi bir kurumun eksikliği bu tip olaylarda daha net ortaya çıkmaktadır.

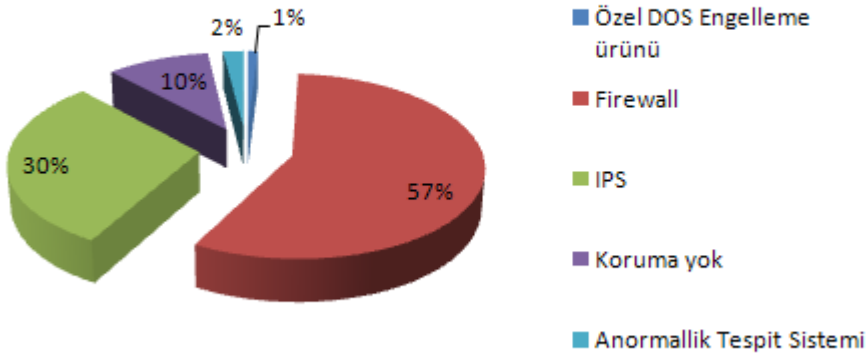
Türkiye Bind DOS Zaafiyeti Durumu



Türkiye’de DOS Saldırıları Açısından Güncel Durum

Türkiye çapında 100’e yakın büyük ölçekli firma üzerinde yaptığımız araştırma sonucu aşağıdaki grafik çıkmıştır. Firmaların büyük bir bölümü DOS/DDOS konusunda kullandıkları güvenlik duvarı ve IPS ürününe güvenmektedirler.

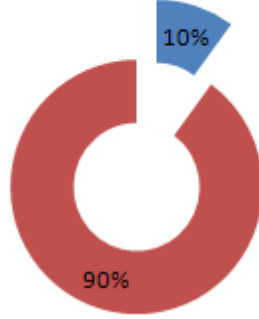
Türkiye’de DDOS Önlemleri



Bu firmalardan seçilen 10 tanesine yapılan DOS deneme testlerinde 9 tanesinin ortalama bir DOS saldırısına karşı sistemlerinin (Firewall, IPS) dayanıksız olduğu belirlenmiştir. Yapılan testler ve yöntemleri hakkında detay bilgi için yazar ile iletişime geçebilirsiniz.

DDOS Dayanıklılık Testleri

■ DDOS'a dayanıklı ■ DDOS'a karşı korumasız



Korunma Yolları ve Yöntemleri

DOS saldırılarından korunmanın sihirbazvari bir yolu yoktur. Korunmanın en sağlam yöntemi korumaya çalıştığınız network yapısının iyi tasarlanması, iyi bilinmesi ve bu konuyla görevli çalışanların TCP/IP bilgisinin iyi derecede olmasıdır. Çoğu DOS saldırısı yukarıda sayılan bu maddelerin eksikliği sebebiyle başarılı olur.

Router (Yönlendirici) Seviyesinde Koruma

Sınır koruma düzeninde ilk eleman genellikle router'dır. Sisteminize gelen-giden tüm paketler öncelikle router'dan geçer ve arkadaki sistemlere iletilir. Dolayısıyla saldırı anında ilk etkilenecek sistemler router'lar olur.

Kullanılan router üzerinde yapılacak bazı ayalar bilinen DOS saldırılarını engellemede, ya da en azından saldırının şiddetini düşürmede yardımcı olacaktır. Yine saldırı anında eğer gönderilen paketlere ait karakteristik bir özellik belirlenebilirse router üzerinden yazılacak ACL (Erişim Kontrol Listesi) ile saldırılar kolaylıkla engellenebilir.

Mesela saldırganın SYN flood yaptığını ve gönderdiği paketlerde src.port numarasının 1024 olduğunu düşünelim (Türkiye'de yapılan dos saldırılarının çoğunluğu sabit port numarasıyla yapılır). Router üzerinde kaynak port numarası 1024 olan paketleri engellersek saldırıdan en az kayıpla kurtulmuş oluruz. Bu arada kaynak portunu 1024 olarak seçen ama saldırı yapmayan kullanıcılardan gelen trafiklerde ilk aşamada bloklanacak ama normal kullanıcılardaki TCP/IP stacki hemen port numarasını değiştirerek aynı isteği tekrarlayacaktır.

Tabi bu engelleme yöntemi her saldırı için geçerli olmayabilir.

Yine routerlar üzerinden alınacak Netflow bilgisiyle saldırının şiddeti, karakteristiği, ne kadar sürdüğü ve nerelerden geldiği bilgileri kayıt altına alınabilir. Dış routerlarda eğer cihaz performans problemine sebep vermeyecek şekilde Netflow alımını sağlıyorsa bu özellik mutlaka kullanılmalıdır. Fakat bazı sistemler düzgün yapılandırılmadığından netflow sunucuya paket göndermeye çalışırken performans problemine sebep olabilirler.

Güvenlik Duvarı Seviyesinde Koruma

Güvenlik duvarlarında alınabilecek önlemlerden ilki –eğer cihaz destekliyorsa- rate limiting özelliğini aktif etmektir. Rate limiting özelliğiyle belirli bir ip adresinden gelecek maksimum paket sayısı belirlenip eşik değerini aşan ip adresleri belirli süreliğine bloklanabilir. Böylece saldırı yapan sistemler ve normal sistemler ayırt edilebilir.

Bu özellik dikkatli kullanılmazsa akıllı bir saldırgan tüm internet bağlantınızı bloklatabilir.

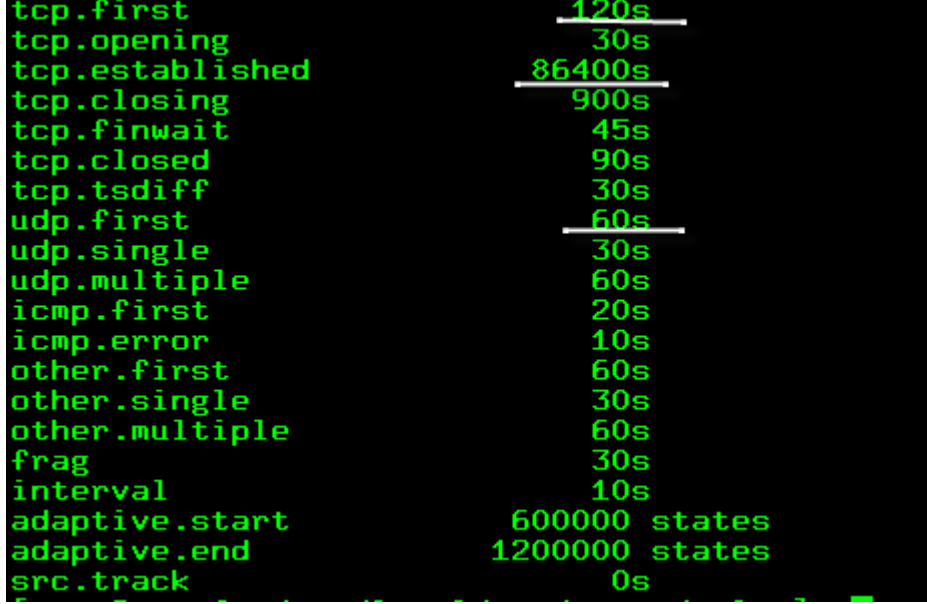
Bunun haricinde güvenlik duvarlarında kurulumla birlikte gelen ön tanımlı bazı ayarlar değiştirilmelidir. Bu ayarlar Firewall'dan gelip-geçen paketler için ne kadarlık bir süre kaynak ayrımı yapılacağını belirtir.

Güvenlik duvarı üzerinde ön tanımlı ayarların değiştirilmesi

Güvenlik duvarı kısaca koruma altına aldığı sistemlere gelen paketleri karşılayan ve üzerinde yazılı politikaya göre paketlerin geçişine izin veren sistemlerdir. Günümüz güvenlik duvarları durum koruma (stateful) özelliğine sahiptir. Böylece her gelen paket için tüm güvenlik duvarı kuralları tekrar tekrar incelenmez, eğer gelen-giden paket daha önceki bir bağlantıya ait ise doğrudan geçirilir.

Bunu sağlayabilmek için güvenlik duvarları üzerinden gelip geçen her bir paket için sistemde kaynak ayırır. (Paketin cevabını ne kadarlık süre bekleneceği vs). Ayrılan bu kaynaklar DDOS saldırısı esnasında çabucak tükenir. DDOS saldırılarına karşı daha sağlam bir güvenlik duvarı için gelip-giden paketler için tutulan zaman aşımı süreleri kısaltılabilir.

TCP ve UDP paketleriyle ilgili oturum bilgilerinin varsayılan değerlerinin değiştirilmesi



```
tcp.first          120s
tcp.opening        30s
tcp.established    86400s
tcp.closing        900s
tcp.finwait        45s
tcp.closed         90s
tcp.tsdiff         30s
udp.first          60s
udp.single          30s
udp.multiple        60s
icmp.first         20s
icmp.error         10s
other.first        60s
other.single       30s
other.multiple     60s
frag               30s
interval           10s
adaptive.start     600000 states
adaptive.end       1200000 states
src.track           0s
```

Yukardaki değerler Packet Filter güvenlik duvarından alınmıştır(pfctl -s timeouts), TCP bağlantıları için başlangıç paketi SYN alındıktan sonra ACK paketinin gelmesi için bağlantı 120 sn açık bırakılmaktadır. Bu değerler günümüz internet dünyası için fazla gelmektedir. Bu değerlerin 10'da biri bile normal işleyen bir ağda yeterlidir. Saldırı esnasında bu değerlerin düşürülmesi saldırının etkisini önemli oranda azaltacaktır.

Güvenlik duvarı syncookie, synproxy özelliklerinden birine sahipse bu özelliğin aktif edilmesi Synflood saldırılarına karşı en ciddi korumayı sağlayacaktır. Syncookie, syncache ve synproxy özellikleri synflood saldırılarında oturum kurulmamış TCP paketlerinin sunucuya ulaşmasını engelleyip DDOS'dan korumuş olur. Gelen saldırının şiddetine göre syncookie koruması yapan güvenlik duvarı da devre dışı kalabilir.

Eğer sisteminiz destekliyorsa syncookie yerine synproxy özelliği daha sağlıklı bir koruma sağlayacaktır.

Saldırı Tespit ve Engelleme (NIPS) Seviyesinde Koruma

IPS'ler bilinen DOS/DDOS saldırılarına yönelik çeşitli saldırı imzalarını veri tabanlarında barındırırlar. Her ne kadar bu saldırı tipleri çok klasik olsa da günümüzde denenmektedir. IPS'ler üzerinde ilk yapılacak işlem DOS/DDOS saldırılarına karşı önlem olabilecek imzaların devreye alınmasıdır.

Basit bir SYNflood önlemi

Aşağıdaki saldırı imzası shaft aracı kullanılarak yapılan DDOS saldırılarını belirler. Saldırı Tespit/Engelleme sistemlerinin bu tip bilinen araçlar için çeşitli imzaları bulunmaktadır.

```
alert tcp $HOME_NET any <> $EXTERNAL_NET any (msg:"DDOS shaft synflood"; flow:stateless; flags:S,12; seq:674711609; metadata:policy security-ips drop; reference:arachnids,253; reference:cve,2000-0138; classtype:attempted-dos; sid:241; rev:11;)
```

Alternatif bir kural (Generic Syn Flood Atağı)

```
alert tcp any any -> $WEB_SUNUCU 80 (msg:"Syn Flood Saldırısı"; flow:stateless; flags:S,12; threshold: type threshold, track by_src, count 100, seconds 1; classtype:attempted-recon; sid:10009;rev2;)
```

Ek olarak eğer destekliyorsa IPS üzerinde syncookie özelliği devreye alınmalıdır ve firewall'dakine benzer şekilde stateful bağlantılarda zamanaşımı sürelerinin iyi ayarlanması saldırıların etkisini azaltacaktır.

Web Sunuculara Yönelik Koruma

Web sunucular şirketlerin dışa bakan yüzü olduğu için genellikle saldırıyı alan sistemlerdir. Web sunuculara yönelik çeşitli saldırılar yapılabilir fakat en etkili saldırı tipleri GET flood saldırılarıdır. Bu saldırı yönteminde saldırgan web sunucunun kapasitesini zorlayarak normal kullanıcıların siteye erişip işlem yapmasını engeller. Bu tip durumlarda güvenlik duvarlarında uygulanan rate limiting özelliği ya da web sunucular önüne koyulacak güçlü yük dengeleyici/dağıtıcı(load balancer)cihazlar ve ters proxy sistemleri oldukça iyi koruma sağlayacaktır.

Güvenlik duvarı kullanarak http GET isteklerine limit koyulamaz. Zira http keepalive özelliği sayesinde tek bir TCP bağlantısı içerisinde yüzlerce http GET komutu gönderebilir. Burada paket içeriğine bakabilecek güvenlik duvarı/ips sistemleri kullanılmalıdır. Mesela Snort saldırı tespit/engelleme sistemi kullanılarak aşağıdaki kuralla 3 saniyede 50'den fazla http GET isteği gönderen ip adresleri bloklanabilmektedir.


```
Drop tcp any any -> $WEB_SUNUCU 80 (msg:"HTTP GET Flood Attack Attempt"; flow:established,to_server; content:"GET /"; detection_filter: track by_src, count 50, seconds 3; sid:1000001; rev:1;)
```

İşletim sistemleri üzerinde basit koruma ayarları

Synflood saldırılarına karşı koruma

- Syncookie

Linux sistemlerde syncookie özelliğinin aktif hale getirilmesi için

```
/etc/sysctl.conf dosyasına net.ipv4.tcp_syncookies = 1
```

eklenmeli ve sysctl –p komutu çalıştırılmalı ya da geçici olarak,

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Komutu kullanılmalıdır.

Windows için aynı özelliği devreye alacak çeşitli registry ayarları mevcuttur.

Ağınızdan DOS/DDOS yapılmasını Engelleme

Yapılan araştırmaya göre Fortune 500 firmasının %30'u çeşitli botnetlere üye gözüküyor. Kendi sistemlerinizden DOS yapılmasını engellemek için ilk olarak güvenlik duvarlarından ip soofing mekanizmasını engellemek gerekir. Yamalarının tam olması ve ortamda bir anormallik tespit sisteminin çalışması size ağınız hakkında bilgi verecektir.

SynFlood DDOS Saldırıları

SYN Flood Nedir?

Synflood servis engelleme saldırılarından (DOS) en bilineni ve sık karşılaşılanıdır.

Amaç: hedef sisteme kapasitesinden fazla SYN bayraklı TCP paket gönderip sistem kaynaklarını hizmet veremez hale getirmektir. Günümüzde genellikle WEB sunuculara yönelik yapılmakta ve sonuç olarak web sayfalarının çalışamaz hale gelmektedir.

Synflood saldırıları ilk olarak 1994 yılında teorik olarak “*Firewalls And Internet Security*” kitabında bahsi geçmiş ve 1996 yılında Phrack dergisinde exploit aracının çıkmasıyla yaygınlaşmaya başlamıştır.

Synflood saldırılarını anlayabilmek ve gerekli önlemleri alabilmek için TCP/IP ailesinin en sık kullanılan bileşeni TCP (RFC 793)'nin yapısı ve çalışma mantığının iyi bilinmesi şarttır. Synflood saldırısını gerçekleştirenin bilmesi gereken ise sadece hedef ip hedef port ve kullanacağı ddos programının ismidir.

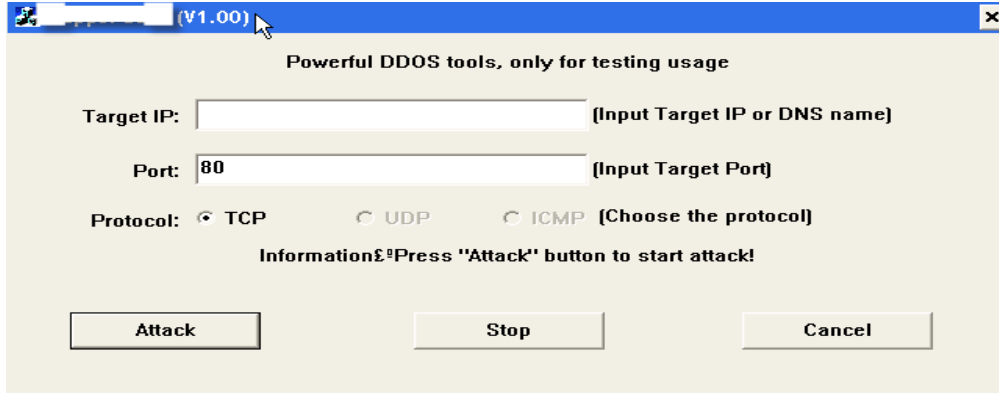
SYN Flood Saldırılarının Temel Kaynağı

SynFlood saldırısının temel sebebi bağlanan istemcilerin herhangi bir doğrulama (bağlantı yapanın gerçekten ilgili IP adresine sahip olduğu bilgisi) mekanizmasından geçmemesidir. Bu aslında Ipv4 'e ait bir eksiklik olmakla birlikte Ipv6'a geçiş yapıldığında kısmen ortadan kalkacaktır.

Syn Flood Saldırı Tipini Gerçekleştirmek

İnternette arama yapabilen her bilgisayar kullanıcı bu saldırı tipini gerçekleştirebilir. Hem Windows hem de Linux/UNIX sistemler için onlarca ddos aracı bulunmaktadır.

Saldırının etkili olabilmesi için sahip olunması gereken trafik miktarı belli bir değerin üzerinde olmalıdır (>50Mbps gibi). Saldırımı kimin yaptığını bulmak teoride mümkün gibi gözükse de günümüz internet dünyasının yapısı düşünüldüğünde spoof edilmiş ip adresleri üzerinden yapılan SynFlood saldırısının gerçek kaynağını bulmak imkânsızdır.



Syn Flood saldırılarının tam olarak anlaşılabilmesi için detay TCP/IP bilgisine ihtiyaç duyulacaktır. Kitap içinde temel TCP/IP bilgisine yer verilmiştir. Detay TCP/IP bilgisi için Richard Stevens'in TCP/IP Volume-I kitabı incelenebilir.

Temel TCP bilgisi

OSI katmanına göre 4. Katta yer alan TCP günümüz internet dünyasında en sık kullanılan protokoldür. Aynı katta yer alan komşu protokol UDP'e göre oldukça karışık bir yapıya sahiptir.

HTTP, SMTP, POP3, HTTPS gibi protokoller altyapı olarak TCP kullanırlar.

TCP Bağlantılarında Bayrak Kavramı (TCP Flags)

TCP bağlantıları bayraklarla (flags) yürütülür. Bayraklar TCP bağlantılarında durum belirleme konumuna sahiptir. Yani bağlantının başlaması, veri transferi, onay mekanizması ve bağlantının sonlandırılması işlemleri tamamen bayraklar aracılığı ile gerçekleşir (SYN, ACK, FIN, PUSH, RST, URG bayrak çeşitleridir).

UDP'de ise böyle bir mekanizma yoktur. UDP'de güvenilirliğin (paketlerin onay mekanizması) sağlanması üst katmanlarda çalışan uygulamalar yazılarak halledilebilir. DNS protokolü UDP aracılığı ile nasıl güvenilir iletişim kurulacağı konusunda detay bilgi verecektir.

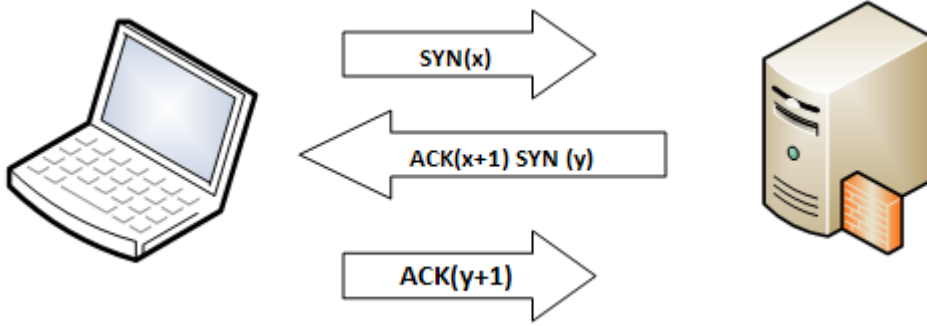
UNIX/Windows sistemlerde bağlantılara ait en detaylı bilgi netstat (Network statistics) komutu ile elde edilir. Netstat kullanarak TCP, UDP hatta UNIX domain socketlere ait tüm bilgileri edinebiliriz.

TCP’de bağlantıya ait oldukça fazla durum vardır. TCP bağlantılarında netstat aracılığı ile görülebilecek çeşitli durumlar:

CLOSE_WAIT, CLOSED, ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, LAST_ACK, LISTEN, SYN_RECEIVED, SYN_SEND ve TIME_WAIT

TCP Oturum başlatma

Web sayfalarını gezmek için kullanılan http üzerinden örnek vermek gerekirse bir web sayfasına ulaşp içeriğini görebilmemiz için öncelikle TCP oturumunun kurulması (3 lü el sıkışma) gerekir. Bu adım tamamlandıktan sonra web sayfasının içeriğini görüntüleyecek komutlar sisteme gönderilir.



Hizmet veren bir TCP portu açıksa kendisine gelen SYN paketine karşılık olarak ACK+SYN paketi döner. Dönen paketlerden ACK (onay paketi), SYN ise hizmet veren tarafın istek başlatma paketidir.

Port kapalıysa RST döner, SYNflood saldırısının başarılı olabilmesi için portun açık ve dinlemede (LISTEN mod) olması gerekir.

Oturum başlangıcında portun alabileceği durumlar

SYN_SEND: Hedef sistemle TCP bağlantısı oluşturma adımının ilkidir. Kısaca SYN bayraklı paket gönderilip buna karşılık cevap bekleme zamanında portun alacağı durum.

SYN_RECEIVED: Hedef sistem portu bağlantı kurulması için gerekli ilk adım olan SYN paketini almıştır ve cevap olarak SYN+ACK dönmüştür, karşı taraftan son ACK paketi gelene kadar bu pozisyonda bekler.

ESTABLISHED: son ACK paketi de gelmiş ve 3 lü el sıkışma tamamlanmış artık taraflar veri transferi yapabilir durumdadır.

LISTEN: O portun bağlantı kabul eder olduğunu belirtir.

Gelen bir SYN paketine kaç kere SYN+ACK döner ve dönen her cevap kaç byte'dir?

Ortalama bir TCP başlığı 60 Byte, buna dönen SYN+ACK cevabı da 60 Byte civarı olacaktır ve yapılandırılışına göre SYN paketini alan sistem son ACK paketini alana kadar 5-6 kere SYN+ACK paketini tekrar gönderir ki bu da ortalama 3 dakika tutar.

```
[root@hackme ~]# tcpdump -i eth0 -tttnn host 99.99.99.109
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
000000 IP 99.99.99.109.2913 > 91.93.119.77.80: S 928182270:928182270(0) win 512
000324 IP 91.93.119.77.80 > 99.99.99.109.2913: S 1661329962:1661329962(0) ack 928182271 win 5840 <mss 1460>
3. 398987 IP 91.93.119.77.80 > 99.99.99.109.2913: S 1661329962:1661329962(0) ack 928182271 win 5840 <mss 1460>
5. 999623 IP 91.93.119.77.80 > 99.99.99.109.2913: S 1661329962:1661329962(0) ack 928182271 win 5840 <mss 1460>
12. 202178 IP 91.93.119.77.80 > 99.99.99.109.2913: S 1661329962:1661329962(0) ack 928182271 win 5840 <mss 1460>
24. 000425 IP 91.93.119.77.80 > 99.99.99.109.2913: S 1661329962:1661329962(0) ack 928182271 win 5840 <mss 1460>
48. 205614 IP 91.93.119.77.80 > 99.99.99.109.2913: S 1661329962:1661329962(0) ack 928182271 win 5840 <mss 1460>
```

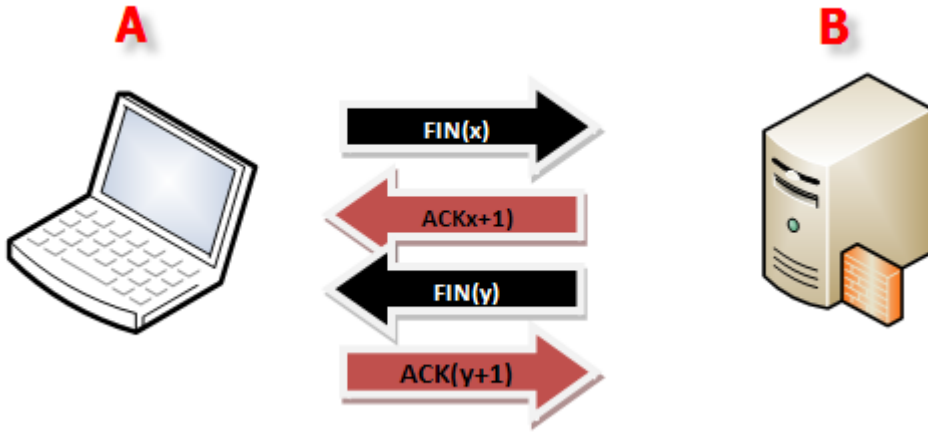
LISTEN durumunda olan bir TCP portuna SYN paketi geldiğinde SYN_RECEIVED durumuna geçer ve son ACK paketi gelene kadar verileri bir veri yapısında tutar. Bu veri yapısı TCB (Transmission Control Block) olarak adlandırılır. İşletim sistemlerine göre TCB değeri değişkenlik gösterse de ortalama olarak sistemden 280-1300 byte arası bellek kullanır (Üçlü el sıkışma tamamlanana kadar bir SYN paketininin sistemden kullandığı bellek miktarı)

İşletim sistemleri gelecek fazla isteklerden kaynaklanacak bellek yetmezliğini önleme amaçlı TCB'lerin tutacağı max bellek miktarını sınırlandırır.

TCP Oturum Sonlandırma

Oturum sonlandırma her iki tarafında anlaşması sonucu tamamlanır. Taraflardan birinin ilgili bayraklı paketi göndermemesi, geç göndermesi bağlantının sağlıklı olarak sonlanmasına engel olur.

Bağlantı sonlandırma aşamalarında çeşitli durumlar oluşur. Bu durumlara geçmeden bir TCP bağlantısının nasıl kapatıldığını inceleyelim.



Görüleceği üzere A ve B sistemleri arasındaki bağlantıyı kapatmak için 4 paket transferi oluyor. Bu paketleri Wireshark ya da tcpdump ile rahatlıkla görebilirsiniz.

tcpdump çıktısı

```

2007-08-13 21:38:57.239126 IP 80.93.212.86.3306 > 88.233.216.57.2175:
F 75:75(0) ack 1 win 65535
2007-08-13 21:38:57.292806 IP 88.233.216.57.2175 > 80.93.212.86.3306:
. ack 76 win 17446
2007-08-13 21:38:57.295927 IP 88.233.216.57.2175 > 80.93.212.86.3306:
F 1:1(0) ack 76 win 17446
2007-08-13 21:38:57.295941 IP 80.93.212.86.3306 > 88.233.216.57.2175:
. ack 2 win 65534
  
```

Bağlantı sonlanması esnasında oluşan durumlar:

FIN_WAIT_1:

Bağlantı sonlandırmak için işlem başlatan taraf(A) hedef sisteme FIN bayraklı TCP paketi gönderir. Ardından karşı taraftan(B) ayrı ayrı ACK ve FIN bayraklı paketleri bekler. Bu arada durumunu FIN_WAIT_1 olarak ayarlar.

FIN_WAIT_2:

Bağlantı sonlandırma isteğini(FIN bayraklı ilk paket) alan taraf(B) bu pakete karşılık olarak ACK(onay) bayraklı TCP paketi hazırlar ve gönderir ve durumunu CLOSE_WAIT'e alır. İlk FIN bayraklı paketi gönderen taraf(A) ACK paketini aldığı anda durumunu FIN_WAIT_2 olarak ayarlar.

Böylece bağlantı sonlandırma işleminin ilk yarısı tamamlanmıştır. Diğer yarıda sağlıklı tamamlandıktan sonra bağlantı tamamen sonlanmış olacaktır.

LAST_ACK:

B tarafı ACK bayraklı paket gönderdikten sonra, kendisinin de bağlantıyı sonlandırmak istediğini bildiren FIN bayraklı paket oluşturarak A sistemine gönderir ve durumunu LAST_ACK olarak ayarlar.

TIME_WAIT

: A sistemi FIN bayraklı paketi aldıktan sonra buna cevaben ACK bayraklı bir paket oluşturarak B'ye gönderir ve durumunu TIME_WAIT olarak belirler.

A sistemi TIME_WAIT durumunda son gönderilen ACK bayraklı paketin hedef sisteme(B) ulaştığını garantilemek için bir müddet bekler. Bu müddet eğer gereğinden fazla(eski tip UNIX sistemlerde 4 dakikaya kadar çıkabiliyor.) ise sisteminizde netstat -an çalıştırdığınızda oldukça fazla TIME_WAIT satırı görebilirsiniz.

Bu da sistemi gereğinden fazla meşgul edeceği için performans problemleri yaşanması kaçınılmaz olacaktır.

Örnek bir sistem üzerinde inceleme:

TCP/80 portuna yapılan bir istek ve isteğin sonlanması sırasında netstat ile alınan durum çıktıları. Sadece sunucu tarafını gösterdiği için bazı durumlar gözükmemektedir. İsteği yapan taraf da incelenecek olursa eksik kalan kısımlar tamamlanır.

```
tcp4 0 0 80.93.212.86.80 88.233.216.57.2348 SYN_RECEIVED
tcp4 0 0 80.93.212.86.80 88.233.216.57.2348 ESTABLISHED
tcp4 0 0 80.93.212.86.80 88.233.216.57.2348 FIN_WAIT_2
tcp4 0 0 80.93.212.86.80 88.233.216.57.2348 TIME_WAIT
```

SynFlood Saldırıları Nasıl Gerçekleştirilir?

Syn Flood saldırısı basitçe açık bir porta hedef sistemin kapasitesinden fazla gönderilecek SYN paketleriyle gerçekleştirilir. Buradaki “kapasite” tanımı önemlidir. Teknik olarak bu kapasiteye Backlog Queue denilmektedir.

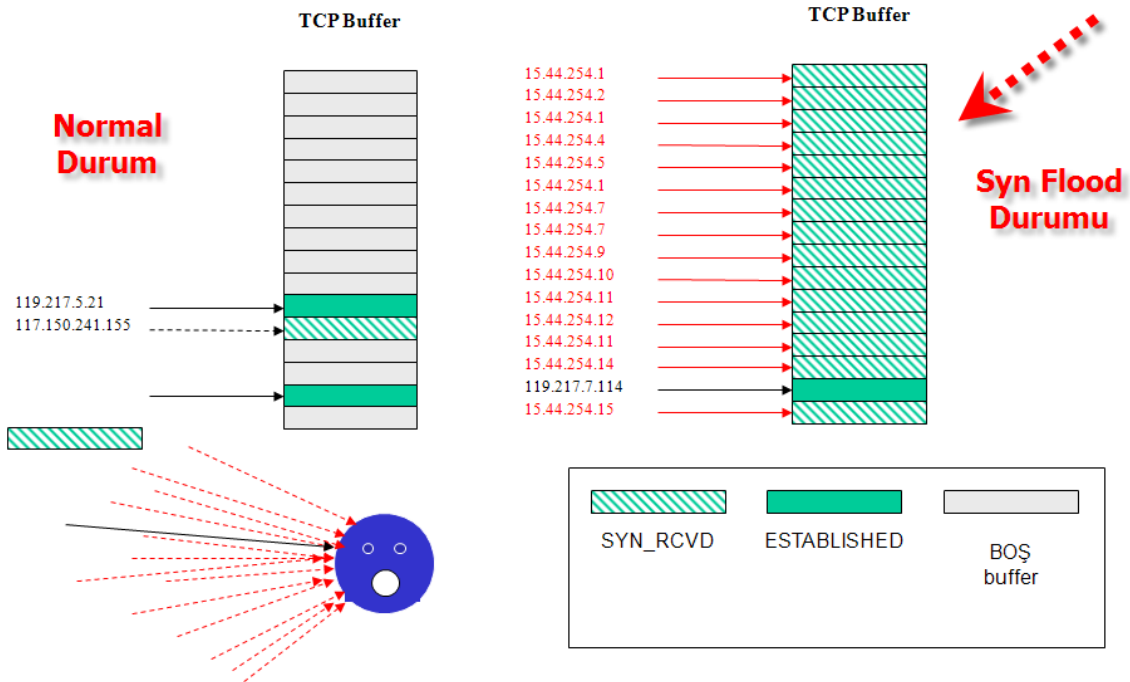
Backlog queue kavramı

İşletim sistemleri aldığı her SYN paketine karşılık üçlü el sıkışmanın tamamlanacağı ana kadar bellekten bir alan kullanırlar, bu alan TCB olarak adlandırılır ve bu alanların toplamı backlog queue olarak adlandırılır.

Başka bir ifadeyle işletim sisteminin half-open olarak ne kadar bağlantı tutabileceğini backlog queue veriyapısı belirler. Bu değer her işletim sisteminde vardır ve ön tanımlı olarak düşük bir değerdir (256 gibi). SYNflood saldırılarında bu değer artırılarak saldırıya karşı ek önlem alınabilir.

Synflood saldırılarında tüm mesele backlog queue'nin dolması ve yeni gelen bağlantıların reddedilmesidir. Backlog queue değerinin büyük olması demek daha fazla half-open(SYN paketi) bağlantı kabul edebilmek demektir.

Backlog queue dolmasıyla birlikte işletim sistemi yeni bağlantı kabul edemez ve bu esnada sunucuya bağlanmaya çalışanlar bağlanamazlar ki bu da SYN Flood saldırısına denk gelir.



Synflood matematik hesabı

Günümüzde yeterli önlem alınmamış sistemlerin Synflood'a karşı ne kadar dayanıksız olduğunu göstermek için basit bir matematik hesabı yapalım:

Backlog queue değeri 1000 olan sisteme 1000 adet SYN paketi göndererek servis veremez duruma getirilebilir.

1000 adet SYN paketi=1000*60byte=60.000 byte=468Kpbs olacaktır ki bu değer günümüzde çoğu ADSL kullanıcısının sahip olduğu hat kapasitesine yakındır.

Servis verememe durumu ilgili porta gelen paketlerin timeout değeriyle doğru orantılıdır. Her 60 saniye de bir bu kadar paket gönderilmesi durumunda sistem tamamen erişilmez olabilir.

Not: Bir porta SYN paketi geldiğinde cevap olarak ACK +SYN döneceği için iki kat malzeme harcanır. Bu durum özellikle güvenlik cihazlarındaki PPS(Saniyede işleyebileceği paket sayısı) değerlerini ölçerken göz ardı edilir. Bir cihaz 100.000 pps diyorsa aslında bu saniyede 50.000 SYN paketi kabul edebilir demektir.

SynFlood DDOS Saldırısı Nasıl Anlaşılır?

UNIX/Linux ve Windows işletim sistemlerinde netstat komutuna uygun parametre vererek SYN Flood saldırıları anlaşılabilir. Güvenlik cihazlarında ise state(durum) tablosuna bakarak ya da connection tablosuna bakarak Syn Flood saldırıları anlaşılabilir.

```
[root@hackme ~]# netstat -ant|grep SYN_RECV
tcp        0      0 0 91.93.119.77:80      87.67.208.159:2550    SYN_RECV
tcp        0      0 0 91.93.119.77:80      222.7.139.211:2553    SYN_RECV
tcp        0      0 0 91.93.119.77:80      84.126.171.37:2541    SYN_RECV
tcp        0      0 0 91.93.119.77:80      23.3.218.61:2545     SYN_RECV
tcp        0      0 0 91.93.119.77:80      61.156.167.224:2538   SYN_RECV
tcp        0      0 0 91.93.119.77:80      197.68.120.6:2533     SYN_RECV
tcp        0      0 0 91.93.119.77:80      75.217.84.238:2546    SYN_RECV
tcp        0      0 0 91.93.119.77:80      199.139.67.162:2555   SYN_RECV
tcp        0      0 0 91.93.119.77:80      108.60.179.141:2552   SYN_RECV
tcp        0      0 0 91.93.119.77:80      88.65.103.236:2540    SYN_RECV
tcp        0      0 0 91.93.119.77:80      141.6.36.83:2554      SYN_RECV
tcp        0      0 0 91.93.119.77:80      183.152.167.29:2556   SYN_RECV
tcp        0      0 0 91.93.119.77:80      51.220.145.73:2535    SYN_RECV
tcp        0      0 0 91.93.119.77:80      53.180.0.25:2542      SYN_RECV
tcp        0      0 0 91.93.119.77:80      26.155.106.211:2559   SYN_RECV
tcp        0      0 0 91.93.119.77:80      162.190.171.87:2561   SYN_RECV
tcp        0      0 0 91.93.119.77:80      112.38.199.120:2544    SYN_RECV
tcp        0      0 0 91.93.119.77:80      106.152.221.42:2537   SYN_RECV
tcp        0      0 0 91.93.119.77:80      85.172.27.202:2557    SYN_RECV
tcp        0      0 0 91.93.119.77:80      194.145.35.130:2548   SYN_RECV
tcp        0      0 0 91.93.119.77:80      198.55.236.250:2560   SYN_RECV
tcp        0      0 0 91.93.119.77:80      113.248.199.145:2549   SYN_RECV
tcp        0      0 0 91.93.119.77:80      98.87.167.160:2534    SYN_RECV
tcp        0      0 0 91.93.119.77:80      8.248.218.198:2563     SYN_RECV
tcp        0      0 0 91.93.119.77:80      85.127.141.187:2564    SYN_RECV
tcp        0      0 0 91.93.119.77:80      153.112.143.250:2547   SYN_RECV
tcp        0      0 0 91.93.119.77:80      9.112.8.154:2532      SYN_RECV
tcp        0      0 0 91.93.119.77:80      4.164.35.136:2543     SYN_RECV
tcp        0      0 0 91.93.119.77:80      204.141.25.61:2536    SYN_RECV
tcp        0      0 0 91.93.119.77:80      179.204.10.160:2531   SYN_RECV
```

Ya da basit olarak aşağıdaki komutu çalıştırılır

```
# netstat -ant|grep SYN_RECV|grep ":80"|wc -l
```

1002

Bu komutun çıktısı belli bir değer üzerindeyse sisteme SYN Flood saldırısı yapılıyor olabilir. (Yukardaki komut sadece 80. Porta gelen syn paketlerini saydırmak için kullanılır, tüm sisteme gelen syn paketlerini görmek için grep “:80” kolonu kaldırılabilir.

Ek olarak Linux altında SYN Flood’a uğrayan bir sistem messages dosyasına aşağıdaki gibi bir log atar.

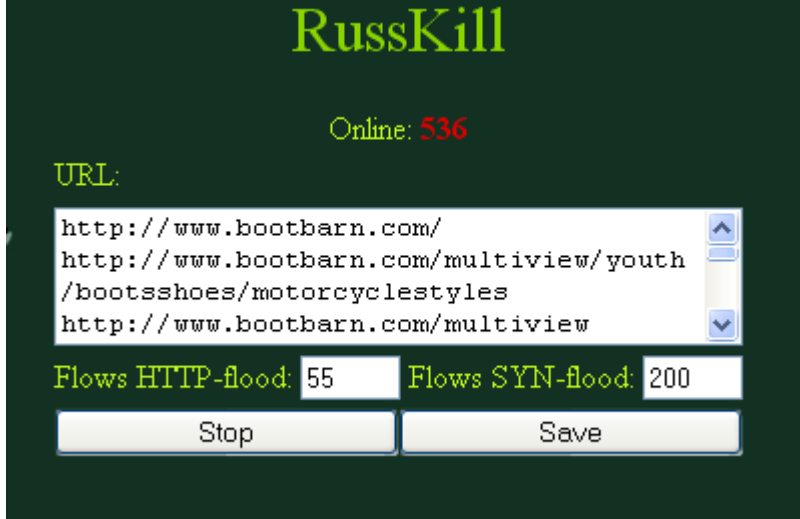
possible SYN flooding on port 80. Sending cookies.

Syn Flood Saldırı Testleri

Syn Flood saldırıları sizi bulmadan önce kendi sistemlerinizi bu tip saldırılara karşı yapılandırmanız ve test yapmanız bir saldırı karşısında hangi durumlara düşebileceğiniz konusunda fikir verecektir. Aşağıda bazı araç isimleri ve

kullanımları verilmiştir, bunlar tamamen kendi sistemlerinizi test etme amaçlı kullanılmalıdır.

İnternet üzerinde bu tip atakları yapanlar daha basit ve etkili araçlar kullanmaktadır. (bkz: botnet yönetim konsolları)



Scapy ile basit Synflood aracı

Scapy, istenilen türde TCP/IP paketleri üretmeye ve bu paketler üzerinde çeşitli işlemler yapmaya izin veren bir çatıdır. Python ile yazıldığından Python destekleyen sistemlerde (Tüm işletim sistemleri) çalışmaktadır.

Scapy kullanarak aşağıdaki gibi istenilen aralıkta spoofed ip üreten SYNflood saldırısı denenebilir.

```
>>>send(IP(src=RandIP('78.0.0.0/16'),dst='www.example.com')/TCP(sport=andShort(),dport=80), loop=1)
```

Hping Kullanarak SYNflood Testleri

Hping TCP/IP paket üretim aracıdır. Hping'in -flood ve rand-source özellikleri kullanılarak yüksek miktarlarda SYN Flood saldırıları simulasyonu yapılabilir.

Hping ile spoof edilmiş ip adreslerinden Syn Flood saldırısı

```
#hping --rand-source -p 80 -S www.hedefsistem1.com --flood
```

Benzeri araçlar kullanarak sahip olduğunuz trafik miktarına göre oldukça yüksek sayılarda SYN paketi üretip gönderebilirsiniz. SYN paketinin 60 byte civarında olduğunu düşünürsek 50Mb hat ile saniyede 100.000 SYN paketi üretip gönderilebilir ki bu değer birçok güvenlik duvarının sahip olduğu bağlantı limiti sayısını kısa sürede aşırır.(günümüz güvenlik duvarlarında en yüksek bağlantı limitleri 400-500k civarındadır).

Bağlantı limit sayısı 500.000 olan(eş zamanlı 500.000 bağlantı tutabilen) bir sistemi 50Mb hat ile 5 saniyede işlevsiz bırakılabilir.

Not: Burada güvenlik duvarları üzerinde syncookie, synproxy gibi özelliklerin aktif olmadığını düşünüyoruz.

SynFlood saldırılarında kaynak ip adresleri eğer internette açık olan ip adreslerinden seçilirse etkisi düşük olur. Zira açık olan bir ip adresini spoof ederek gönderilecek SYN paketine sunucu SYN+ACK cevabı dönecektir ve bu cevap spoof edilen makineye gelecektir, makine de daha önce böyle bir paket göndermediği için RST paketi gönderip bağlantının hızlıca kapanmasını sağlayacaktır.

SYNFlood Saldırısı Esnasında Yapılacaklar

İlk olarak gelen giden tüm paketleri görebilecek bir sistem üzerinde(Port mirroring, TAP cihazı vs yi dinleyen Linux/UNIX makine)n saldırıyı analiz etmek için paket kaydı yapılmalıdır. Bunun için tcpdump kullanılabilir.

```
#tcpdump -ttttnn -s0 tcp -w SYNFLOOD.pcap &
```

Tcpdump ile sadece SYN bayraklı TCP paketlerini görmek için verilecek komut:

```
tcpdump 'tcp[13] & 2 != 0' -i eth0 -nnn
```

Saldırı esnasında paketleri incelemek bazı basit saldırılarda faydalı olabilmektedir. Mesela Synflood için çok sık tercih edilen Juno aracı öntanımlı olarak kaynak port numarasını 1024/3072 yapmaktadır. Eğer saldırgan Juno.c'nin kaynak kodunu inceleyip kaynak port numaralarını random hale getirmemişse bu saldırı router üzerinden yazılacak basit bir ACL(erişim listesi) ile engellenebilir.

Sistemleriniz güvenlik duvarı, IPS vs ile korunuyorsa bu sistemlerindeki Syncookie, Synproxy özellikleri aktif edilmeli. (Eğer saldırının boyutu

yükseğe bu tip engellemelerin kısa sürede devre dışı kalacağı da bilinmelidir).

Bu gibi durumlarda alternatif yöntemler denenerek sistemlerin ulaşılabilir olması sağlanabilir. Dns kayıtlarının TTL değerlerinin düşürülerek farklı ip adresine geçilmesi gibi.

Ülkeye göre IP bloklama

SynFlood esnasında syncookie vs işe yaramadıysa portu dünyaya kapatıp sadece Türkiye IP bloklarına açabiliriz, eğer SYNFLood yapan tamamen random ip ürettiyorsa bu tip bir önlem saldırıyı %70-%90 oranında etkisiz hale getirecektir.

Peki, spoof edilmiş ip adresleri arasında Türkiye ip adresleri de olursa? Bunlar için de çözüm doğrudan Türkiye ip aralığından paket kabul etmek yerine bu ip aralığından gelecek istekleri SYNProxy'e devredip bu ip aralığından gelebilecek muhtemel saldırı paketlerini de elemiş oluruz.

Sık kullanılan SYNflood araçları kaynak ip adresini isteğe göre oluşturma özelliğine sahip değildir, genelde kaynak ip adresi random üretilir. Fakat bu yöntem bu konuda tecrübeli saldırganlar için sadece bir engeldir, ilgili programın kaynak kodunda yapılacak basit değişikliklerle isteğe göre kaynak ip adresi üretilebilir. (78.*.* bloğundan random ip üret gibi.) mz aracı bu amaç için kullanılabilir.

Türkiye IP Blokları Nereden Edinilebilir

İnternet üzerinde ülkelerle ait ip bloklarını ücretsiz dağıtan çeşitli servisler vardır. Google üzerinden yapılacak “country ip blocks” araması yeterli sonuç verecektir. Buradan alınacak ip blokları ciddi saldırılar esnasında işe yarayacaktır.

Synflood Önleme Yöntem ve Çeşitleri

SynFlood saldırılarına karşı çeşitli önlemler geliştirilmiştir. Bunlar arasında günümüzde teoriden pratiğe geçiş yapabilmiş üç temel yöntem bulunmaktadır.

- Syncookie
- syncache(FreeBSD default)
- SynProxy
- TCP Authentication

Syncookie Nasıl çalışır?

Normal TCP bağlantılarında gelen SYN bayraklı pakete karşılık ACK paketi ve SYN paketi gönderilir. Gönderilen ikinci(sunucunun gönderdiği) SYN paketinde ISN değeri random olarak atanır ve son gelecek ACK paketindeki sıra numarasının bizim gönderdiğimizden bir fazla olması beklenir, son paket gelene kadar da sistemden bu bağlantı için bir kaynak ayrılır(backlog queue). Eğer bizim gönderdiğimiz SYN paketine dönen ACK cevabı bizim ISN+1 değilse paket kabul edilmez.

Syncookie aktif edilmiş bir sistemde gelen SYN paketi için sistemden bir kaynak ayrılmaz, bunun aksine SYN paketine dönecek cevaptaki ISN numarası özel olarak hesaplanır (kaynak.ip+kaynak.port+.hedef.ip+hedef.port+x değeri) ve hedefe gönderilir, hedef son paket olan ACK'i gönderdiğinde ISN hesaplama işlemi tekrarlanır ve eğer ISN numarası uygunsa bağlantı kurulur, değilse bağlantı kurulmaz. Böylece spoof edilmiş binlerce ip adresinden gelen SYN paketleri için sistemde bellek tüketilmemiş olacaktır ki bu da sistemin SYNflood saldırıları esnasında daha dayanıklı olmasını sağlar.

Syncookie mekanizması backlogqueue kullanmadığı için sistem kaynaklarını daha az tüketir. Syncookie aktif iken hazırlanan özel ISN numarası cookie olarak adlandırılır.

İstemci tarafı syncookie özelliği Inverse syn cookie (Scanrand aracı) araçları kullanılarak syncookie engellemesi aşılabılır. Bu durumda da bir ip adresinden gelecek max bağlantı sayısı limitlenerek saldırı engellenmiş olur.

Syn cookie dezavantajları

Syncookie'de özel hazırlanacak ISN'ler için üretilen random değerler sistemde matematiksel işlem gücü gerektirdiği için CPU harcar ve eğer saldırının boyutu yüksekse CPU performans problemlerinden dolayı sistem yine darboğaz yaşar. DDOS Engelleme ürünleri(bazı IPS'ler de) bu darboğazı aşmak için sistemde Syncookie özelliğini farklı özel bir CPU'ya devredeler. Böylece Syncookie işlemleri için farklı, sistemin işleyişi için farklı CPU'lar kullanılır.

Syncookie özelliği sadece belirli bir sistem için açılmaz. Ya açıktır ya kapalı, bu özellik çeşitli IPS sistemlerinde probleme sebep olabilir.

Syncookie uygulamalarından bazıları TCP seçeneklerini tutmadığı için bazı bağlantılarda sorun yaşatabilir.

SynCache nasıl çalışır?

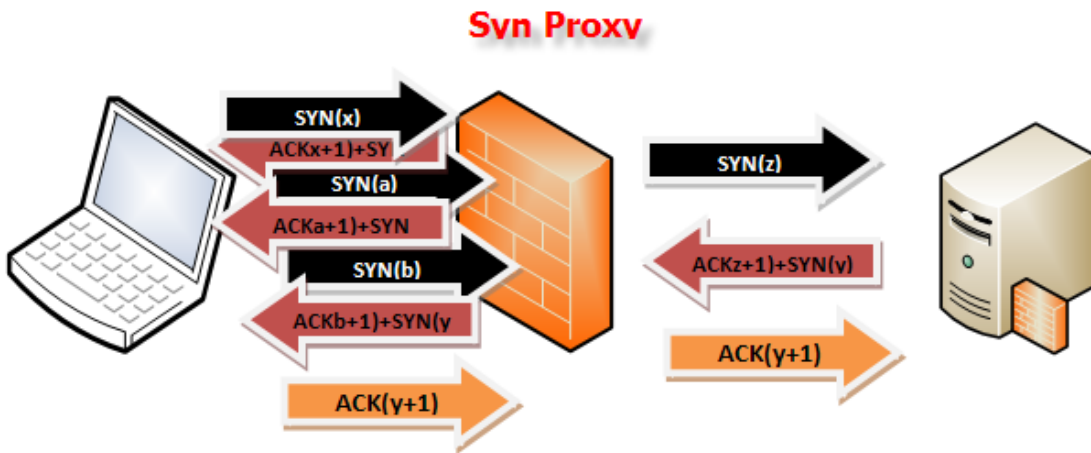
LISTEN modundaki bir portun gelen SYN paketlerinde bellekten bir alan ayırdığını ve bu alanın belirli boyutlarda olduğundan bahsetmiştik. SynCache özelliği FreeBSD sistemlerde gelen SYN paketleri için TCB değerinden daha az yer kaplayan başka bir veri yapısı kullanmayı önerir. Böylece sisteme gelen SYN paketlerinde daha az bellek alanı harcanır(Normalde 700 Byte civarı, 160 Byte Syncache kullanıldığında). Fakat yoğun bir saldırı da bu özellik kısa sürede işe yaramaz hale gelecektir.

Bu sebeptendir ki Syncache tek başına synflood saldırılarına karşı efektif bir koruma sağlamaz. Syncookie'i tetikleyici olarak kullanılır. Yani sistemde ön tanımlı olarak syncookie aktif edilmez, syncache aktif edilir. Syncache belli bir değerin üzerinde SYN paketi almaya başladığında SYNCookie'ei tetikler ve sistem koruma moduna geçer.

Syn Proxy nasıl çalışır?

SynProxy, adından da anlaşılacağı üzere SYN paketlerine karşı proxylik yapmaya yarayan bir özelliktir. Güvenlik duvarlarında ve Syncookie'nin kullanımının sıkıntılı olduğu durumlarda rahatlıkla kullanılabilir.

Syncookie gibi arkasında korumaya aldığı sistemlere gelecek tüm SYN paketlerini karşılar ve üçlü el sıkışma tamamlandıktan sonra paketleri koruduğu sistemlere yönlendirir.



Synproxy yapan sistem kendisi de SYNflood'a dayanıklı olmalıdır.

Ağınızdan SYN Flood yapılmasını Engelleme

SYNFlood saldırıları genelde iki şekilde yapılır:

-BotNet kullanarak gerçek makinelerden

-Bw'i yüksek sistemler üzerinde spoof edilmiş ip adreslerinden

Günümüz SynFlood saldırıları genellikle spoof edilmiş ip adresleri kullanılır. Diğer türlü saldırı çok efektif olmayacaktır.

Eğer firmalar, özellikle de telekom firmaları Spoof edilmiş ip adreslerinden trafik çıkışına izin vermeyecek yapılar kursa ip adres spoofing işlemi yapılamayacağı için Syn flood saldırıları büyük oranda engellenebilecektir. Günümüz internet dünyasında şirketler spoofed ipleri engellese de internetin çoğunluğunu oluşturan masaüstü kullanıcılarına hizmet veren Telekom firmaları bu özelliği kullanmamaktadır(Ya da kullanamamaktadır)

Kullandığınız network/güvenlik sistemlerinde URFP (Unicast Reverse Path Forwarding) özelliği varsa kullanarak ağınızdan dışarı çıkacak spoof edilmiş paketleri engelleyebilirsiniz.

OpenBSD PF ile URPF kullanımı?

URPF kullanılarak eğer paket uygun arabirimden gelmiyorsa paketin spoofed olduğuna karar verilir ve PF ile bu tip paketler yasaklanır. Anlasilir olması için bir örnekleyelim.

Güvenlik duvarımızda iki ağ arabirimi olsun. Biri iç ağa bakan(fxp0) ve ip adresi 192.168.0.1, diğeri de dış ağa (Modem/router vs) baksın(xl0) ve ip adresi 172.16.10.2

```
#netstat -rn -f inet
Routing tables

Internet:
Destination Gateway Flags Refs Use Mtu Interface

default 172.16.10.1 UGS 2 20010970 – xl0
192.168.0/24 link#2 UC 1 0 – fxp0
172.16.10/24 link#1 UC 2 0 – xl0
```

Normal durumlarda fxp0 arabirimine sadece 192.168.0/24 networküne ait ip adresli makinelerden paket gelmesi gerekir. Bunun harici ip adreslerinden

paket geliyorsa ya ağın içerisinde başka alt ağlar kurulmuş ya da birileri ip adreslerini spoof etmeye çalışıyor demektir.

OpenBSD PF'e urpf kuralı girerek istenen ağ arabirimi üzerinde geriye doğru yönlendirme yapılamayacak ip adreslerinden gelen isteklerin bloklanması sağlanabilir.

urpf için OpenBSD PF kuralı.

block in quick on \$int_ if from urpf-failed

Not-I: urpf kullanırken dikkatli olmak lazım, zira günümüz ağları artık alt ağlardan oluşuyor . URPF kullanmadan önce yönlendirme tablosu detaylıca incelenip karar verilmeli.

Snort Kullanarak SYN Flood saldırılarının Belirlenmesi

Snort saldırı tespit ve engelleme sistemidir. Spoof edilmiş ip adreslerinden yapılacak SYN Flood saldırıları karşısında herhangi bir engelleme özelliği olmamakla beraber, synflood saldırısının yapıldığını belirleyebilir.

```
alert tcp any any -> $WEB_SUNUCU 80 (msg:"Syn Flood Saldirisi"; flow: stateless; flags:S,12; threshold: type threshold, track by_src, count 100, seconds 1; classtype:attempted-recon; sid:10009;rev2;)
```

İşletim Sistemlerinde SYN Flood Koruması

Her işletim sistemi Synflood saldırılarına karşı benzer çözüm önermektedir.

1. Açılabilir maksimum half-open sayısının artırılması (SAYN_RECEIVED durumu)
2. Half-open bağlantılarda bekleme süresini kısaltma
3. TCP paketleri için zamanaşımı sürelerini düşürme

Backlog Queue değerini artırma

Backlog queue değeri doğrudan sistemde var olan fiziksel bellekle alakalıdır. Bu değeri arttırırken sistemdeki belleği de göz önüne almak gerekir.

Linux için backlog queue değerini arttırma

Ön tanımlı değeri:

```
# cat /proc/sys/net/ipv4/tcp_max_syn_backlog  
1024
```

Arttırmak için

```
echo 4096 > /proc/sys/net/ipv4/tcp_max_syn_backlog
```

bu değerin kalıcı olabilmesi için **/etc/sysctl.conf** dosyasına yazılması gerekir.

FreeBSD backlog queue değeri arttırma

```
# sysctl kern.ipc.somaxconn=4096  
kern.ipc.somaxconn: 128 -> 4096
```

Zaman aşımı(Timeout) değerlerini düşürme

İşletim sistemleri SYN paketi aldığı anda buna SYN+ACK cevabı döner ve ACK cevabı bekler. Eğer ACK cevabı gelmezse belirli bir süre SYN+ACK cevabını tekrarlar. Bir Synflood saldırısı esnasında kaynak ip'ler spoof edilmiş olduğundan gereksiz yere binlerce ip adresine defalarca SYN+ACK cevabı dönülecek ve kaynak tüketilecektir.

Backlog queue'de bekleme zamanını değiştirerek spoof edilmiş ip adreslerinden gelen syn paketlerinin çabucak paketlerin drop edilmesini sağlayabiliriz.

Linux altında gelen SYN paketine karşılık kaç kere ACK+SYN dönüleceği tcp_synack_retries değerinde saklanır.

```
cat /proc/sys/net/ipv4/tcp_synack_retries  
5
```

Bu değerin 5 olması demek aşağı yukarı bağlantının üç dakika asılı kalması ve backlog queue'i şişirmesi demektir. Bu değeri iki ya da üç yaparak bağlantıların çok daha kısa sürelerde düşürülmesini sağlayabiliriz.

Syncookie aktivasyonu

Linux için;

```
sysctl net.ipv4.tcp_syncookies = 1
```

FreeBSD için

```
sysctl net.ipv4.tcp_syncookies = 1
```

ayarların kalıcı olabilmesi için /etc/sysctl.conf dosyasına yukardaki formatta yazılmış olması gerekir.

Syn Flood DDOS saldırıları oldukça basit, basit olduğu kadar da etkili bir saldırı yöntemidir. Teknik olarak saldırının altyapısı ve ağ yapısı bilinirse daha kolay önlem alınabilir. DDOS konusunda akıldan çıkarılmaması gereken husus bu tip saldırıların tak çalıştır cihazlarla engellenemeyeceğidir. Alınacak DDOS koruma ürünleri mutlaka kullanılacak ağın trafik yapısına göre ayarlanmalı ve özellikleri iyi bilinmelidir. Tak ve çalıştır tipi cihazlar karışık saldırılara karşı etkisiz kalacaktır.

Web Sunuculara Yönelik DoS/DdoS Saldırıları

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.

HTTP’e Giriş

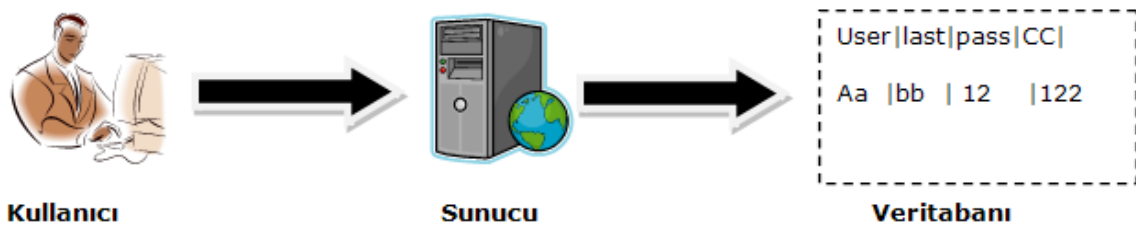
HTTP(Hypertext Transfer Protocol) OSI modelinde uygulama katmanında yer alan iletişim protokolüdür. Günümüzde zamanımızın çoğunu geçirdiğimiz sanal dünyada en sık kullanılan protokoldür.(%96 civarında)

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 veri tabanları 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 veritabanında 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

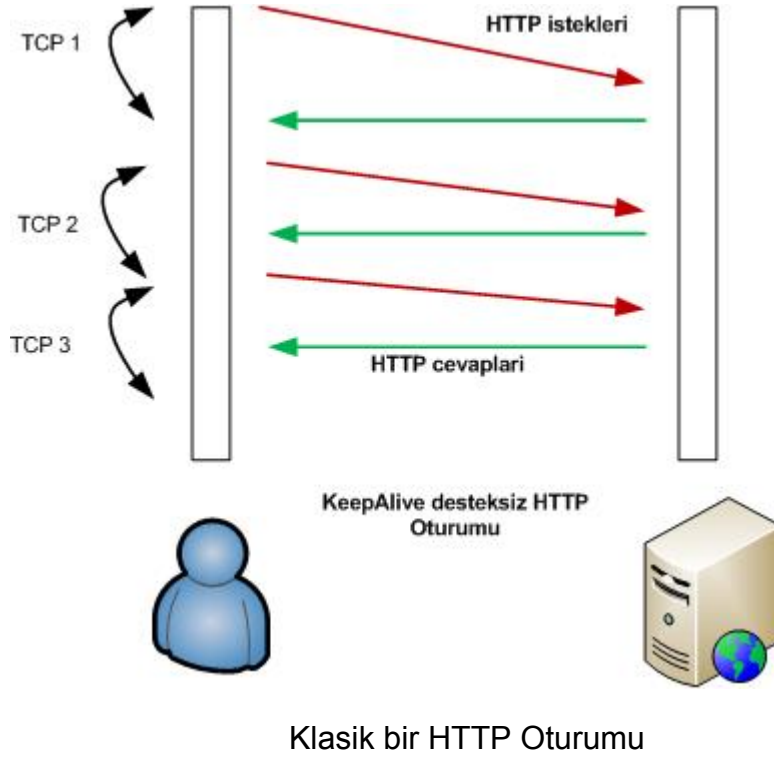
```
HTTP/1.1 200 OK
Date: Fri, 04 Dec 2009 09:09:34 GMT
Server: Apache/2.3.4 (Unix) mod_fcgid/2.3.2-dev
Content-Location: keepalive.html.en
Vary: negotiate,accept-language,accept-charset,Accept-Encoding
TCN: choice
Accept-Ranges: bytes
Content-Encoding: gzip
Content-Length: 1752
Keep-Alive: timeout=30, 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 yetkisi olmayan bir dosyayı istiyorsa 403 forbidden cevabı dönecektir.

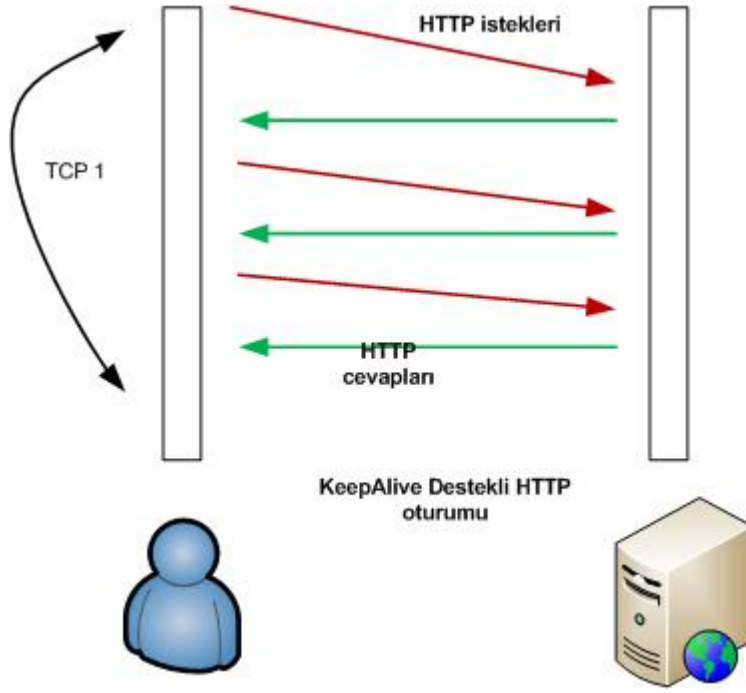
HTTP ve TCP ilişkisi

HTTP TCP kullanan bir protokoldür. Her HTTP bağlantısı öncesinde mutlaka TCP bağlantısı kurulmalıdır. Basit bir hespla 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 portalinin yüklenmesi için ortalama 40-50 HTTP GET isteği gönderilmektedir. Bunu da hesaplarsak portal sayfasının açılması için ortalama $50 \times 10 = 500$ TCP paketinin gidip gelmesi gerekir ki bu değer haber kullanıcıyı okumaktan vazgeçirecek kadar fazladır.



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 içerisinde belirli sayıda (5, 10, ..) HTTP isteğinin aktarılabilmesini sağlar.

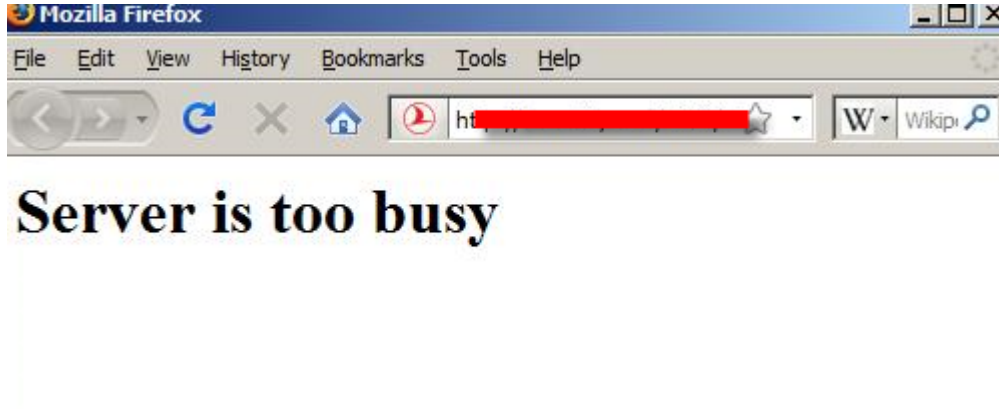


Http Pipelining

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

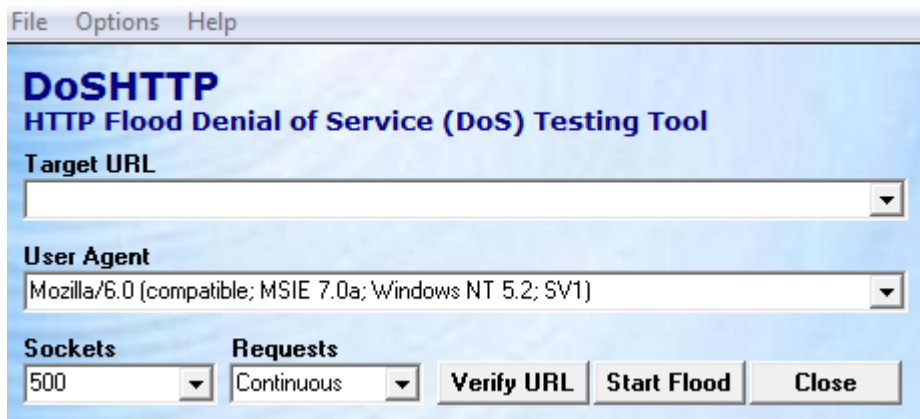
Tasarımsal zafiyetlerden kaynaklanan DOS saldırılarına en iyi örnek geçtiğimiz aylarda yayınlanan Slowloris aracıdır. Bu araçla tek bir sistem üzerinden Apache HTTP sunucu yazılımını kullanan sistemler rahatlıkla devre dışı bırakılabilir.

Benzeri şekilde Captcha kullanılmayan formlarda ciddi DOS saldırılarına yol açabilir. Mesela form üzerinden alınan bilgiler bir mail sunucu aracılığıyla 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 zafiyetler çıkmaktadır. Mesela yine geçtiğimiz günlerde yayınlanan bir PHP zafiyeti (PHP “multipart/form-data” denial of service)ni kullanılarak web sunucu rahatlıkla işlevsiz bırakılabilir. Bu tip zafiyetler 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 konumlandığımız 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

```

HTTP Flood Test Report
Date: 12.05.2009 07:01:06

Target URL:
Target Port: 80

Duration: 33 seconds
Requests Issued: 9998
Responses Received: 33
Requests Lost: 99.67%
Request Rate: 302.97 requests per second

```

Bu yazıda saldırganların kullandıkları 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/searchme.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. Yine de 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, Referer vs) IPS üzerinden özel imzalar yazılarak bu veri alanına sahip paketler engellenebilir fakat bunun normal kullanıcıları da etkileyeceği bilinmesi gereken bir konudur.
- Web sunucunun desteklediği dos koruma modülleri kullanılabilir (Apache Mod_dosevasive)

İyi 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 bloklamak 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.

DNS Hizmetine Yönelik DOS/DDDOS Saldırıları

İnternet dünyasının çalışmasını sağlayan ana protokoller incelediğinde güvenlik açısından en önemli protokollerden birinin DNS olduğu ortaya çıkmaktadır. Basitçe DNS, günümüz e-posta iletişiminin ve internet altyapısının sağlıklı çalışmasında kritik rol oynamaktadır.

Örnek olarak Türkiye'nin internet altyapısına yönelik gerçekleştirilecek en önemli saldırı ülkenin en yetkili DNS sunucularına yapılacak saldırıdır. Bu sistemler yeteri kadar korunmuyorsa internetten edinilecek çeşitli yazılımlarla DNS sunucu uzun süreler çalışamaz hale getirilebilir. Bunun sonucu olarak da Türkiye'deki tr. uzantılı sistemlere erişim ve e-posta trafiğinde ciddi aksamalar yaşanabilir.

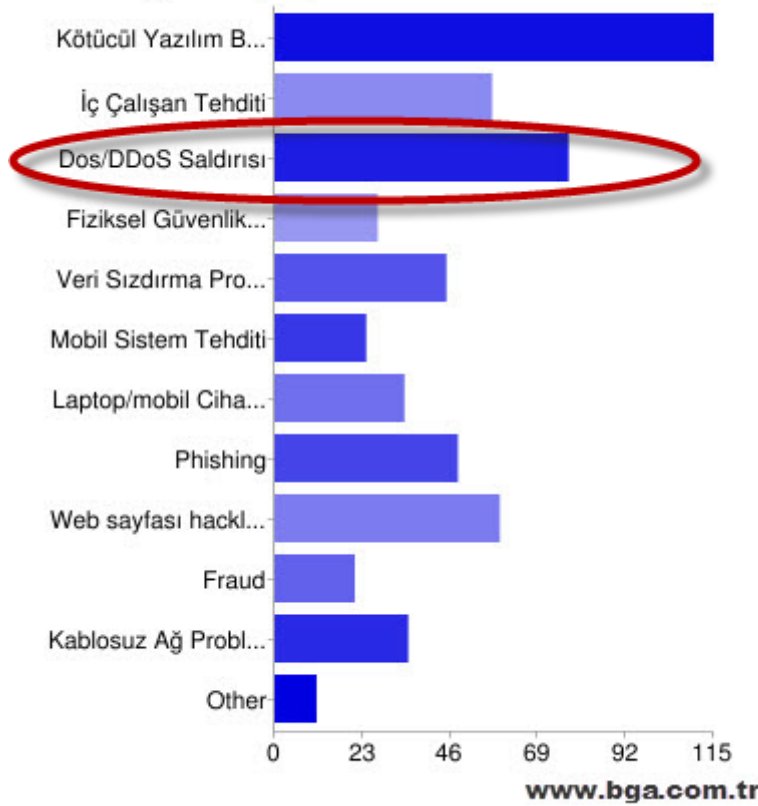
DNS, UDP üzerinden çalışan basit bir protokoldür ve son on yıl incelendiğinde güvenlik açısından karnesi sınıfta kalmaya yetecek kadar kötüdür. DNS 'in sık kullanılıyor olması da bu protokol üzerine gerçekleştirilen istismar çalışmalarını yönlendirmektedir.

DNS güvenliğinden kasıt genellikle DNS kullanılarak gerçekleştirilen dns cache poisoning ve erişilebilirliği hedef alan DOS saldırıları olmaktadır. Özellikle DNS'e yönelik DoS/DDoS saldırıları son yıllarda ciddi oranda artış göstermektedir.

DNS'in UDP üzerine kurulmuş olması ve UDP üzerinden gerçekleştirilen iletişimde kaynak IP adresinin gerçek olup olmadığını anlamının kesin bir yolunun olmaması saldırganın kendini gizleyerek saldırı gerçekleştirmesini kolaylaştırmakta ve engellemeyi zorlaştırmaktadır.

Bilgi Güvenliği AKADEMİSİ 2011 Yılı Siber Tehditler Anketi sonucu da DoS saldırılarının en önemli tehditler arasında yer aldığını göstermektedir.

2011 Yılı içinde karşılaştığınız siber tehditler hangileridir?



DNS Hakkında Temel Bilgiler

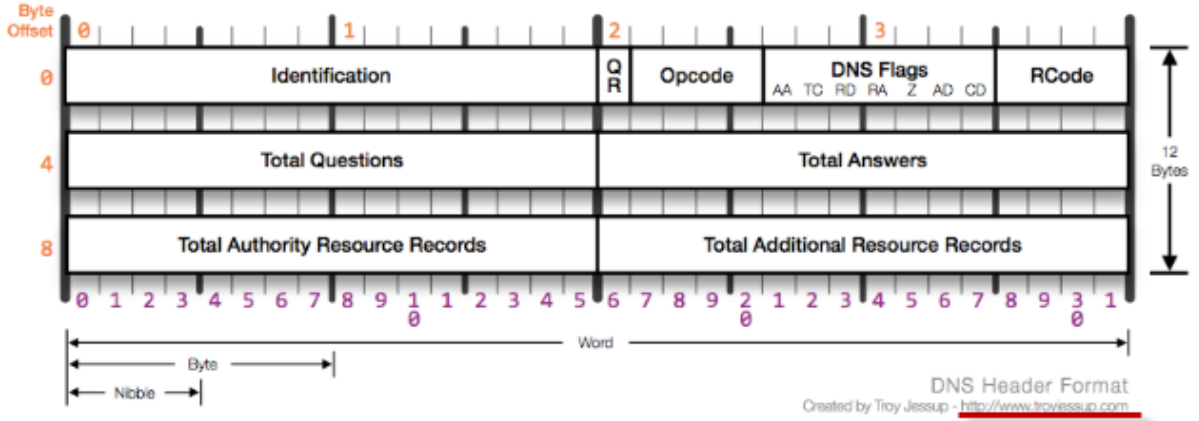
DNS Nedir?

DNS(Domain Name System), temelde TCP/IP kullanılan ağ ortamlarında isim-IP/IP-isim eşleşmesini sağlar ve e-posta trafiğinin sağlıklı çalışması için altyapı sunar. Günümüzde DNS'siz bir ağ düşünülemez denilebilir. Her yerel ağda –ve tüm internet ağında- hiyerarşik bir DNS yapısı vardır.

Mesela bir e-postanın hangi adrese gideceğine DNS karar verir. Bir web sayfasına erişilmek istendiğinde o sayfanın nerede olduğuna, nerede tutulacağına yine DNS üzerinden karar verilir. Bir sistemin DNS sunucusunu ele geçirmek o sistemi ele geçirmek gibidir.

DNS Protokol Detayı

DNS, UDP temelli basit bir protokoldür. DNS başlık bilgisi incelendiğinde istek ve bu isteğe dönecek çeşitli cevaplar (kodlar kullanılarak bu cevapların çeşitleri belirlenmektedir)



Detaylı DNS başlık bilgisi incelemesi için <http://www.networksorcery.com/enp/protocol/dns.htm> adresinden faydalanılabilir.

DNS Paket boyutu

DNS paketi denildiğinde akla DNS isteği ve DNS cevabı gelmektedir. Bir DNS istek paketinin ortalama boyutu 40-60 Byte civarında değişmektedir (alt protokol bilgileri dâhil). DNS cevabı da yine sorgulanan alan adı ve kayda göre değişebilir ve 512 Byte'dan küçük olmalıdır.

Örnek DNS Paketi Boyutu

Dig komutunun çıktısı (son satır: MSG SIZE) incelenerek dönen DNS paketinin boyutu hakkında bilgi edinilebilir. Buradaki boyut bilgisi protokol başlık bilgileri eklenmemiştir.

```
$ dig www.bga.com.tr @8.8.8.8
; <<>> <<>> www.bga.com.tr @8.8.8.8
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15731
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.bga.com.tr.          IN      A
```

```
;; ANSWER SECTION:
```

```
www.bga.com.tr.      59      IN      A      50.22.202.162
```

```
;; Query time: 225 msec
```

```
;; SERVER: 8.8.8.8#53(8.8.8.8)
```

```
;; WHEN: Sun Jan  8 07:28:27 2012
```

```
;; MSG SIZE rcvd: 48
```

DNS cevap paketinin boyutu 512 Byte'ı aşarsa DNS cevabı TCP üzerinden dönmek ister ve DNS sunucu sorgulama yapan sisteme bununla ilgili bilgi(Truncated) döner.

DNS'in TCP üzerinden çalışmasına yazının ilerleyen kısımlarında detaylı değinilecektir.

DNS Kayıt Tipleri

DNS, istek ve cevap mantığıyla çalışan bir protokoldür. İsteklerin çeşitleri de kayıt tipleriyle belirlenir. Bu kayıt tiplerinden en sık kullanılanları aşağıdaki tabloda yer verilmiştir.

DNS Kayıt Tipleri ve İşlevleri

DNS Kayıt Tipi	İşlevi	Örnek Sorgulama
A	Alan adının IP adresini gösterir.	\$dig A abc.com
MX	Alan adına ait e-postaların nereye gideceğini gösterir.	\$dig MX abc.com
NS	İlgili alan adından sorumlu DNS sunucuyu gösterir	\$dig NS abc.com
TXT	DNS sunucuya ait çeşitli özellikleri gösterir.	\$dig TXT abc.com

PTR	Verilen IP adresine ait alan adını gösterir.	\$dig -x ip_adresi
-----	--	--------------------

DNS Sorgulamaları

nslookup, host veya dig komutları kullanılarak dns kayıt tipleri sorgulanabilir. Yazı boyunca dns sorgulamaları için dig yazılımı tercih edilmiştir.

DNS Sorgulamalarını Yorumlama - Dig

Dig, nslookup ve host gibi dns sorgulama araçları yerine kullanılabilen gelişmiş bir araçtır.

ISC tarafından geliştirilen BIND DNS sunucusu ile birlikte geliştirilir ve uzun vadede Linux dağıtımlarında nslookup komutunun yerini alması beklenmektedir. Dig komutu alan adı sorgulama için çalıştırıldığında cevapla birlikte detay bilgiler de döner.

Bu detay bilgiler ek parametrelerle gizlenebilir.

```
# dig www.lifeoverip.net
; <<>> DiG 9.3.3 <<>> www.lifeoverip.net
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47172
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION:
;www.lifeoverip.net.      IN      A
;; ANSWER SECTION:
www.lifeoverip.net. 14400 IN  A      80.93.212.86
;; AUTHORITY SECTION:
lifeoverip.net.      30637 IN  NS      ns3.tekrom.com.
lifeoverip.net.      30637 IN  NS      ns4.tekrom.com.
;; ADDITIONAL SECTION:
ns4.tekrom.com.      91164 IN  A      70.84.223.227
ns3.tekrom.com.      165971 IN  A      70.84.223.226
;; Query time: 213 msec
```



```
:: SERVER: 1.2.39.40#53(1.2.39.40)
:: WHEN: Sat Jan 24 10:56:14 2009
:: MSG SIZE rcvd: 130
```

Çıktıların Detay Açıklaması

Status: NOERROR

Sorgulanan domain adının var olduğunu ve bu domainden sorumlu dns sunucunun sorgulara sağlıklı cevap verdiğini gösterir.

Status: SERVFAIL

Domainin olduğunu fakat domainden sorumlu DNS sunucunun sorgulara sağlıklı cevap veremediğini gösterir. Yani sorun domainden sorumlu DNS sunucusundadır.

Status: NXDOMAIN

Domain ile ilgili ana DNS sunucuların bilgisinin olmadığını gösterir. Bu da ya o domain yoktur ya da bazı sebeplerden dolayı root dns sunuculara yayınlanmamıştır manasına gelir.

Olmayan bir domain sorgulandığında cevap olarak NXDOMAIN denecektir.

```
[root@mail ~]# dig www.huzeyfe.net
; <<>> DiG 9.3.3 <<>> www.huzeyfe.net
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 8419
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;www.huzeyfe.net.      IN      A
;; AUTHORITY SECTION:
```

```
net. 0 IN SOA a.gtld-servers.net. nstld.verisign-  
grs.com. 1232788241 1800 900 604800 900  
;; Query time: 119 msec  
;; SERVER: 1.2.39.40#53(1.2.39.40)  
;; WHEN: Sat Jan 24 11:02:25 2009  
;; MSG SIZE rcvd: 106
```

Soru Kısmı

```
;; QUESTION SECTION:
```

```
;www.lifeoverip.net. IN A
```

DNS sunucuya giden sorgu kısmı.

Cevap Kısmı

```
;; ANSWER SECTION:
```

```
www.lifeoverip.net. 14400 IN A 80.93.212.86
```

DNS sunucudan dönen cevap kısmı.

```
;; AUTHORITY SECTION:
```

```
lifeoverip.net. 30637 IN NS ns3.tekrom.com.
```

```
lifeoverip.net. 30637 IN NS ns4.tekrom.com.
```

Sorgulanan domainden sorumlu dns sunucu adresleri

```
;; ADDITIONAL SECTION:
```

```
ns4.tekrom.com. 91164 IN A 70.84.223.227
```

```
ns3.tekrom.com. 165971 IN A 70.84.223.226
```

Ek bilgiler.

```
;; Query time: 213 msec
```

Sorgulamanın ne kadar sürdüğü.

```
;; SERVER: 1.2.39.40#53(1.2.39.40)
```

sorgulanan dns sunucu

```
;; WHEN: Sat Jan 24 10:56:14 2009 tarih
```

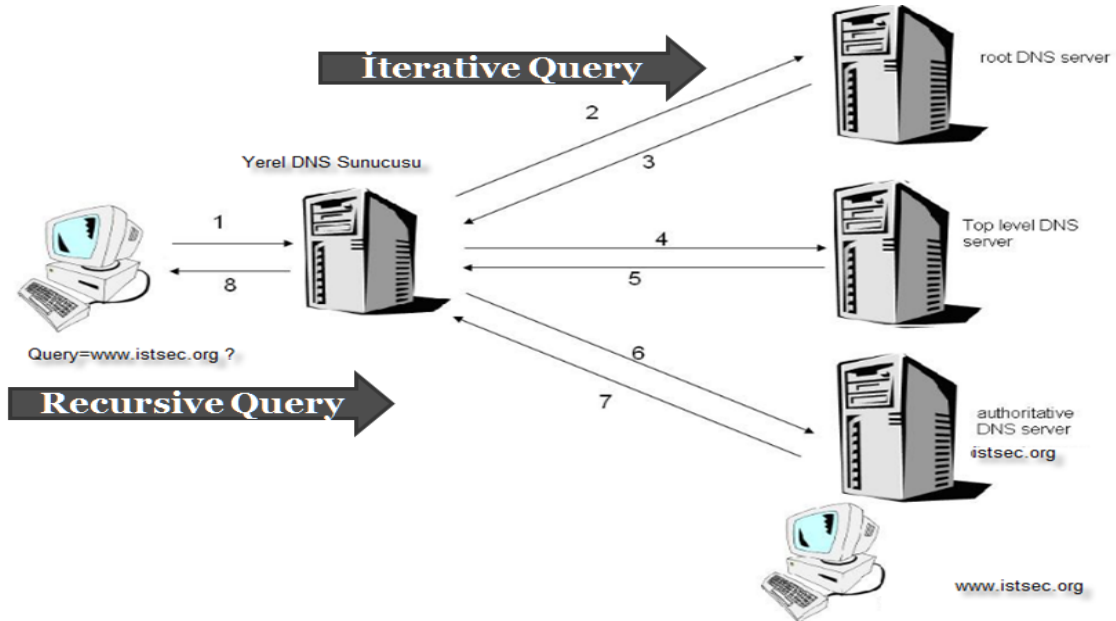
```
;; MSG SIZE rcvd: 130 boyut
```

Ön Belleğe Alma(caching):

Yapılan dns sorgusu sonrası sunucudan dönen cevap bir TTL alanı içerir ve bu alan istemcinin aynı domaine aynı tipte yapacağı bir sonraki sorgulama zamanını belirler.

DNS Sorgu Çeşitleri

DNS sisteminde iki çeşit sorgu tipi vardır. Bunlar, iterative sorgular ve recursive sorgulardır.



Recursive dns sorgular

Recursive sorgulama tipinde istemci dns sunucuya rekursif bir sorgu gönderir ve cevap olarak sorgusuna karşılık gelen tam cevabı – sorguladığı domaine ait cevap- ya da bir hata bekler.

DNS sorgulamaları için kullanılan nslookup komutu öntanımlı olarak rekursif sorgular gönderir, non rekursif sorgu göndermek için nslookup komutu set norecurse seçenekleri ile çalıştırılması gerekir.

Genellikle son kullanıcı – **DNS** sunucu arasındaki sorgulamalar Recursive tipte olur.

Iterative dns sorgular

Iterative sorgu tipinde, istemci dns sunucuya sorgu yollar ve ondan verebileceği en iyi cevabı vermesini bekler, yani gelecek cevap ya ben bu sorgunun cevabını bilmiyorum şu DNS sunucuya sor ya da bu sorgunun cevabı şudur şeklindedir.

Genellikle DNS sunucular arasındaki sorgulamalar Iterative tipte olur.

Genele Açık DNS Sunucular

Herkese açık DNS sunucular (public dns) kendisine gelen tüm istekleri cevaplamaya çalışan türde bir dns sunucu tipidir. Bu tip dns sunucular eğer gerçekten amacı genele hizmet vermek değilse genellikle eksik/yanlış yapılandırmanın sonucu ortaya çıkar.

Bir sunucunun genele açık hizmet(recursive DNS çözücü) verip vermediğini anlamamanın en kolay yolu o DNS sunucusu üzerinden google.com, yahoo.com gibi o DNS sunucuda tutulmayan alan adlarını sorgulamaktır.

Eğer hedef DNS sunucu genele açık bir DNS sunucu olarak yapılandırıldıysa aşağıdakine benzer çıktı verecektir.

```
# dig www.google.com @91.93.119.70
; <<>> DiG 9.5.0-P2.1 <<>> www.google.com @91.93.119.70
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26294
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;www.google.com. IN A
;; ANSWER SECTION:
www.google.com. 44481 IN CNAME www.l.google.com.
```

```
www.l.google.com. 118 IN A 66.102.13.147
www.l.google.com. 118 IN A 66.102.13.99
www.l.google.com. 118 IN A 66.102.13.105
www.l.google.com. 118 IN A 66.102.13.103
www.l.google.com. 118 IN A 66.102.13.104
www.l.google.com. 118 IN A 66.102.13.106
;; Query time: 16 msec
;; SERVER: 91.93.119.70#53(91.93.119.70)
;; WHEN: Sat Jul 24 13:23:59 2010
;; MSG SIZE rcvd: 148
```

Eğer DNS sunucu genele açık hizmet verecek şekilde yapılandırılmadıysa aşağıdakine benzer çıktı verecektir.

```
[root@seclabs ~]# dig @ns1.gezginler.net www.google.com
; <<>> DiG 9.6.1-P1 <<>> @ns1.gezginler.net www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33451
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 0
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
;www.google.com. IN A
;; AUTHORITY SECTION:
. 518400 IN NS H.ROOT-SERVERS.NET.
. 518400 IN NS I.ROOT-SERVERS.NET.
```

```
. 518400 IN NS J.ROOT-SERVERS.NET.  
. 518400 IN NS K.ROOT-SERVERS.NET.  
. 518400 IN NS L.ROOT-SERVERS.NET.  
. 518400 IN NS M.ROOT-SERVERS.NET.  
. 518400 IN NS A.ROOT-SERVERS.NET.  
. 518400 IN NS B.ROOT-SERVERS.NET.  
. 518400 IN NS C.ROOT-SERVERS.NET.  
. 518400 IN NS D.ROOT-SERVERS.NET.  
. 518400 IN NS E.ROOT-SERVERS.NET.  
. 518400 IN NS F.ROOT-SERVERS.NET.  
. 518400 IN NS G.ROOT-SERVERS.NET.  
;; Query time: 140 msec  
;; SERVER: 208.43.98.30#53(208.43.98.30)  
;; WHEN: Sat Aug 7 16:18:15 2010  
;; MSG SIZE rcvd: 243
```

Bir IP aralığındaki tüm public DNS sunucuları bulmak için Nmap NSE (Nmap Scripting Engine) kullanılabilir.

```
root@seclabs:~# nmap -PN -n -sU -p 53 --script=dns-recursion.nse  
91.93.119.65/28
```

```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-07-24 13:19 EDT
```

```
Interesting ports on 91.93.119.64:
```

```
PORT STATE SERVICE
```

```
53/udp open|filtered domain
```

```
Interesting ports on 91.93.119.65:
```

```
PORT STATE SERVICE
```

```
53/udp open|filtered domain
```

Public DNS Sunucular Neden Güvenlik Açısından Risklidir?

Public dns sunucuların özellikle DNS flood saldırılarına karşı sıkıntılıdır. Saldırgan public dns sunucuları kullanarak **amplification dns flood** saldırılarında size ait dns sunuculardan ciddi oranlarda trafik oluşturarak istediği bir sistemi zor durumda bırakabilir.

NOT: DNS sunucu olarak ISC BIND kullanılıyorsa aşağıdaki tanımla recursive dns sorgularına –kendisi hariç- yanıt vermesi engellenebilir.

```
options { allow-recursion { 127.0.0.1; };
```

DNS Sunucu Yazılımları

DNS hizmeti veren çeşitli sunucu yazılımlar bulunmaktadır. ISC Bind, Djbdns, Maradns, Microsoft DNS yazılımları bunlara örnektir. Bu yazılımlar arasında en yoğun kullanıma sahip olanı ISC Bind'dir. Internetin %80 lik gibi büyük bir kısmı Bind dns yazılımı kullanmaktadır. [1]

DNS Sunucu Tipini Belirleme

DNS sunucu yazılımlarına gönderilecek çeşitli isteklerin cevapları incelenerek hangi tipte oldukları belirlenebilir.

Bunun için temelde iki araç kullanılır:

1. Nmap gibi bir port tarama/yazılım belirleme aracı
2. Dig, nslookup gibi klasik sorgulama araçları

Nmap Kullanarak DNS Sunucu Versiyonu Belirleme

```
#nmap -PN -sU -sV dns_sunucu_ip_adresi
```

```
root@bt:~# nmap -sU -sV -p 53 ns1.abcdef.com.tr.
```

```
Starting Nmap 5.59BETA1 ( http://nmap.org ) at 2011-12-18 17:52 EET
```

```
Nmap scan report for ns1.abcdef.com.tr. (1.1.1.13)
```

```
Host is up (0.0044s latency).
```

PORT STATE SERVICE VERSION

53/udp open domain ISC **BIND (Fake version: 9.3.6-P1-RedHat-9.3.6-4.P1.el5)**

Diğer bir yöntem de

```
dig version.bind chaos txt @dns_sunucu_ip_adresi
```

Bu yöntem sadece bind kullanan sistemlerde sağlıklı sonuçlar üretir.

İsteğe Göre DNS Paketi Üretmek

TCP/IP paket üreteçleri kullanılarak isteğe göre DNS paketi oluşturulabilir. DNS paketi oluşturmak için DNS istek ve DNS cevap paketlerine ait temel başlık bilgilerinin bilinmesi gerekmektedir.

DNS Paketi Üretim Araçları

Güvenlik ve performans testlerinde kullanılmak üzere tercih edilen DNS paketi üretim araçları.

- Scapy
- Mz
- Hping
- Netstress

Örnek DNS Paketi Üretimi

```
# mz -A 5.5.5.5 -B 1.2.39.40 -t dns "q=www.bga.com.tr" -c 1000
```

Mausezahn will send 1000 frames... 0.02 seconds (50000 packets per second)

Mz kullanılarak üretilebilecek detaylı DNS paketleri için -t dns help parametreleri yeterli olacaktır.

```
root@bt:~# mz -t dns help
```

Mausezahn 0.34.9 - (C) 2007-2009 by Herbert Haas -
http://www.perihel.at/sec/mz/

| DNS type: Send Domain Name System Messages.

|

| Generally there are two interesting general DNS messages: queries and answers. The easiest

| way is to use the following syntax:

|

query|q = <name>[:<type>] where type is per default "A"
(and class is always "IN")

answer|a = [<type>:<ttl>:]<rdata> ttl is per default 0.
= [<type>:<ttl>:]<rdata>/[<type>:<ttl>:]<rdata>/...

Note: If you only use the 'query' option then a query is sent. If you additionally add

an 'answer' then an answer is sent.

Examples:

q = www.xyz.com

q = www.xyz.com, a=192.168.1.10

q = www.xyz.com, a=A:3600:192.168.1.10

q = www.xyz.com, a=CNAME:3600:abc.com/A:3600:192.168.1.10

Note: <type> can be: A, CNAME, or any integer

OPTIONAL parameter hacks: (if you don't know what you do this might cause invalid packets)

Parameter	Description	query / reply)
<hr/>		
request/response reply flag only	request / n.a.
id packet id (0-65535)	random / random
opcode (or op) accepts values 0..15 or one of these keywords:	std / 0
	= std Standard Query
	= inv Inverse Query
	= sts Server Status Request
aa or !aa Authoritative Answer	UNSET / SET
tc or !tc Truncation	UNSET / UNSET
rd or !rd Recursion Desired	SET / SET
ra or !ra Recursion Available	UNSET / SET
z Reserved (takes values 0..7) (z=2...authenticated)	0 / 0
rcode Response Code (0..15); interesting values are:	0 / 0
	= 0 No Error Condition
	= 1 Unable to interpret query due to format error
	= 2 Unable to process due to server failure
	= 3 Name in query does not exist
	= 4 Type of query not supported
	= 5 Query refused

Count values (values 0..65535) will be set automatically! You should not set these

```
| values manually except you are interested in invalid packets.
| qdcount (or qdc) ..... Number of entries in question section      1 / 1
| ancoun (or anc) ..... Number of RRs in answer records section      0 / 1
| nscount (or nsc) ..... Number of name server RRs in authority      0 / 0
|                          records section
| arcount (or arc) ..... Number of RRs in additional records section 0 / 0
```

DNS Güvenlik Zafiyetleri

DNS çok önemli bir protokol olduğu için yaygın kullanılan DNS sunucu yazılımları hem güvenlik uzmanları hem de hackerlar tarafından sık sık kurcalanır ve güvenlik zafiyetleri yayınlanır.

Genel olarak DNS sunucularda bulunan güvenlik zafiyetlerini üç kategoride incelenebilir:

- DNS sunucunun çalışmasını durdurabilecek zafiyetler
- DNS sunucunun güvenliğini sıkıntıya sokacak zafiyetler
- DNS sunucuyu kullanan istemcilerin güvenliğini sıkıntıya sokabilecek zafiyetler

2011 Yılı ISC Bind Yazılımında Çıkmış Güvenlik Zafiyetleri

Son yıllarda sık kullanılan DNS yazılımları incelendiğinde DoS zafiyetlerinin daha fazla bulunduğu görülmektedir.

1. BIND 9 Resolver crashes after logging an error in query.c

Severity: Serious

Exploitable: Remotely

2. ISC BIND 9 Remote packet Denial of Service against Authoritative and Recursive Servers

Severity: High

Exploitable: Remotely

3. ISC BIND 9 Remote Crash with Certain RPZ Configurations

Severity: High

Exploitable: Remotely

4. Large RRSIG RRsets and Negative Caching can crash named

Severity: High

Exploitable: remotely

5. RRSIG Queries Can Trigger Server Crash When Using Response Policy Zones

Severity: High

Exploitable: remotely

6. BIND: Server Lockup Upon IXFR or DDNS Update Combined with High Query Rate

Severity: High

Exploitable: remotely

Yıllara Göre ISC Bind Yazılımında Bulunan Güvenlik Zafiyetleri

Aşağıdaki çıktıdan da görüleceği gibi Bind üzerinde çıkan açıklıkların büyük oranı (%57) DOS tipindedir.

Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
1999	4	3		1											
2000	3	1	1	1											
2001	5			2								3			
2002	10	4	4	4											
2003	1														
2005	2	2		1											
2006	5	4										1			
2007	6	3													
2008	3	2	1		1										3
2009	4	1								2					
2010	9	4									1				
2011	6	5													
Total	58	30	6	9	1	0.0	0.0	0.0	0.0	3.4	1.7	6.9	0.0	0.0	3
% Of All		51.7	10.3	15.5	1.7	0.0	0.0	0.0	0.0	3.4	1.7	6.9	0.0	0.0	

<http://www.cvedetails.com>'un katkılarıyla

DNS Protokolünde IP Sahteciliği (IP Spoofing)

DNS, UDP tabanlı bir protokol olduğu için hem DNS istekleri hem de DNS cevaplarında kullanılan ip adresleri istenildiği gibi belirlenebilir.

IP spoofing yapılabilir olması demek hem DNS isteklerinin hem de cevaplarının sahte olabileceği anlamına gelmektedir. Sahte DNS isteği üretmeyi engelleyecek herhangi bir yöntem bulunmamaktadır (URPF [2] hariç)

UDP katmanında IP spoofing için bir önlem olmaması nedeniyle DNS ip sahteciliğini önlemek için uygulama seviyesinde iki temel önlem almıştır. Bu önlemlerden ilki DNS TXID başlık bilgisinin random olması diğeri de kaynak port numarasının random olarak belirlenmesidir.

Kaynak Portun Rastgeleliğinin Sorgulanması

DNS cevabı olarak dönen paketlerin kaynak portlarının sabit mi yoksa rastgele mi belirlendiği aşağıdaki nmap komutuyla belirlenebilir.

```
root@bt:~# nmap -sU -p 53 --script=dns-random-srcport 8.8.8.8
Starting Nmap 5.59BETA1 ( http://nmap.org ) at 2011-12-18 17:43 EET
Nmap scan report for google-public-dns-a.google.com (8.8.8.8)
Host is up (0.041s latency).
PORT      STATE SERVICE
53/udp    open  domain
|_dns-random-srcport: 74.125.38.86 is GREAT: 6 queries in 3.0 seconds
from 6 ports with std dev 6324

Nmap done: 1 IP address (1 host up) scanned in 0.66 seconds
```

DNS Transaction ID Değerinin Rastgeleliğinin Sorgulanması

DNS cevabı olarak dönen paketlerdeki TXID değerinin sabit mi yoksa rastgele mi belirlendiği aşağıdaki nmap komutuyla belirlenebilir.

```
root@bt:~# nmap -sU -p 53 --script=dns-random-txid ns1.abc.com.tr
```

```
Starting Nmap 5.59BETA1 ( http://nmap.org ) at 2011-12-18 17:45 EET
Nmap scan report for ns1.abc.com.tr (1.1.3.3)
Host is up (0.0035s latency).
PORT      STATE SERVICE
53/udp    open  domain
|_dns-random-txid: 91.199.73.23 is GREAT: 26 queries in 5.2 seconds
from 26 txids with std dev 21394
Nmap done: 1 IP address (1 host up) scanned in 6.10 seconds
```

DNS ve TCP İlişkisi

DNS paketleri 512 byte'ı geçmediği müddetçe UDP üzerinden taşınabilir. 512 byte'ı aşan DNS cevapları UDP üzerinden taşınamayacağı için TCP kullanılır(EDNS hariç).

Cevabın 512 Byte'dan fazla olduğu ve TCP üzerinden taşınması gerektiğini istemci, DNS paketine(dönen DNS paketi) ait başlık bilgisine bakarak anlamaktadır.

Aşağıdaki gibi DNS paketinde Truncated=1 olması durumunda dns isteğinde bulunan aynı isteği TCP/53 üzerinden yapmayı deneyecektir.

```
Domain Name System (response)
[Request In: 1]
[Time: 0.152073000 seconds]
Transaction ID: 0x28b3
Flags: 0x8380 (Standard query response, No error)
1... .. = Response: Message is a response
.000 0... .. = Opcode: Standard query (0)
.... .0.. .. = Authoritative: Server is not an authority for domain
.... ..1. .... = Truncated: Message is truncated
.... ..1 .... = Recursion desired: Do query recursively
.... .... 1... .. = Recursion available: Server can do recursive queries
```

```
.....0..... = Z: reserved (0)
```

```
.....0..... = Answer authenticated: Answer/authority portion was not
authenticated by the server
```

EDNS destekli DNS sunucularda dns cevapları ~4000 Byte olabilir.

Aşağıdaki örnekte cevabı 512 Byte'ı aşacak şekilde yapılandırılmış bir alan adı kaydının sorgulaması ve cevabın TCP üzerinden dönüşü gösterilmektedir.

```
[huzeyfe@seclabs ~]$ dig test.bga.com.tr @1.2.39.39
```

```
;; Truncated, retrying in TCP mode.
```

```
; <<> <<> test.bga.com.tr @1.2.39.39
```

```
;; global options: printcmd
```

```
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3117
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 36, AUTHORITY: 2, ADDITIONAL: 0
```

```
;; QUESTION SECTION:
```

```
;test.bga.com.tr.      IN      A
```

```
;; ANSWER SECTION:
```

```
test.bga.com.tr.      60      IN      A      1.2.3.0
```

```
test.bga.com.tr.      60      IN      A      1.2.3.1
```

```
test.bga.com.tr.      60      IN      A      1.2.3.2
```

```
test.bga.com.tr.      60      IN      A      1.2.3.3
```

```
test.bga.com.tr.      60      IN      A      1.2.3.4
```

```
test.bga.com.tr.      60      IN      A      1.2.3.5
```

```
test.bga.com.tr.      60      IN      A      1.2.3.6
```

test.bga.com.tr.	60	IN	A	1.2.3.8
test.bga.com.tr.	60	IN	A	1.2.3.11
test.bga.com.tr.	60	IN	A	1.2.3.12
test.bga.com.tr.	60	IN	A	1.2.3.41
test.bga.com.tr.	60	IN	A	1.2.3.42
test.bga.com.tr.	60	IN	A	1.2.3.43
test.bga.com.tr.	60	IN	A	1.2.3.44
test.bga.com.tr.	60	IN	A	1.2.3.45
test.bga.com.tr.	60	IN	A	1.2.3.46
test.bga.com.tr.	60	IN	A	1.2.3.47
test.bga.com.tr.	60	IN	A	1.2.3.48
test.bga.com.tr.	60	IN	A	1.2.3.49
test.bga.com.tr.	60	IN	A	1.2.34.21
test.bga.com.tr.	60	IN	A	1.2.34.23
test.bga.com.tr.	60	IN	A	1.2.34.24
test.bga.com.tr.	60	IN	A	1.2.34.25
test.bga.com.tr.	60	IN	A	1.2.34.26
test.bga.com.tr.	60	IN	A	1.2.34.28
test.bga.com.tr.	60	IN	A	1.2.34.29
test.bga.com.tr.	60	IN	A	1.2.34.30
test.bga.com.tr.	60	IN	A	1.2.34.31
test.bga.com.tr.	60	IN	A	1.4.34.32
test.bga.com.tr.	60	IN	A	1.5.34.32
test.bga.com.tr.	60	IN	A	1.6.34.32
test.bga.com.tr.	60	IN	A	1.7.34.32
test.bga.com.tr.	60	IN	A	1.8.34.32
test.bga.com.tr.	60	IN	A	1.9.34.32
test.bga.com.tr.	60	IN	A	222.222.222.223
test.bga.com.tr.	60	IN	A	222.222.222.224


```
:: AUTHORITY SECTION:
bga.com.tr.      60   IN   NS   ns1.bga.com.tr.
bga.com.tr.      60   IN   NS   ns2.bga.com.tr.

:: Query time: 156 msec
:: SERVER: 1.2.39.39#53(1.2.39.39)
:: WHEN: Sun Jan  8 08:13:30 2012
;; MSG SIZE rcvd: 645
```

DNS sunucu tarafından istek öncelikle “truncated” mesajı ile TCP’e çevriliyor ve DNS isteği yapan tarafın TCP üzerinden tekrar DNS isteği göndermesi isteniyor.

NOT: Çoğu DNS sunucu TCP/53’e kapalı olduğu için bu tip isteklere cevap vermeyecektir.

DNS sunucu üzerinde TCP/53’ün açık olup olmadığı bu porta gönderilecek SYN paketlerine SYN/ACK cevabının dönmesi ile anlaşılabilir.

```
[root@seclabs ~]# hping -S -p 53 8.8.8.8 -c 2
HPING 8.8.8.8 (eth0 8.8.8.8): S set, 40 headers + 0 data bytes
len=46 ip=8.8.8.8 ttl=47 id=46413 sport=53 flags=SA seq=0 win=5720
rtt=47.1 ms
len=46 ip=8.8.8.8 ttl=47 id=62723 sport=53 flags=SA seq=1 win=5720
rtt=47.3 ms

--- 8.8.8.8 hping statistic ---
2 packets tramitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 47.1/47.2/47.3 ms
```

DNS sunucu üzerinde TCP/3 portu açık ise bu porta yönelik SYN Flood, TCP Connection flood tipinde DDoS atakları gerçekleştirilebilir.

DNS sunucu önünde SYN cookie, SYN Proxy ya da benzeri bir koruma sistemi yoksa DNS sunucu kısa sürede hizmet veremez hale gelecektir.

DNS'e Yönelik DoS ve DDoS Saldırıları

DNS hizmetine yönelik Dos/DDoS saldırılarını iki kategoride incelenebilir

- Yazılım temelli DoS saldırıları
- Tasarım temelli DoS saldırıları

Yazılım Temelli DoS Saldırıları

BIND 9 Dynamic Update DoS Zaafiyeti

28.07.2009 tarihinde ISC Bind yazılım geliştiricileri tüm Bind 9 sürümlerini etkileyen acil bir güvenlik zaafiyeti duyurdular. Duyuruya göre eğer DNS sunucunuz Bind9 çalıştırıyorsa ve üzerinde en az bir tane yetkili kayıt varsa bu açıklıktan etkileniyor demektir.

Aslında bu zafiyet bind 9 çalıştıran tüm dns sunucularını etkiler anlamına gelmektedir. Bunun nedeni dns sunucunuz sadece caching yapıyorsa bile üzerinde localhost için girilmiş kayıtlar bulunacaktır ve açıklık bu kayıtları değerlendirerek sisteminizi devre dışı bırakabilir.

Güvenlik Açığı Nasıl Çalışır?

Açıklık dns sunucunuzdaki ilgili zone tanımı(mesela:www.lifeoverip.net) için gönderilen özel hazırlanmış dynamic dns update paketlerini düzgün işleyememesinden kaynaklanmaktadır.

Açıklığın sonucu olarak dns servisi veren named prosesi durmakta ve DNS sorgularına cevap dönememektedir.

```
# perl dnstest.pl
;; HEADER SECTION
;; id = 51444
;; qr = 0      opcode = UPDATE  rcode = NOERROR
```

```
;; zocount = 0 prcount = 1 upcount = 1 adcount = 1
;; ZONE SECTION (1 record)
;; test.com.      IN      SOA
;; PREREQUISITE SECTION (1 record)
www.test.com.    0      IN      ANY  ; no data
;; UPDATE SECTION (1 record)
www.test.com.    0      ANY  ANY  ; no data
;; ADDITIONAL SECTION (1 record)
<key. 0      IN      ANY  ; rdlength = 0
```

Bu aşamadan sonra ilgili zone barındıran dns sunucu sorgulara cevap veremez hale gelmektedir.

Örnek Hata Logu:

```
Aug      1      16:03:52      mail      named[45293]:
/usr/src/lib/bind/dns/../../../../contrib/bind9/lib/dns/db.c:595: REQUIRE(type !=
((dns_rdatatype_t)dns_rdatatype_any)) failed
Aug  1 16:03:52 mail named[45293]: exiting (due to assertion failure)
Aug  1 16:03:52 mail kernel: pid 45293 (named), uid 0: exited on signal 6
(core dumped)
```

DNS Flood DoS/DDoS Saldırıları

Bu saldırı tipi genelde iki şekilde gerçekleştirilir:

- Hedef DNS sunucuya kapasitesinin üzerinde (bant genişliği olarak değil) DNS istekleri göndererek, normal isteklere cevap veremeyecek hale gelmesini sağlamak
- Hedef DNS sunucu önündeki Firewall/IPS'in "session" limitlerini zorlayarak Firewall arkasındaki tüm sistemlerin erişilemez olmasını sağlamak

Her iki yöntem için de ciddi oranlarda DNS sorgusu gönderilmesi gerekir. İnternet üzerinden edinilecek poc(proof of concept) araçlar incelendiğinde çoğunun perl/python gibi script dilleriyle yazıldığı ve paket gönderme kapasitelerinin max 10.000-15.000 civarlarında olduğu görülecektir.

Bu araçlar kullanılarak ciddi DNS DDoS testleri gerçekleştirilemez.

DNS Flood DDoS Saldırıları

DNS Flood ve UDP Flood DDoS Saldırıları Arasındaki Farklar

UDP flood saldırılarında temel amaçlardan biri UDP servisini koruyan Güvenlik Duvarı'nın oturum tablosunun dolması ve cevap veremez hale gelmesidir.

Boş UDP/53 paketi ile DNS paketi arasındaki farklar

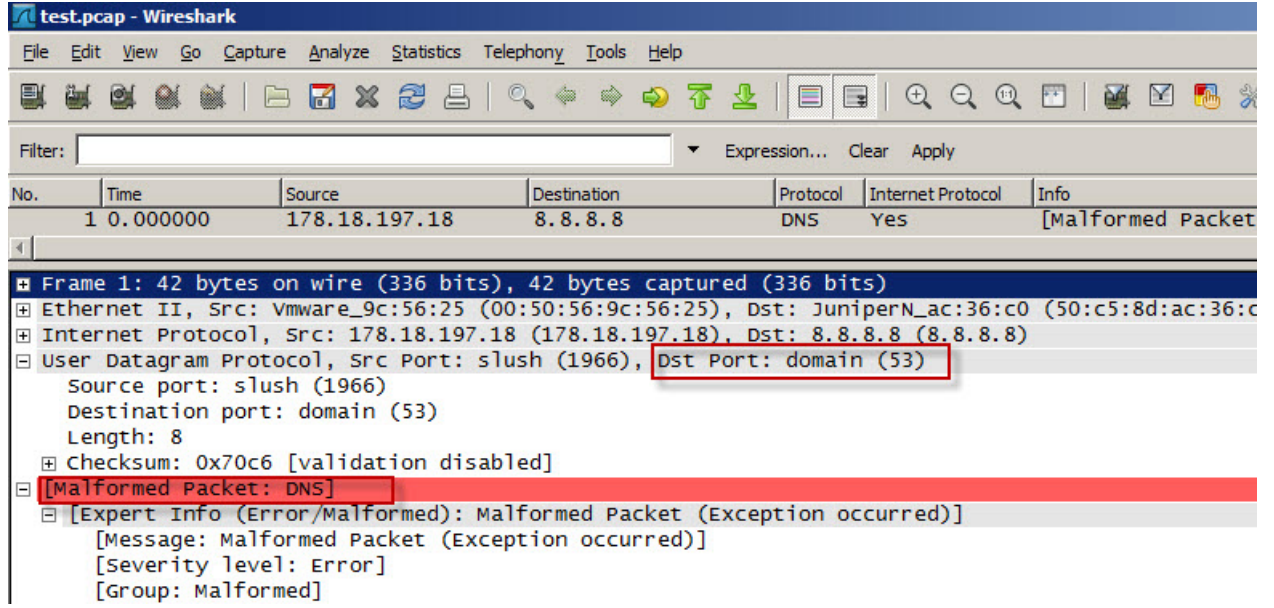
DNS Flood saldırılarında sık yapılan hatalardan biri UDP 53 portuna gönderilen her paketin DNS olduğunu düşünmektir. Bu şekilde gerçekleştirilecek DDoS denemeleri hedef sistem önündeki IPS ve benzeri sistemler tarafından protokol anormalliğine takılarak hedefe ulaşamayacaktır.

UDP port 53'e gönderilen boş /doludns olmayan içerik ve DNS istekleri farklıdır. Hping gibi araçlar kullanılarak gerçekleştirilen udp port 53 flood saldırıları DNS flood saldırısı olarak adlandırılmaz.

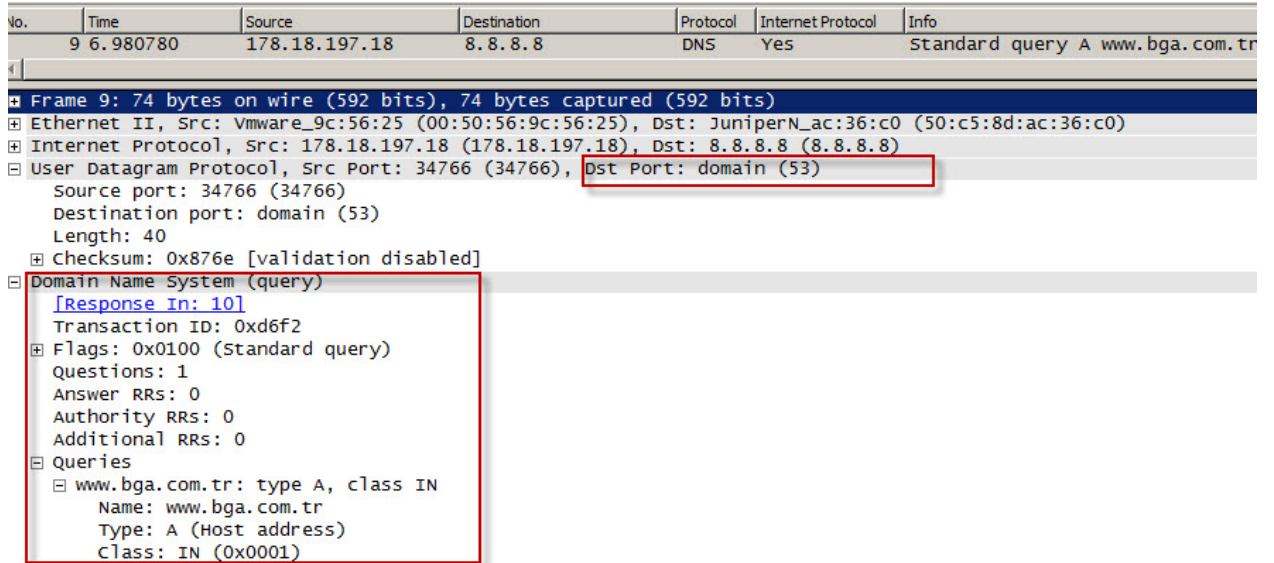
UDP Flood belirlenen hedefe boş UDP (ya da rastgele doldurularak) paketleri hedefe göndermektir.

DNS flood ise DNS servisine yönelik rastgele DNS istekleri (DNS istekleri UDP)gönderilerek gerçekleştirilir.

UDP Port 53 Paketi(Boş UDP Paketi)



Gerçek DNS Paketi Örneği



Ancak DNS sorgularını ikili olarak kaydedip bunları Hping kullanarak hedefe dns sorgusu gibi gönderme işlemi yapılabilir.

Aşağıda adım adım Hping kullanarak nasıl DNS flood denemeleri gerçekleştirileceği anlatılmıştır. Test edilen her alan adı için bu şekilde dns sorgusu ikili dosya olarak kaydedilmeli ve hping'e parametre olarak verilmelidir.

<http://blog.lifeoverip.net/2011/07/25/hping-kullanarak-dns-flood-dosddos-saldirilari-gerceklestirme/>

Netstress Kullanarak DNS Flood DDoS Atağı Gerçekleştirme

Netstress, saniyede ortalama 400.000 DNS isteği gönderebilmektedir. Bandwith ve test için kullanılan makine gücüne bağlı olarak saniyede 3.000.000 –teorik olarak- DNS isteğine kadar çıkabilmektedir.

```
[root@seclabs netstress-2.2.4] ./netstress_fullrandom -d 8.8.8.8 -a dns -t  
a -n 4 -P 53
```

```
----- netstress stats -----
```

```
----- netstress stats -----
```

```
PPS:          122374  
BPS:          23495936  
MPS:           22.41  
Total seconds active: 3  
Total packets sent: 367124
```

```
-----
```

```
PPS:          110916  
BPS:          21295936  
  
MPS:           20.31  
Total seconds active: 3  
Total packets sent: 332749
```

```
----- netstress stats -----
```

```
PPS:          116075  
BPS:          22286528  
MPS:           21.25  
Total seconds active: 3
```

Total packets sent: 348227

Saldırılarda Sahte(spoofed) IP Adreslerinin Kullanımı

DNS flood DoS/DDoS saldırıları genellikle sahte ip adresleri kullanılarak gerçekleştirilir. Sahte IP adresi kullanımı da temelde iki şekilde olmaktadır.

1- Rastgele seçilmiş ip adresleri

2-Bilinen DNS sunucuların IP adreslerinin kaynak olarak kullanımı.

Netstress her iki yöntemi de gerçekleştirebilmektedir.

```
--- NetStress Configuration ---
-----
Select your attacks --->
Source IP type (Random) --->
[*] Random Source Port
[ ] Random Destination Port
[ ] Request Random URLs In GET Flood
---
Load an Alternate Configuration File
Save an Alternate Configuration File
```

Rastgele IP Adreslerinden DNS Flood DDoS Denemesi

```
IP 79.137.73.96.timbuktu-srv3 > 8.8.8.8.domain: 46615+ A?
mk2082987667.net. (34)
IP 111.92.61.23.ms-sql-m > 8.8.8.8.domain: 53015+ A? mk904269199.net.
(33)
IP 172.143.231.2.nucleus > 8.8.8.8.domain: 31255+ A? mk1623054336.net.
(34)
```

IP 183.146.232.84 .os-licman	>	8.8.8.8.domain:	51223+	A?
mk1557036557.net. (34)				
IP 126.151.69.29.elan	>	8.8.8.8.domain:	37143+	A?
mk650936905.net. (33)				
IP 242.130.243.90.af	>	8.8.8.8.domain:	46615+	A?
mk191742245.net. (33)				
IP 104.22.108.9.eicon-x25	>	8.8.8.8.domain:	34839+	A?
mk116675712.net. (33)				
IP 152.236.4.72.sbook	>	8.8.8.8.domain:	43799+	A?
mk1620707131.net. (34)				
IP 181.154.241.105.nms	>	8.8.8.8.domain:	46103+	A?
mk958054365.net. (33)				
IP 70.247.18.90.nrcabq-lm	>	8.8.8.8.domain:	55319+	A?
mk588765509.net. (33)				
IP 78.237.69.35 .localinfosrvr	>	8.8.8.8.domain:	43543+	A?
mk893603990.net. (33)				
IP 65.29.179.63.dca	>	8.8.8.8.domain:	32023+	A?
mk1585426588.net. (34)				
IP 229.58.66.14.iclpv-sc	>	8.8.8.8.domain:	41751+	A?
mk1300398297.net. (34)				
IP 147.228.32.81.prm-nm-np	>	8.8.8.8.domain:	35095+	A?
mk2137711157.net. (34)				
IP 80.77.213.23 .genie-lm	>	8.8.8.8.domain:	39191+	A?
mk605526808.net. (33)				
IP 56.211.51.117.goldleaf-licman	>	8.8.8.8.domain:	29719+	A?
mk770182864.net. (33)				
IP 217.205.176.45.proxima-lm	>	8.8.8.8.domain:	33559+	A?
mk1699691839.net. (34)				
IP 137.133.60.40.netware-csp	>	8.8.8.8.domain:	45079+	A?
mk558564538.net. (33)				
IP 142.165.219.3 .oc-lm	>	8.8.8.8.domain:	44567+	A?
mk1600988605.net. (34)				
IP 207.83.168.58.informatik-lm	>	8.8.8.8.domain:	40727+	A?
mk1082503180.net. (34)				
IP 232.42.197.57.blueberry-lm	>	8.8.8.8.domain:	29719+	A?
mk1479252390.net. (34)				
IP 245.252.232.82.kjtsiteserver	>	8.8.8.8.domain:	50199+	A?
mk2105094468.net. (34)				
IP 122.181.116.57.sbook	>	8.8.8.8.domain:	47127+	A?
mk260490390.net. (34)				

(33)

Bilinen DNS Sunucu IP Adreslerinden DNS Flood Gerçekleştirme

Aynı şekilde subnet kullanımı da gerçekleştirilebilir. Tüm atak bir subnet ya da bir ip aralığı ya da bir ülke ip adresinden geliyormuş gibi gösterilebilir. Bu tip saldırılarda hedef DNS sunucunun önünde gönderilen paket sayısına göre rate limiting/karantina uygulayan IPS, DDoS Engelleme sistemi varsa paket gönderilen sahte ip adresleri bu cihazlar tarafından engellenecektir. Bu da saldırgana internet üzerinde istediği ip adreslerini engelleme lüksü vermektedir.

```
[root@seclabs netstress-2.2.4# ./netstress_patternip_randomport -s
1.2.39. -d 8.8.8.8 -a dns -t a -n 4 -P 53
```

IP 1.2.39.85.ibm-pps > 8.8.8.8.domain: 44823+ A? mk1650317290.net. (34)

IP 1.2.39.252.chromagrafx > 8.8.8.8.domain: 55063+ A? mk885119093.net. (33)

IP 1.2.39.234.ms-sql-s > 8.8.8.8.domain: 50711+ A? mk1480111032.net. (34)

IP 1.2.39.251.ndm-server > 8.8.8.8.domain: 47127+ A? mk211899066.net. (33)

IP 1.2.39.25.gv-us > 8.8.8.8.domain: 37143+ A? mk521909797.net. (33)

IP 1.2.39.223.netlabs-lm > 8.8.8.8.domain: 37911+ A? mk581801644.net. (33)

IP 1.2.39.51.dwf > 8.8.8.8.domain: 58391+ A? mk58082171.net. (32)

IP 1.2.39.229.alta-ana-lm > 8.8.8.8.domain: 51223+ A? mk1292956077.net. (34)

IP 1.2.39.27.wmc-log-svc > 8.8.8.8.domain: 37143+ A? mk836758631.net. (33)

IP 1.2.39.246.ms-sql-m > 8.8.8.8.domain: 31767+ A? mk269450214.net. (33)

IP 1.2.39.46.nms_topo_serv > 8.8.8.8.domain: 40471+ A? mk1053310989.net. (34)

IP 1.2.39.58.informatik-lm > 8.8.8.8.domain: 40215+ A? mk1614262787.net.

(34)

IP 1.2.39.25.nms > 8.8.8.8.domain: 37399+ A? mk2058427683.net. (34)

IP 1.2.39.154.menandmice-dns > 8.8.8.8.domain: 50967+ A?
mk1409302037.net. (34)

IP 1.2.39.193.ndm-server > 8.8.8.8.domain: 39447+ A? mk1607501323.net.
(34)

IP 1.2.39.217.innosys > 8.8.8.8.domain: 30999+ A? mk34802544.net. (32)

IP 1.2.39.121.dbsa-lm > 8.8.8.8.domain: 41239+ A? mk460693496.net. (33)

IP 1.2.39.105.cadkey-licman > 8.8.8.8.domain: 54551+ A?
mk280556416.net. (33)

IP 1.2.39.28.eicon-slp > 8.8.8.8.domain: 63511+ A? mk1903514959.net. (34)

IP 1.2.39.121.world-lm > 8.8.8.8.domain: 51223+ A? mk62585107.net. (32)

IP 1.2.39.163.nms > 8.8.8.8.domain: 536+ A? mk1231830388.net. (34)

IP 1.2.39.106.eicon-x25 > 8.8.8.8.domain: 58391+ A? mk178733798.net.
(33)

IP 1.2.39.223.bbn-mmx > 8.8.8.8.domain: 62487+ A? mk1952301711.net.
(34)

IP 1.2.39.195.taligent-lm > 8.8.8.8.domain: 34327+ A? mk851097929.net.
(33)

Özellikle son zamanlarda Türk Telekom, Google ve OpenDNS'in ip adresleri kullanarak gerçekleştirilen DNS flood saldırılarına rastlanmaktadır.

Bu tip gerçek DNS sunucuların ip adreslerinden geliyormuş gibi gerçekleştirilen DNS flood ataklarını engellemek çok zordur.

DNS Performans Ölçümü

DNS sunucuya gelen isteklere döndüğü paketlerin süresi ölçülürse DNS sunucunun performansı ile ilgili bilgi edinilebilir. Performans ölçümü için çeşitli araçlar bulunmaktadır. En temel araç Linux sistemlerle birlikte gelen dig komutudur. Dig komutu ile DNS sunucunun cevap vermesinin ne kadar sürdüğü belirlenebilir.

Dig komutu çıktısındaki “;; **Query time**” satırı DNS’in sorguya döndüğü cevap süresini belirler.

```
[huzeyfe@seclabs ~]$ dig www.bga.com.tr @8.8.8.8

; <<>> DiG 9.3.6-P1-RedHat-9.3.6-16.P1.el5 <<>> www.bga.com.tr @8.8.8.8
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57086
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.bga.com.tr.          IN      A

;; ANSWER SECTION:
www.bga.com.tr.          37      IN      A      50.22.202.163

;; Query time: 41 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sat Oct 1 21:26:56 2011
;; MSG SIZE rcvd: 48
```

Saldırı altındaki DNS sunucunun cevabı

```
[huzeyfe@seclabs ~]$ dig www.bga.com.tr @4.2.2.1
```

```
; <<>> D <<>> www.bga.com.tr @4.2.2.1
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 17532
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.bga.com.tr.          IN      A

;; Query time: 326 msec
;; SERVER: 4.2.2.1#53(4.2.2.1)
;; WHEN: Sat Oct 1 21:27:54 2011
;; MSG SIZE rcvd: 32
```

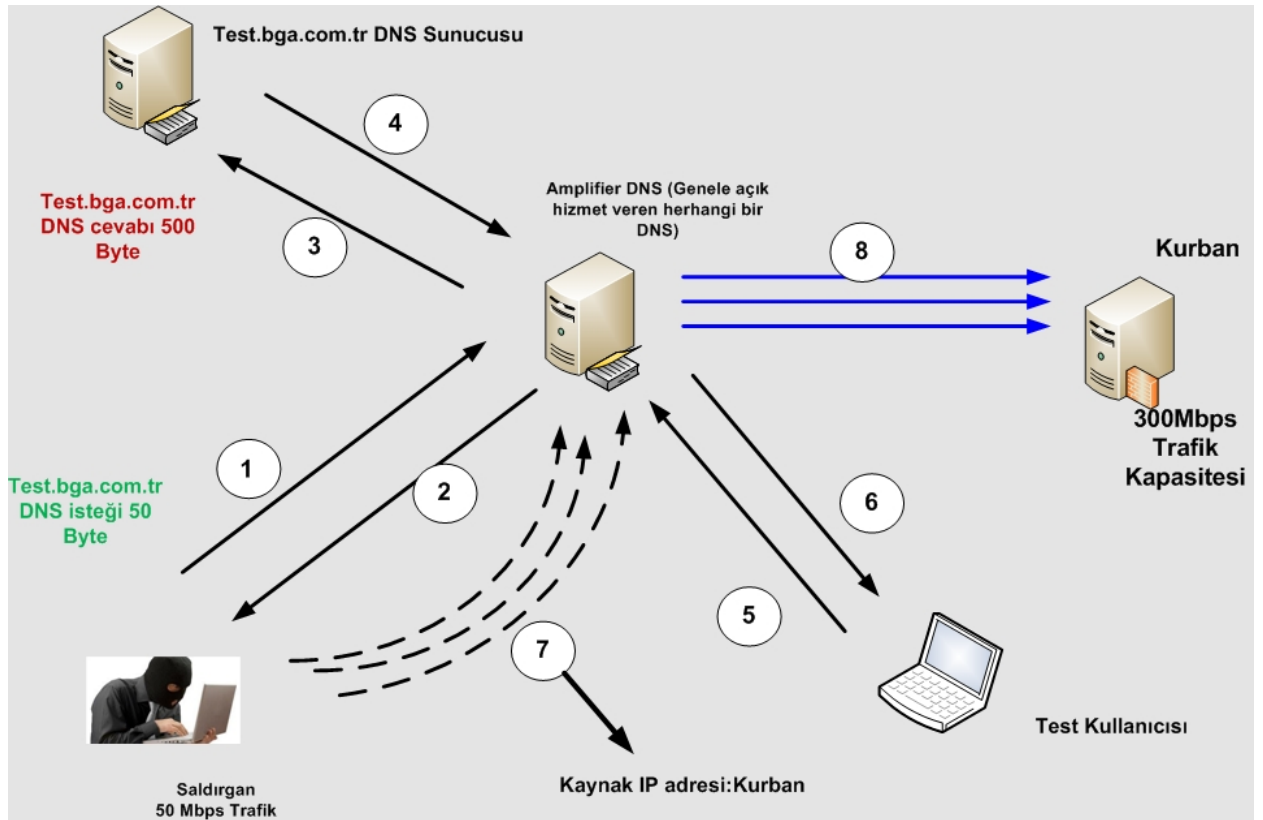
Basit bir script ile anlık olarak DNS sunucunun cevap performansı ölçülebilir.

```
while true; do dig . @ns1.tr.net|grep "Query time:";sleep 2;done

;; Query time: 11 msec
;; Query time: 11 msec
;; Query time: 10 msec
;; Query time: 11 msec
;; Query time: 13 msec
;; Query time: 11 msec
;; Query time: 398 msec          → Saldırı altındaki DNS sunucu
cevabı
```

Amplified DNS DoS Saldırıları

Bu saldırı tipinde gönderilen DNS isteğine dönecek cevabın kat kat fazla olması özelliğini kullanır. Sisteme gönderilecek 50 byte'lık bir DNS isteğine 500 Byte~cevap döndüğü düşünülürse saldırgan elindeki bant genişliğinin 10 katı kadar saldırı trafiği oluşturabilir.



Adım Adım DNS Amplification DoS Saldırısı

1. Saldırgan rekursif sorguya açık DNS sunucu bulur ve daha önce hazırladığı özel alan adını sorgular (Gerçek hayatta özel bir alan adı değil "." sorgulanır.). Bu isteğin boyutu 50 Byte tutmaktadır.
2. DNS sunucu kendi ön belleğinde olmayan bu isteği gidip ana DNS sunucuya sorar (50 Byte)
3. Ana DNS sunucu test.bga.com.tr için gerekli cevabı döner (500~byte)

4. Ara DNS sunucu cevabı ön belleğine alarak bir kopyasını Saldırgana döner. Burada amaç ARA DNS sunucunun dönen 500 Byte'lık cevabı ön belleğe almasını sağlamaktır.
5. Test kullanıcısı (saldırganın kontrolünde) test.bga.com.tr alan adını sorgular ve cevabın cache'de olup olmadığını anlamaya çalışır.
6. Ara DNS sunucu ön belleğinden 500 byte cevap döner
7. Saldırgan Kurban'ın IP adresinden geliyormuş gibi sahte DNS paketleri gönderir. DNS paketleri test.bga.com.tr'i sorgulamaktadır (ortalama 100.000 dns q/s). Bu üretilen paketlerin Saldırgana maliyeti 100.000 X53 Byte
8. Ara DNS sunucu gelen her paket için 500 Byte'lık cevabı Kurban sistemlere dönmeye çalışacaktır. Böylece Ara DNS sunucu 100.000X500 Byte trafik üreterek saldırırganın kendi trafiğinin 10 katı kadar çoğaltarak Kurban'a saldırıyor gözükcektir.

```
$ dig . @ns1.tr.net

; <<>> DiG 9.7.0-P1 <<>> . @ns1.tr.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27323
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 14
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;      IN      A

;; AUTHORITY SECTION:
.      512544 IN      NS      k.root-servers.net.
```

```

.          512544 IN    NS    l.root-servers.net.
.          512544 IN    NS    m.root-servers.net.
.          512544 IN    NS    a.root-servers.net.
.          512544 IN    NS    b.root-servers.net.
.          512544 IN    NS    c.root-servers.net.
.          512544 IN    NS    d.root-servers.net.
.          512544 IN    NS    e.root-servers.net.
.          512544 IN    NS    f.root-servers.net.
.          512544 IN    NS    g.root-servers.net.
.          512544 IN    NS    h.root-servers.net.
.          512544 IN    NS    i.root-servers.net.
.          512544 IN    NS    j.root-servers.net.

```

;; ADDITIONAL SECTION:

```

a.root-servers.net. 598944 IN    A      198.41.0.4
a.root-servers.net. 598944 IN    AAAA   2001:503:ba3e::2:30
b.root-servers.net. 598944 IN    A      192.228.79.201
c.root-servers.net. 598944 IN    A      192.33.4.12
d.root-servers.net. 598944 IN    A      128.8.10.90
d.root-servers.net. 598944 IN    AAAA   2001:500:2d::d
e.root-servers.net. 598944 IN    A      192.203.230.10
f.root-servers.net. 598944 IN    A      192.5.5.241
f.root-servers.net. 598944 IN    AAAA   2001:500:2f::f
g.root-servers.net. 598944 IN    A      192.112.36.4
h.root-servers.net. 598944 IN    A      128.63.2.53
h.root-servers.net. 598944 IN    AAAA   2001:500:1::803f:235
i.root-servers.net. 598944 IN    A      192.36.148.17
i.root-servers.net. 598944 IN    AAAA   2001:7fe::53

```

```
:: Query time: 14 msec
:: SERVER: 195.155.1.3#53(195.155.1.3)
:: WHEN: Mon Jan 23 13:52:52 2012
:: MSG SIZE rcvd: 512
```

Örnek DNS Amplified DoS Saldırısı

DoS yapılacak Hedef Sistem: kurban.example.com (V)

Aracı olarak kullanılacak DNS sunucu dns-sunucu.example.com (A)

Saldırgan (C)

```
./amfdns -a dns-sunucu.example.com -t A -q . -target kurban.example.com
```

DNS Flood DDoS Saldırılarını Yakalama

<http://www.adotout.com/dnsflood.html> yazılımı kullanılabilir.

```
root@bt:~/dns_flood_detector# dns_flood_detector -i eth0 -t 100 -v -b
[22:16:17] source [85.95.238.172] - 0 qps tcp : 419 qps udp
[22:16:27] source [85.95.238.172] - 0 qps tcp : 139 qps udp
```

DNS Flood DDoS Saldırılarını Engelleme

DNS Flood saldırılarını engellemek için kullanılan temel yöntemler:

- DNS Caching
- Dns anycast
- Rate limiting
- DFAS

Rate Limiting Yöntemi

Rate limiting yöntemi ile belirli ip adreslerinden yapılacak UDP/DNS flood saldırılarında kaynak ip adresi engellemesi amaçlanır. Ama UDP tabanlı protokollerde kaynak ip adresinin gerçek olup olmadığını anlamak çok zor olduğu için genellikle işe yaramaz bir yöntemdir.

Bu yöntemi kullanan bir hedefe doğru saldırgan istediği ip adresinden geliyormuş gibi paketler göndererek istediği ip adresinin engellenmesini sağlayabilir (Türkiye ip bloklarından paket göndermek gibi)

DFAS

TCP üzerinden gerçekleştirilecek olan DDoS saldırılarını engellemek göreceli olarak daha kolaydır diyebiliriz. Bunun temel nedeni TCP üzerinden yapılacak saldırılarda saldırganın gerçek ip adresle mi yoksa sahte adresle mi saldırıp saldırmadığının anlaşabiliyor olmasıdır(basit mantık 3'lü el sıkışmayı tamamlıyorsa ip gerçektir).

UDP üzerinden gerçekleştirilecek DDoS saldırılarını (udp flood, dns flood vs)engellemek saldırı gerçekleştiren ip adreslerinin gerçek olup olmadığını anlamamanın kesin bir yolu olmadığı için zordur. UDP kullanarak gerçekleştirilen saldırılarda genellikle davranışsal engelleme yöntemleri ve ilk paketi engelle ikinci paketi kabul et(dfas) gibi bir yöntem kullanılır.

DFAS Yönteminin Temeli

TCP ya da UDP ilk gelen paket için cevap verme aynı paket tekrar gelirse pakete uygun cevap ver ve ilgili ip adresine ait oturumu tutmaya başla veya ilk pakete hatalı cevap dön (sıra numarası yanlış SYN-ACK) ve karşı taraftan RST gelmesini bekle.

Ardından istemcinin gönderdiği TCP isteğine DDoS engelleme sistemi tarafından hatalı bir cevap dönülerek karşı taraftan RST paketi bekleniyor ve RST paketi alındıktan sonra ip adresinin gerçek olduğu belirlenerek paketlere izin veriliyor.

```
[root@netdos1 ~]# tcpdump -i em0 -tn host 5.6.7.8
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 96 bytes
IP 1.2.3.4.19399 > 5.6.7.8.53: 8818+ A? www.example.com (37)
IP 5.6.7.8.53 > 1.2.3.4.19399: 8818*| 0/0/0 (37)
IP 1.2.3.4.34096 > 5.6.7.8.53: Flags [S], seq 3183103590, win 65535, options [mss 1460,nop,wscale 3,sackOK,TS val 4045396826 ecr 0],length 0
IP 5.6.7.8.53 > 1.2.3.4.34096: Flags [S.], seq 4110155774, ack 3060256364, win 65535, options [mss 1460,nop,wscale 3,sackOK,TS val4045396826 ecr 0], length 0
IP 1.2.3.4.34096 > 5.6.7.8.53: Flags [R], seq 3060256364, win 0, length 0
```

```
IP 1.2.3.4.34096 > 5.6.7.8.53: Flags [S], seq 3183103590, win 65535, options [mss 1460,nop,wscale 3,sackOK,TS val 4045399827 ecr 0],length 0
IP 5.6.7.8.53 > 1.2.3.4.34096: Flags [R.], seq 184811522, ack 122847228, win 0, length 0
```

DFAS yöntemi gelen giden tüm paketler için değil saldırı anında ilk paketler için gerçekleştirilir.

Saldırı Anında Sistemin DNS İsteklerine Döndüğü Cevap:

```
IP 1.2.3.4.51798 > 5.6.7.8.53: 53698+ A? www.example.com. (37)
IP 5.6.7.8.53 > 1.2.3.4.51798: 53698 ServFail- 0/0/0 (37)
IP 1.2.3.4.34623 > 5.6.7.8.53: 61218+ A? www.example.com (37)
IP 5.6.7.8.53 > 1.2.3.4.34623: 61218*- 1/0/0 A 1.21.2.72 (53)
```

Örnek:

Bir müddet aşağıdaki gibi udp flood (dns portundan) gerçekleştirdikten sonra

```
hping --flood -p 53 --udp hedef_dns
```

```
root@bt:~# tcpdump -i eth0 -tn udp port 53 -v
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
IP (tos 0x0, ttl 64, id 5558, offset 0, flags [none], proto UDP (17), length 65)
  85.95.238.172.51518 > 1.2.227.77.53: 37837+ A? www.example.com. (37)
IP (tos 0x0, ttl 119, id 5558, offset 0, flags [none], proto UDP (17), length 65)
  1.2.227.77.53 > 85.95.238.172.51518: 37837*| 0/0/0 (37)

IP (tos 0x0, ttl 64, id 5559, offset 0, flags [none], proto UDP (17), length 65)
  85.95.238.172.44470 > 1.2.227.77.53: 29161+ A? www.example.com. (37)
```

```
IP (tos 0x0, ttl 119, id 5559, offset 0, flags [none], proto UDP (17), length 65)
  1.2.227.77.53 > 85.95.238.172.44470: 29161*| 0/0/0 (37)
```

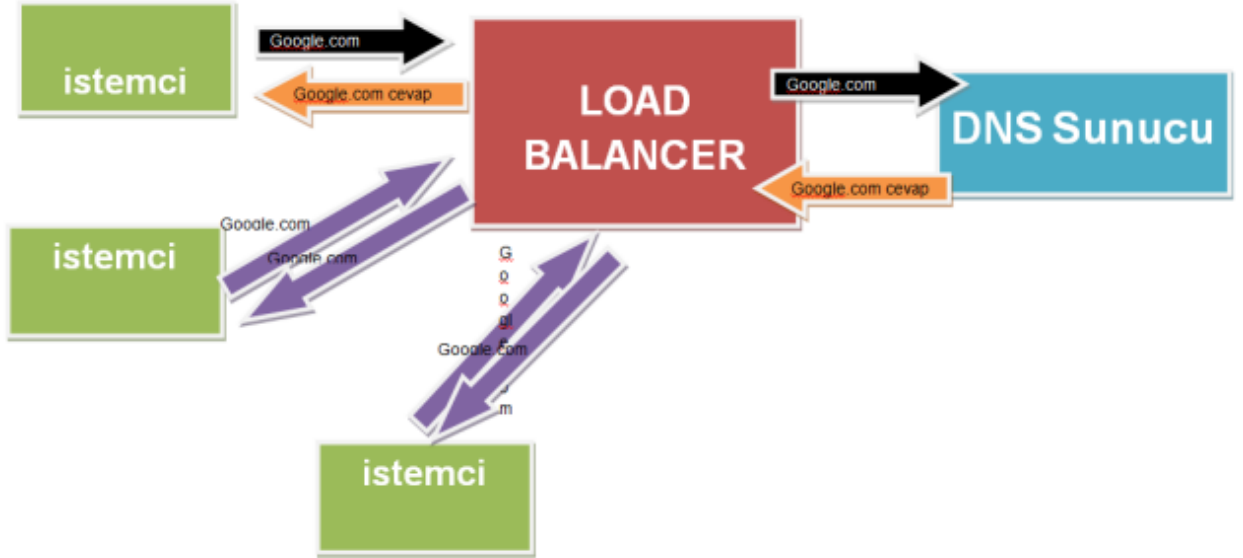
Aşağıdaki gibi sorgulamalar TCP DNS'e yönlendirilmektedir.

```
root@bt:~# dig www.example.com @1.2.227.77
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.7.0-P1 <<>> www.example.com @1.2.227.77
;; global options: +cmd
;; connection timed out; no servers could be reached
```

DNS Caching Cihazlarını Atlatma Saldırıları

Caching cihazları aynı tipte gelen sorgulamalar için caching işlemi yapabilmektedir ve yoğun saldırılarda DNS sunucuların en az seviyede etkilenmesini sağlamaktadır.



DNS flood saldırılarında gönderilen tüm DNS isteklerindeki alan adlarını rastgele seçilirse caching cihazları gelen tüm istekleri gerçek DNS sunuculara yönlendirecektir.

Eğer test yapılan DNS sunucu authoritative (yetkili) tipte sunucu ise rastgele domainler için yapılacak sorgulamalara cevap dönülmeyecektir. Bu tip sunuculara karşı hedef DNS sunucuda tutulan herhangi bir alan adının alt alan adlarına(rastgele üretilmiş) yönelik paketlerin gönderilmesi DNS sunucunun performansını etkileyecektir.

DDoS Saldırı Analizi ve Adli Bilişim İncelemesi

DDoS saldırılarında dikkate alınması gereken iki temel husus vardır. İlki saldırıyı engelleme ikincisi saldırının kim tarafından ne şiddette ve hangi yöntemler, araçlar kullanılarak yapıldığının belirlenmesidir.

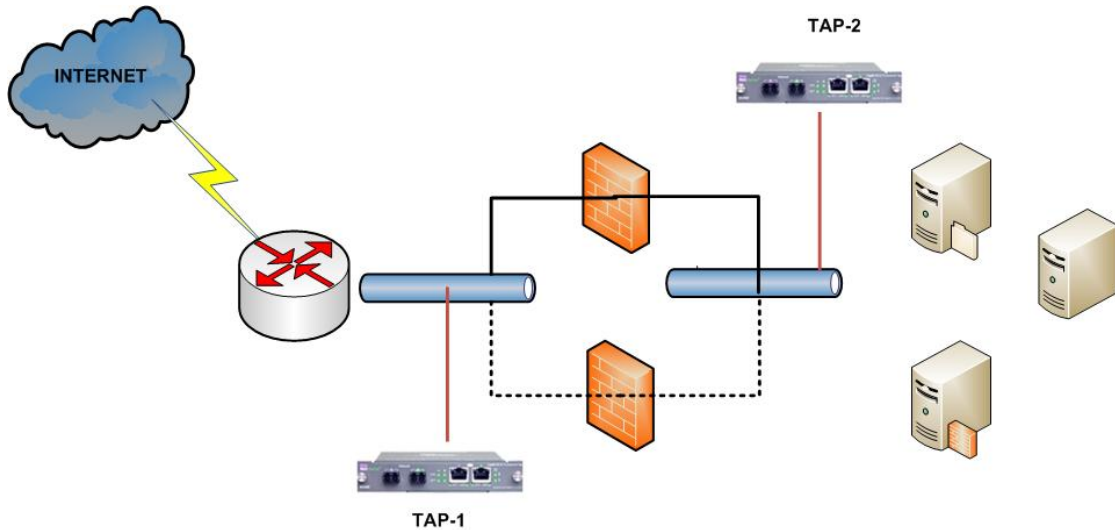
Genellikle saldırı engelleme kısmı dikkate alınmaktadır ve plansız bir şekilde DDoS saldırıları anlık olarak durdurulmaya çalışılmaktadır. Oysa yapılan araştırmalar göstermiştir ki bir kere DDoS saldırısına maruz kalıp yenilen bir kurum/sistem aynı yıl içerisinde defalarca DDoS saldırısına maruz kalmıştır.

Yapılması gereken hem saldırının acilen “planlı” bir şekilde durdurulması, engellenmesi hem de saldırı sonrası analiz için kullanılacak delillerin toplanmasıdır.

DDoS Analizi İçin Gerekli Yapının Kurulması

DDoS saldırısı esnasında çok basit işlemlerle toplanacak deliller saldırı sonrası analizlerde oldukça yardımcı olacaktır. Saldırının hangi şiddette, hangi protokoller kullanılarak (TCP, UDP, ICMP, HTTP, SMTP vs) ne tip (packet flood, bandwith aşırma) ve kimler (gerçek ip adresleri, spoof edilmiş ip adresleri, botnet kullanımı) tarafından gerçekleştirildiği vs.

DDoS Saldırılarındaki sağlıklı analiz yapabilmek için uygun yerlere TAP cihazları yerleştirilmelidir. Bu cihazlar aracılığıyla saldırı anında aktif sistemleri etkilemeden log toplama imkânı olacaktır.



Saldırı Analizinde Cevabı Aranılan Sorular

Herhangi bir konuda analize başlamadan yapılması gereken ilk iş konuyla ilgili sorulabilecek soruları çıkarmak ve analizi bu sorulara göre planlamak olmalıdır. DDoS saldırı analizi yaparken aynı yöntemi uygulayarak sağlıklı sonuçlar elde edilebilir. Bu yazıda cevabını aradığımız sorular:

- Gerçekten bir DDoS saldırısı var mı?
- Varsa nasıl anlaşılır?
- DDoS saldırısının tipi nedir?
- DDoS saldırısının şiddeti nedir?
- Saldırı ne kadar sürmüştü?
- DDoS saldırısında gerçek IP adresleri mi spoofed IP adresleri mi kullanılmış?
- DDoS saldırısı hangi ülke/ülkelerden geliyor?

Alet Çantasında Bulunması Gereken Araçlar

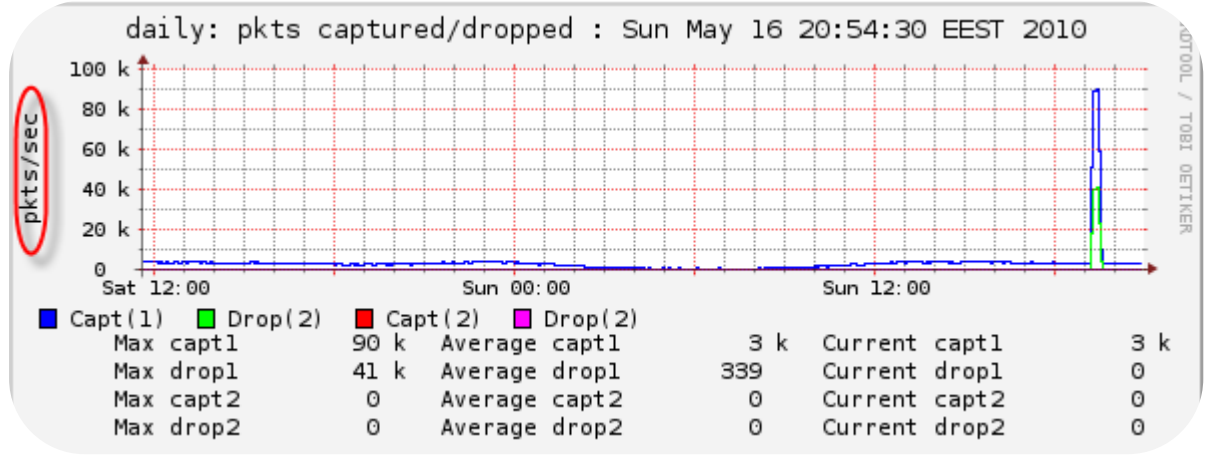
DDoS saldırı analizine başlamadan elimizin altında bulunması gereken çeşitli araçlar vardır. Bu yazıda DDoS analizi için kullanılan tüm araçlar internet üzerinden ücretsiz edinilebilecek açık kaynak kodlu yazılımlardır.

Tcpstat, tcpdstat, tcptrace tcpdump, ourmon, argus, urlsnarf, snort, aguri, cut, grep, awk, wc...

DDoS Saldırı Tespit Sistemleri

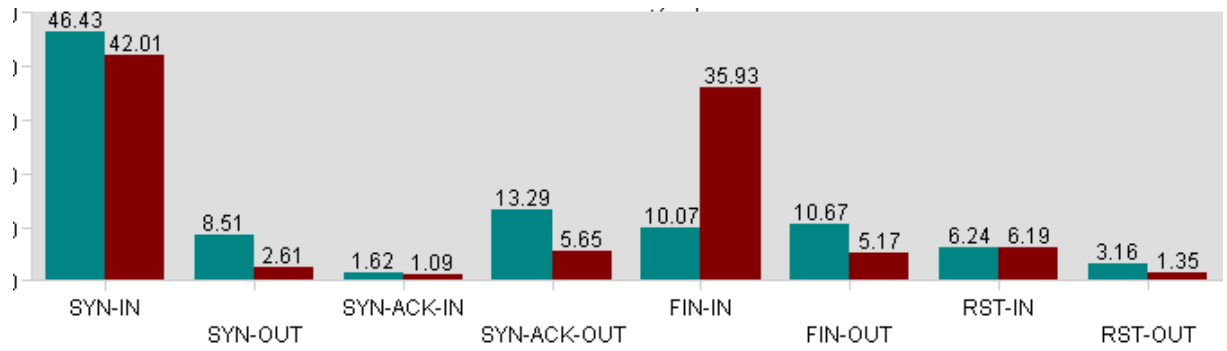
İhtiyacımız DDoS saldırılarını en kısa sürede belirlemek ve herhangi bir DDoS saldırısı esnasında saldırıya ait tüm paketleri loglayacak bir sistemdir. İnternet üzerinden ücretsiz edinilebilecek açık kod ADS sistemi olan Ourmon, DDoS saldırılarını belirleme amaçlı kullanılabilir. Benzer şekilde tcpstat aracı da sistemdeki paket anormalliklerini tespit etmek ve saldırı anında otomatik paket kuyruğuna başlamak için kullanılabilir.

Resim-2’de Ourmon arabiriminden alınan çıktıda net bir şekilde DDoS saldırısı gözükmektedir. Ortalama 10.000 ler seviyesinde seyreden PPS(Packet Per Second) değeri aniden 90.000ler seviyesine çıkmıştır.



Resim-2

Resim-3 McAfee Intrushield IPS sisteminin paket anormalliğini gösteren bileşeninden alınmıştır. Bu bileşen kullanılarak saldırılar rahatlıkla farkedilebilir.



Resim-3

DDoS Saldırılarında Delil Toplama

DDoS saldırılarında sonradan incelenmek üzere paketler kaydedilmelidir. Bunun için kaydedilen trafik miktarına bağlı olarak ciddi sistemlere(CPU, RAM, Disk alanı bakımından) ihtiyaç olabilir.

Dikkat edilmesi gereken en önemli husus paket kaydetme işleminin kesinlikle aktif cihazlar tarafından (IPS, DDoS engelleme Sistemi, Firewall) yapılmaması gerektiğidir. Bunun nedeni açıktır. DDoS esnasında aktif sistemler zaten normalin üzerinde bir yoğunluğa sahiptir ve gelen-giden paketleri kaydetmek için ek performansilemci gücü bulamayabilir. Daha da kötüsü aktif sistemler paket kaydetmeye çalışırken asıl işlevi olan engelleme işlemini gerçekleştiremeyebilir.

Eğer DDoS saldırı engelleme sistemi kısa sürede saldırının tipini anlayabildiyse ve eğer saldırı uygulama seviyesi bir protocol kullanılarak

gerçekleştirildiyse(HTTP GET Flood) sadece paket başlık bilgilerini kaydetmek yeterli olmayacaktır, tüm protocol bilgileri(+payload) kaydedilmesi gerekir.

Eğer saldırı SYN flood, ACK flood, UDP flood gibi sadece paket başlık bilgilerini kullanarak gerçekleştirilmişse payload bilgisinin kaydedilmesi gerekmeyecektir.

Paket Kaydetme

Paket kaydetme için Linux/FreeBSD üzerinde tcpdump en uygun seçenektir. 10 Gb ortamlarda klasik libpcap yerine alternative kütüphaneler tercih edilmelidir.

Tcpdump ile paket kaydetme

```
#tcpdump -n -w ddostest1.pcap
```

Eğer payload bilgisi de gerekliyse tcpdump'a -s0 parametresi de eklenmelidir. Kayıt esnasında tek bir dosya değil de farklı farklı dosyalara kayıt yapılması istenirse -C parametresi incelenmelidir.

DDoS Saldırı Tipi Belirleme

DDoS saldırı tipini belirlemek için saldırı esnasında kaydedilen paket dosyalarını kullanılacaktır.

Saldırı tipi belirlemede ilk olarak hangi protokol ne kadar istek almış bilgisine ihtiyaç duyulur. Bu bilgi sonrasında DDoS saldırısının tipi hakkındaki ilk bilgi ortaya çıkacaktır.

Tcpdstat kullanılarak pcap dosyalarında(saldırı esnasındaki kayıt dosyaları) hangi protocol ne oranda kullanılmış bilgisi aşağıdaki gibi alınabilir.

```
# tcpdstat -n ddos.pcap
```

```
DumpFile: ddos.pcap
FileSize: 45.58MB
Id: 201005181114
StartTime: Tue May 18 11:14:57 2010
EndTime: Tue May 18 11:16:19 2010
TotalTime: 81.59 seconds
TotalCapSize: 37.38MB CapLen: 96 bytes
# of packets: 537187 (170.55MB)
```


AvgRate: 17.55Mbps stddev:7.87M

Packet Size Distribution (including MAC headers)

<<<<

[32- 63]: 337610
 [64- 127]: 13257
 [128- 255]: 5341
 [256- 511]: 19289
 [512- 1023]: 104016
 [1024- 2047]: 57674

>>>>

Protocol Breakdown

<<<<

protocol	packets	bytes	bytes/pkt
[0] total	537187 (100.00%)	178836761 (100.00%)	332.91
[1] ip	537082 (99.98%)	178830375 (100.00%)	332.97
[2] tcp	529590 (98.59%)	178126550 (99.60%)	336.35
[3] http(s)	169318 (31.52%)	123244600 (68.91%)	727.89
[3] http(c)	113553 (21.14%)	34132760 (19.09%)	300.59
[3] squid	9 (0.00%)	540 (0.00%)	60.00
[3] smtp	238109 (44.33%)	14288975 (7.99%)	60.01
[3] nntp	3 (0.00%)	180 (0.00%)	60.00
[3] ftp	24 (0.00%)	1510 (0.00%)	62.92
[3] pop3	6 (0.00%)	360 (0.00%)	60.00
[3] imap	1 (0.00%)	60 (0.00%)	60.00
[3] telnet	7 (0.00%)	448 (0.00%)	64.00
[3] ssh	4 (0.00%)	366 (0.00%)	91.50
[3] dns	4 (0.00%)	240 (0.00%)	60.00
[3] bgp	4 (0.00%)	240 (0.00%)	60.00
[3] napster	5 (0.00%)	300 (0.00%)	60.00
[3] realaud	3 (0.00%)	180 (0.00%)	60.00
[3] rtsp	7 (0.00%)	420 (0.00%)	60.00
[3] icecast	5 (0.00%)	300 (0.00%)	60.00
[3] hotline	3 (0.00%)	180 (0.00%)	60.00
[3] other	8525 (1.59%)	6454891 (3.61%)	757.17
[2] udp	7478 (1.39%)	702824 (0.39%)	93.99
[3] dns	268 (0.05%)	32774 (0.02%)	122.29
[3] other	7210 (1.34%)	670050 (0.37%)	92.93
[2] icmp	14 (0.00%)	1001 (0.00%)	71.50
[1] ip6	1 (0.00%)	146 (0.00%)	146.00
[2] udp6	1 (0.00%)	146 (0.00%)	146.00
[3] other	1 (0.00%)	146 (0.00%)	146.00

>>>>

Çıktıda dikkatimizi aşağıdaki satırlar çekmekte.

[2] tcp 529590 (98.59%) 178126550 (99.60%) 336.35

[3] smtp 238109 (44.33%) 14288975 (7.99%) 60.01

Bu satırlara bakarak şu yorum yapılabilir: Saldırı TCP kullanılarak gerçekleştirilmiş ve hedef port SMTP'dir.

Eğer çıktı aşağıdaki gibi olsaydı: (%99 oranında UDP) rahatlıkla saldırının TCP tabanlı değil UDP tabanlı olduğu yorumu yapılabilirdi.

tcpdstat -n ddos1.pcap

DumpFile: ddos1.pcap

FileSize: 0.36MB

Id: 201005181127

StartTime: Tue May 18 11:27:53 2010

EndTime: Tue May 18 11:28:20 2010

TotalTime: 27.78 seconds

TotalCapSize: 0.30MB CapLen: 96 bytes

of packets: 3464 (320.10KB)

AvgRate: 91.67Kbps stddev:50.34K

Packet Size Distribution (including MAC headers)

<<<<

[64- 127]: 3362

[128- 255]: 89

[256- 511]: 13

>>>>

Protocol Breakdown

<<<<

protocol	packets	bytes	bytes/pkt

[0] total	3464 (100.00%)	327782 (100.00%)	94.63

[1] ip	3462 (99.94%)	327490 (99.91%)	94.60
[2] udp	3462 (99.94%)	327490 (99.91%)	94.60
[3] dns	106 (3.06%)	12378 (3.78%)	116.77
[3] other	3356 (96.88%)	315112 (96.13%)	93.90
[1] ip6	2 (0.06%)	292 (0.09%)	146.00
[2] udp6	2 (0.06%)	292 (0.09%)	146.00
[3] other	2 (0.06%)	292 (0.09%)	146.00

Saldırının hangi protocol (TCP/UDP/ICMP) kullanılarak gerçekleştirildiği bilgisi elde edildikten sonraki aşama gerçekte hangi saldırı yönteminin kullanıldığını bulmak olacaktır. Eğer UDP flood ise doğrudan kaynak IP adresi inceleme gerçekleştirilebilir fakat TCP kullanıldıysa işin seyri biraz değişecektir.

TCP kullanılarak gerçekleştirilen DDoS saldırı çeşitlerinden en sık tercih edilen ikili SYN Flood ve HTTP GET flood'dur.

TCP Bayrakları Kullanılarak Gerçekleştirilen DDoS Saldırıları

SYN Flood Saldırısı Analizi

Tcpdump aracının özellikleri kullanılarak trafik içerisinde sadece SYN bayrağı taşıyan paketler ayıklanabilir.

Sadece SYN bayraklı paketleri yakalama

```
# tcpdump -r ddos.pcap -n 'tcp[tcpflags] & tcp-syn == tcp-syn'
22:04:22.809998 IP 91.3.119.80.59204 > 11.22.33.44.53: Flags [S], seq
2861145144, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:22.863997 IP 91.3.119.80.59135 > 82.8.86.175.25: Flags [S], seq
539301671, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:22.864007 IP 91.3.119.80.59205 > 11.22.33.44.53: Flags [S], seq
4202405882, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:23.033997 IP 91.3.119.80.64170 > 11.22.33.44.53: Flags [S], seq
1040357906, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:23.146001 IP 91.3.119.80.59170 > 11.22.33.44.53: Flags [S], seq
3560482792, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:23.164997 IP 91.3.119.80.59171 > 20.17.222.88.25: Flags [S], seq
```

```
1663706635, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:23.384994 IP 91.3.119.80.59136 > 11.22.33.44.53: Flags [S], seq
192522881, win 65535, options [mss 1460,sackOK,eol], length 0
22:04:23.432994 IP 91.3.119.80.59137 > 11.22.33.44.53: Flags [S], seq
914731000, win 65535, options [mss 1460,sackOK,eol], length 0
```

ya da aynı işi yapan 'tcp[13] & 2 != 0' parametresi kullanılabilir.

Eğer saldırı klasik syn flood değilse alternatif flagleri deneyerek benzer sonuçlar elde edilebilir.

ACK Flood Analizi

Tcpdump kullanarak ACK bayraklı paketleri ayıklama

```
# tcpdump -i bce1 -n 'tcp[13] & 16 != 0'
```

FIN Flood Analizi

Tcpdump kullanarak FIN bayraklı paketleri ayıklama

```
# tcpdump -i bce1 -n 'tcp[13] & 1 != 0' and tcp port 80
```

tcp[13] demek TCP başlığındaki 13. byte anlamına gelir. Bu da bayrakları temsil eden byte'dır. Her bayrak için verilecek değer aşağıdaki resimden alınabilir.

SYN	ACK	FIN	RST	PUSH	URG	SYN+ACK
2	16	1	4	8	32	2+16=18

HTTP GET Flood Saldırısı

TCP paketleri içerisindeki GET komutlarının tcpdump ile ayıklanabilmesi için kullanılması gereken parametreler.

```
#tcpdump -n -r ddos3.pcap tcp port 80 and \( tcp[20:2] = 18225 \)
```

Saldırının Şiddetini Belirleme

DDoS saldırı analizine başlarken cevaplamaya çalıştığımız sorulardan biri de saldırının şiddetiydi. Saldırının şiddetini iki şekilde tanımlayabiliriz

1. Gelen trafiğin ne kadar bant genişliği harcadığı
2. Gelen trafiğin PPS değeri

Tcpstat aracı kullanılarak trafik dosyaları üzerinde saldırının PPS değeri, ne kadar bantgenişliği harcadığı bilgileri detaylı olarak belirlenebilir.

```
# tcpstat -r ddos_analizi.pcap -o "Byte/s:%B MinPacketSize:%m PPS:%p
TCP:%T UDP:%U \n" 5

Byte/s:3401176.20 MinPacketSize:40 PPS:5929.20 TCP:29004 UDP:637
Byte/s:3145824.60 MinPacketSize:40 PPS:5247.60 TCP:25797 UDP:436
Byte/s:3140760.20 MinPacketSize:40 PPS:5252.40 TCP:25661 UDP:594
Byte/s:3850993.20 MinPacketSize:40 PPS:13602.80 TCP:66808 UDP:756
Byte/s:4360434.30 MinPacketSize:40 PPS:14904.80 TCP:73681 UDP:435

Byte/s:4460434.40 MinPacketSize:40 PPS:14874.80 TCP:73681 UDP:457
Byte/s:4960434.60 MinPacketSize:40 PPS:13904.80 TCP:73681 UDP:535
Byte/s:5460434.20 MinPacketSize:40 PPS:24904.80 TCP:73681 UDP:456
```

Saldırı Kaynağını Belirleme

DDoS saldırılarında en önemli sorunlardan biri saldırıyı gerçekleştiren asıl kaynağın bulunamamasıdır. Bunun temel sebepleri saldırıyı gerçekleştirenlerin zombie sistemler kullanarak kendilerini saklamaları ve bazı saldırı tiplerinde gerçek IP adresleri yerine spoof edilmiş IP adreslerinin kullanılmasıdır.

Saldırı analizinde saldırıda kullanılan IP adreslerinin gerçek IP'ler mi yoksa spoofed IP'ler mi olduğu rahatlıkla anlaşılabilir.

İnternet üzerinde sık kullanılan DDoS araçları incelendiğinde IP spoofing seçeneği aktif kullanılırsa random üretilmiş sahte IP adreslerinden tek bir paket gönderildiği görülecektir. Yani saldırı sırasında kaydedilen dosya incelendiğinde fazla sayıda tek bağlantı gözüküyorsa saldırının spoof edilmiş IP adresleri kullanılarak gerçekleştirildiği hükmüne varılabilir.

Tek cümleyle özetleyecek olursak: **Eğer aynı IPden birden fazla bağlantı yoksa spoofed IP kullanılmış olma ihtimali yüksektir.**

```
#tcpdump -n -r ddos.pcap |awk -F" " '{print $3}'|cut -f1,2,3,4 -d"."|sort -
```

n|uniq -c

1 6.65.194.168
1 6.65.208.248
1 6.65.226.233
1 6.65.232.125
1 6.65.235.140
1 6.65.248.199
1 6.65.249.104
1 6.65.32.97
1 6.65.44.199
1 6.65.48.49
1 6.65.62.221
1 6.65.62.30
1 37.83.136.81
1 37.83.14.12
1 37.83.152.203
1 37.83.164.223
1 37.83.165.146
1 37.83.166.132
1 37.83.185.89
1 37.83.194.21
1 62.185.46.86
1 62.185.60.100
1 62.185.64.248
1 62.185.66.32
1 62.185.75.23

```
1 62.185.9.193
1 62.185.92.77
1 62.185.96.16
```

Yukarıdaki tcpdump komutu saldırı yapan IP adreslerini ve ilgili IP adresinden saldırı boyunca kaç adet paket gönderildiğini bulmaya yarar. Çıktıdan da görüleceği üzere yoğun şekilde spoofed IP kullanılmıştır.

Saldırıda Kullanılan Top 10 IP Adresi

Saldırıda kullanılan ve en fazla paket gönderen 10 ip adresine ulaşmak istenirse aşağıdaki komut satırı iş görecektir.

```
# 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
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
```

Çıktıda sol taraf IP adresi, sağ taraf ise ilgili IP adresinden saldırı boyunca kaç adet paket gönderildiğidir.

HTTP GET Flood Saldırısında Kullanılan IP Adresleri

HTTP GET flood saldırılarında IP spoofing yapmak mümkün değildir. Bir system HTTP isteği gönderebilmesi için öncelikli olarak 3'lü el sıkışmasını tamamlaması gerekmektedir. Günümüz işletim sistemi/ağ/güvenlik cihazlarında 3'lü el sıkışma esnasında TCP protokolünü kandırarak IP spoofing yapmak mümkün gözükmemektedir. Dolayısıyla HTTP GET flood saldırıları analizinde saldırı yapan IP adresleri %99 gerçek IP adreslerdir.

```
# tcpdump -n -r ddos3.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
```

reading from file ddos3.pcap, link-type EN10MB (Ethernet)

1092 62.202.27.120

92 62.111.223.1

7 62.227.26.27

52000 62.227.33.111

63 62.72.23.102

1300 66.229.63.26

2 67.193.112.72

1 77.77.31.226

31020 77.160.72.77

93 77.161.12.233

71 77.161.227.192

90232 77.161.32.210

23 77.162.1.137

2 77.162.3.170

12900 77.162.76.177

21 77.163.6.127

3 77.163.132.37

79100 77.163.217.137

21 77.165.97.107

9 77.166.197.232

2700 77.166.60.175

35100 77.166.65.133

74200 77.167.126.119

22009 77.169.152.239

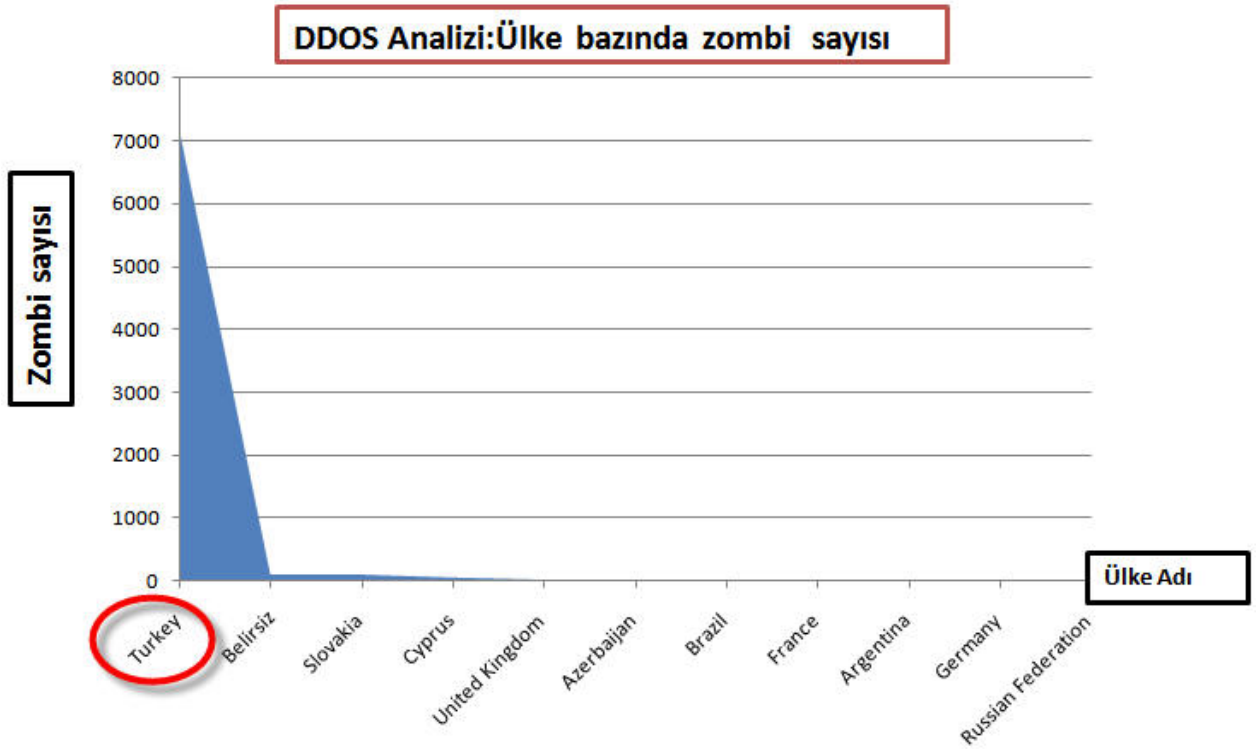
11891 77.171.175.77

Sağ taraf IP adresi, sol taraftaki sayı da ilgili IP adresinden kaç adet HTTP GET Flood isteği gönderildiğidir.

Saldırıda Kullanılan IP Adresleri Hangi Ülkeden?

Gerçekleştirilen saldırı bir botnet aracılığıyla gerçekleştirilmiş ve IP adresleri spoof edilmemişse saldırıda kullanılan IP adreslerinin hangi ülkelere ait olduğu bulunabilir.

Çıkan sonuç grafiğe döküldüğünde aşağıdaki çıktı alınacaktır.



Saldırı Paketlerini Pasif Snort Sisteminden Geçirme

Snort açık kaynak kodlu bir IPS sistemidir ve bünyesinde barındırdığı saldırı imzalarıyla çoğu klasik DDoS aracını /tipini tanımaktadır. Saldırı esnasında kaydedilen paketler Snort'un pasif IPS motorundan geçirilirse hangi saldırı tipleri/araçları kullanılmış bilgisi alınabilir.

```
#snort -r pids.pcap -c /usr/local/etc/snort/snort.conf -q -O
```

```
Jun  9 12:15:37 netdos1 snort: [1:2000545:6] ET SCAN NMAP -f -sS  
[Classification: Attempted Information Leak] [Priority: 2]: {TCP} 0.0.0.0:45295 -  
> 0.0.0.0:80
```

```
Jun  9 12:15:37 netdos1 snort: [1:2000545:6] ET SCAN NMAP -f -sS  
[Classification: Attempted Information Leak] [Priority: 2]: {TCP} 0.0.0.0:45296 -  
> 0.0.0.0:707
```

```
Jun  9 12:15:40 netdos1 snort: [1:408:5] ICMP Echo Request Flood  
[Classification: Misc activity] [Priority: 3]: {ICMP} 0.0.0.0 -> 0.0.0.0
```

```
Jun  1 10:15:23 netdos1 snort: [1:1000003:6] SYN Flood [Classification:  
DDoS] [Priority: 3]: {TCP} 0.0.0.0:1024 -> 0.0.0.0:80
```

```
Jun  9 12:15:37 netdos1 snort: [1:1000002:6] HTTP GET FLOOD  
[Classification: DDoS] [Priority: 2]: {TCP} 0.0.0.0:15295 -> 0.0.0.0:80
```

NTP Servisi Kullanarak Gerçekleştirilen Amplification DDoS Saldırıları

DDoS saldırıları her geçen gün önemi artırıyor ve yeni yeni yöntemler, teknikler keşfediliyor. Son zamanlarda kullanılan yöntemler standart araç tabanlı yöntemlerden oldukça farklı, arka planı düşünülmüş, tasarlanmış ve yüksek boyutta olmaktadır. Yeni olarak nitelendirilse de teknik olarak daha önceden bilinen, teorik olarak dökümanite edilmiş fakat pratiğini görmediğimiz tipte saldırılar bunlar. 2014 yılında NTP servisindeki monlist özelliğini istismar eden Amplification DDoS saldırısı 400 Gbps(2013 yılı Türkiye internet çıkışına yakın) civarında idi ve bu rakam dünyadaki en ciddi ddos saldırısı olarak tarihe geçmiştir.

Amplification DDoS Saldırıları

Standart DDoS saldırılarında amaç olabildiğince çok fazla sayıda sistem üzerinden hedef sistemlere belirli sayıda paket gönderimi yaparak devre dışı kalmasını sağlamaktır. Amplification tipi saldırılarında ise trafik kapasitesi yüksek aracı sistemler kullanarak saldırgan sahip olduğu bandwidth miktarından çok daha fazlasını hedef sisteme yönlendirir.

İlk olarak Smurf olarak adlandırılan bir ddos saldırısında kullanılan bu yöntem hızlı bir şekilde alınan önlemlerle internet dünyasının gündemini uzunca bir süre meşgul etmemiştir. Tekrar 2009 yılında DNS kullanılarak karşımıza çıktı, 2013 yılında ise DNS kullanılarak o zamana kadar ki en büyük DDoS saldırısı (Yaklaşık 300 Gbps) gerçekleştirildi. 2014 yılında NTP ile birlikte 400 Gbps'e ulaşmış oldu. Bu rakamlar ciddi koruması ve dağıtık altyapısı olmayan erişim sağlayıcılar için oldukça tehlikeli ve önlemesi bir o kadar da zordur.

Smurf saldırısı broadcast'e gönderilen bir adet ICMP paketine karşılık ilgili ağda açık olan tüm sistemlerin cevap vermesi mantığıyla çalışır. Böylece hedef broadcast adresinde 100 tane sistem açıkça bir paket ile 100 paketlik cevap alınabilir. Gönderilen paketlerin kaynak ip adresi ddos yapılmak istenen hedef olarak verilirse saldırgan 10 Mbps trafikle hedefe 1 Gbps saldırı trafiği üretebilir. Burada saldırıya yapan kaynak adresleri ilgili ağda bulunan ve broadcast ICMP paketlerine cevap dönen sıradan sistemler olacaktır.

Broadcast'e gelen ICMP isteklerine cevap vermeyecek şekilde yapılandırılmasıyla bu zafiyet hızlıca kapatılmıştır.

Linux sistemlerin ICMP paketlerine (broadcast) cevap verip vermediği aşağıdaki komutla öğrenilebilir.

```
sysctl net.ipv4.icmp_echo_ignore_broadcasts
```

komutun çıktısının aşağıdaki gibi olması gerekir.

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

Smurf ICMP kullandığı için ve ICMP genellikle yardımcı protokol görevine sahip olduğu için ICMP'nin kapatılması ile Smurf ve benzeri birçok atak engellenmiş oldu. Amplification saldırıları ICMP'nin yanında NTP, SNMP ve DNS protokolleri üzerinden de gerçekleştirilebilir. Son zamanlarda daha çok DNS ve NTP kullanılarak gerçekleştirildiğini görüyoruz.

NTP üzerinden gerçekleştirilen amplification ddos saldırıları

NTP, zaman senkronize protokolüdür. Bilişim sistemlerinin merkezi olarak zaman bilgilerini alıp güncelleyeceği bir servistir. UDP/123 portundan çalışır ve herhangi bir kimlik doğrulama aşaması bulunmamaktadır.

Öncelikle belirtmek gerekir ki bir protokol UDP tabanlı ise onun güvenliğini sağlamak için protokolden iki kat daha fazla uygulama uygulama geliştiricisine iş düşer.

Bir NTP sunucusunun durumunu öğrenmek için aşağıdaki komut yeterli olacaktır.

```
[root@s-guard19 ~]# ntpq -pn
remote      refid      st t when poll reach  delay  offset jitter
=====
=====
-208.53.158.34 164.244.221.197 2 u 266 512 377 20.715 8.447 0.091
+50.116.55.65 200.98.196.212 2 u 247 512 377 7.651 -1.170 0.225
+129.250.35.250 209.51.161.238 2 u 269 512 377 1.025 -0.229 0.096
*10.0.77.54 172.18.1.12 3 u 437 1024 377 0.135 0.568 0.522
```

NTP Monlist özelliği ve istismarı

ntp sunucular, kendine daha önce sorgu yapan ip adreslerini bellekte tutar ve bunu bir sorgu ile öğrenmemize fırsat tanır. Aşağıdaki komut ile o NTP sunucuyu kullanan son 600 ip adresi alınabilir.

```
[root@s-guard19 ~]# ntpdc -n -c monlist 50.22.202.163|more
remote address      port local address  count m ver code avgint 1stint
=====
=====
50.22.202.163      58609 50.22.202.163      2 7 2  0  32  0
184.45.66.119      80 50.22.202.163      69 7 2  0  7  0
83.250.130.244      80 50.22.202.163      1 7 2  0  0  0
199.255.209.211     6005 50.22.202.163     4914 7 2  0  4  0
89.108.86.169       21 50.22.202.163      736 7 2  0  4  0
83.108.22.62        80 50.22.202.163      76 7 2  0  4  1
141.0.23.147        80 50.22.202.163     140 7 2  0  4  2
83.98.143.20        80 50.22.202.163     142 7 2  0  4  2
76.76.4.146         80 50.22.202.163     2577 7 2  0  4  2
85.153.46.92        80 50.22.202.163     1383 7 2  0  3  2
5.39.114.89         53 50.22.202.163     4876 7 2  0  2  3
139.216.201.12      80 50.22.202.163      22 7 2  0 269  3
207.244.74.132      6005 50.22.202.163      97 7 2  0  4  3
178.235.0.18        80 50.22.202.163      38 7 2  0  7  4
184.173.86.203      80 50.22.202.163     105 7 2  0  80  8
31.169.77.59        35157 50.22.202.163     9 7 2  0 154 22
```

Burada gönderilen isteğin (NTP isteği) boyutu incelenirse yaklaşık olarak 250 Byte civarında olduğu gözükcektir. Bu pakete dönen cevapların toplamı (bir adet isteğe karşı toplamda 10-15 cevap dönmektedir) 7500 Byte'a yakındır. Buradan bir istekle hedef sistem üzerinden 30 kat daha fazla trafik üretebileceğimizi görebiliriz.

NTP, UDP tabanlı olduğu için gönderilecek isteklerde kaynak ip adresi olarak ddos saldırısı gerçekleştirilmek istenen hedef verilirse saldırgan 10 Mbps ile 300 Mbps trafik üretebilir. Bunun gibi 100lerce açık NTP sunucusu bularak

Monlist özelliği aktif sistemlerin tespiti

Nmap'in "ntp monlist" scripti kullanarak bir ağdaki monlist özelliği aktif olan NTP sunucuları tespit edilebilir.

```
nmap -sU -pU:123 -Pn -n --script=ntp-monlist 192.168.0.0/24
```

Örnek bir çıktı aşağıdaki gibi olacaktır.

```
[root@s-guard19 /usr/local/share/nmap/scripts]# nmap -sU -pU:123 -Pn -n --script=ntp-monlist localhost

Starting Nmap 5.35DC1 ( http://nmap.org ) at 2014-03-09 10:10 CDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00010s latency).
PORT      STATE SERVICE
123/udp   open  ntp
| ntp-monlist:
| Target is synchronised with 129.250.35.251
| Alternative Target Interfaces:
|   10.32.83.4   50.22.202.133  50.22.202.163
| Private Servers (1)
|   10.0.77.54
| Public Servers (3)
|   38.229.71.1  50.116.38.157  129.250.35.251
| Other Associations (14)
|   127.0.0.1 (You?) seen 3 times. last tx was unicast v2 mode 7
```

```
| 84.24.85.156 seen 11 times. last tx was unicast v2 mode 7
| 94.242.255.62 seen 10 times. last tx was unicast v2 mode 7
| 199.255.209.211 seen 11 times. last tx was unicast v2 mode 7
| 141.0.23.147 seen 11 times. last tx was unicast v2 mode 7
| 130.193.170.56 seen 10 times. last tx was unicast v2 mode 7
| 178.235.0.18 seen 10 times. last tx was unicast v2 mode 7
| 84.248.95.88 seen 21 times. last tx was unicast v2 mode 7
| 86.141.107.15 seen 11 times. last tx was unicast v2 mode 7
| 76.76.4.146 seen 10 times. last tx was unicast v2 mode 7
| 31.220.4.151 seen 10 times. last tx was unicast v2 mode 7
| 106.219.29.214 seen 10 times. last tx was unicast v2 mode 7
| 83.98.143.20 seen 9 times. last tx was unicast v2 mode 7
|_ 50.90.225.202 seen 4 times. last tx was unicast v2 mode 7

Nmap done: 1 IP address (1 host up) scanned in 1.17 seconds
```

Korunma Yöntemleri

En temel ve önemli korunma yöntemi NTP sunucusu açık olması gerekmiyorsa servisin kapatılması veya güvenlik duvarı arkasında ise portun Firewalldan kapatılmasıdır.

NTP sunucu olarak hizmet verilmesi gerekiyorsa ilk adımdaki öneriler işe yaramayacaktır. Bunun için NTP yapılandırma dosyasına gidip “*disable monitor*” satırının eklenmesi ve ntp servisinin yeniden başlatılması yeterli olacaktır.

```
[root@s-guard19 ~]# /etc/rc.d/ntpd restart
```

```
Stopping ntpd.
```

```
Starting ntpd.
```

```
[root@s-guard19 ~]  
# ntpdc -n -c monlist 50.22.202.163  
***Server reports data not found
```

İnceleme amaçlı örnek bir monlist paket dosyasına www.bga.com.tr/ntp.pcap adresinden erişim sağlanabilir.

EKLER:

DoS/DDoS, 2012 yılında tüm dünyada gerçekleştirilen siber saldırıların başında gelmektedir. Bunun temel nedeni DDoS saldırısını gerçekleştirmek için herhangi bir bilgi birikimi gerekmemesi ve etkisini anında göstermesidir. İnternet üzerinden elde edilecek çeşitli otomatik araçlar kullanılarak çok rahatlıkla kurumsal web sayfaları çalışamaz hale getirilebilir.

DDoS'un bu kadar basit bir saldırı olması çoğu güvenlik uzmanı ve kurum tarafından yeteri kadar ciddiye alınmamasına neden olmaktadır. Oysa siber saldırıların büyük çoğunluğu kurumların ve uzmanların yeteri kadar ciddiye almadıkları yerlerden gelmektedir.

DDoS bir altyapı problemidir ve tüm ISP'ler biraraya gelip ortak kurallar çerçevesinde hareket etmedikçe sonlanmayacaktır.

DDoS Pentest / Hizmet Durdurma Simulasyon Saldırıları

DDoS testlerinde hedef seçerken dikkatli olunmalı, ana sayfanın erişilemez olmasını isteyen bir saldırgan sadece ana sayfaya yönelik bir ddos saldırısı gerçekleştirmez. DNS, Firewall, veya daha korumasız gördüğü bir sisteme saldırı denemesinde bulunabilir. Bu nedenle ddos testlerinde farklı hedefler belirleyerek bu hedeflere yönelik gerçekleştirilecek saldırıların hangi noktalarda sıkıntı oluşturduğu bir excel olarak tutulmalıdır.

A	B	C	
Test Edilen / Etkilenen Sistem	Kenar Yönlendiriciler (Edge Routers)	Güvenlik Duvarları (Firewalls)	Saldırı Tespit ve Engelleme Sistemleri (IPS)
Kenar Yönlendiriciler (Edge Routers)			
Güvenlik Duvarları (Firewalls)		(X)	
Saldırı Tespit ve Engelleme Sistemleri (IPS)			
Yük Dengeleme Sistemleri (Load Balancer)		(X)	
Web Sunucular (HTTP ve HTTPS)			
E-posta Sunucuları ve Spam Engelleme Sistemleri			
DNS Sunucular	(X)		
VPN Sunucular (SSL VPN, IPSEC VPN)		(X)	
SIP Sunucular			
Hedef Sistemin Trafik Kapasitesi			

DDoS Test Çeşitleri

İnternet üzerinde 60'a yakın DDoS saldırısı gerçekleştirilmektedir. Bunlardan en temel saldırılar SYN Flood, DNS Flood, UDP flood ve HTTP Flood olarak bilinmektedir.

DDoS testlerinin gerçek anlamda sağlıklı sonuçlar verebilmesi için aşağıdaki

ana başlıkları içermesi beklenmektedir:

- Syn Flood Saldırıları
- ACK Flood Saldırıları
- FIN Flood Saldırıları
- TCP Connection Flood Saldırıları
- UDP Flood DDoS Saldırıları
- ICMP Flood DDoS Saldırıları
- HTTP GET, POST Flood Saldırıları
- DNS Flood DDoS Saldırıları
- Botnet Simulasyonu
- Rate Limiting, Karantina Özelliklerinin Test Edilmesi
- Uygulamalara Özel DoS Testleri
- SSL, HTTPS DoS Testleri

Sahte IP Paketleriyle Test

DDoS testlerinde Botnet gibi yasal olmayan sistemler kullanılamayacağı için test yapacak firmanın internet üzerinde ip spoofing yapabilecek özelliklere sahip sistemleri olması gerekmektedir.

Günümüzde çoğu ISP kendi sistemlerinden sahte ip adresleri ile trafiğin çıkmaması için UPRF gibi çeşitli önlemler almaktadır. Gerçekleştirilen testler için kullanılacak sistemlerin önünde UPRF korumalı bir yönlendirici bulunuyorsa siz paketlerin gittiğini düşünürsünüz fakat sahte üretilen paketler bir sonraki yönlendirici cihazdan ileri gidemez.

Test Trafik Kapasitesi

Ortalama trafik üretimi 2-5 Gbps ve üretilmesi gereken paket miktarı 3.000.000 PPS civarında olmalıdır. Bunun altında kalacak ddos testleri klasik ddos engelleme sistemleri tarafından rahatlıkla engellenebilir.

İşin Mantığını Anlayarak Test Yapma

DDoS testlerinin kolay gerçekleştirilebiliyor olması bu konuda hiç bir bilince sahip olmayan kişilerin de test gerçekleştirebilmesini sağlamaktadır. DDoS

engelleme sistemleri internet üzerinden indirilip çalıştırılacak çoğu ddos test/paket üretici yazılım için basit koruma özelliklerine sahiptir. Mesela Hping kullanarak yapılacak klasik bir SYN flood saldırısının korumalı bir sistemde hiç bir etkisi olmayacaktır. Hping SYN paketleri üretirken sıradan bir TCP bağlantısından farklı üretmektedir. Bu farkı yakalayan DDoS engelleme sistemleri paketleri işlemeyen düşürmektedir.

Hping TCP SYN Paketi Anormallik Örneği

Klasik bir TCP bağlantısı başlatma isteğinde başlık bilgileri aşağıdaki gibi olacaktır.

Transmission Control Protocol, Src Port: 57306 (57306), Dst Port: http (80), Seq: 688192453, Len: 0 Source port: 57306 (57306)

Destination port: http (80)

[Stream index: 0]

Sequence number: 688192453

Header length: 40 bytes

Flags: 0x002 (SYN)

000. = Reserved: Not set

...0 = Nonce: Not set

.... 0... = Congestion Window Reduced (CWR): Not set

.... .0.. = ECN-Echo: Not set

.... .0. = Urgent: Not set

.... ...0 = Acknowledgment: Not set

.... 0... = Push: Not set

....0.. = Reset: Not set

....1. = Syn: Set

[Expert Info (Chat/Sequence): Connection establish request (SYN): server port http]

[Message: Connection establish request (SYN): server port http]

[Severity level: Chat]

Hping tarafından üretilen SYN paketlerinde TCP SYN paketinde ise ACK bayrağı set edilmediği halde ACK numarası alanı dolu olarak gönderilmektedir.

Transmission Control Protocol, Src Port: here-lm (1409), Dst Port: http (80), Seq: 239285634, Len: 0 Source port: here-lm (1409)

Destination port: http (80)

[Stream index: 0]

Sequence number: 239285634

Acknowledgment Number: 0x1f7f9dc1 [should be 0x00000000 because

ACK flag is not set]

[Expert Info (Warn/Protocol): Acknowledgment number: Broken TCP. The acknowledge field is nonzero while the ACK flag is not set]

[Message: Acknowledgment number: Broken TCP. The acknowledge field is nonzero while the ACK flag is not set]

[Severity level: Warn]

[Group: Protocol]

Header length: 20 bytes

Flags: 0x002 (SYN)

000. = Reserved: Not set

...0 = Nonce: Not set

.... 0... = Congestion Window Reduced (CWR): Not set

.... .0.. = ECN-Echo: Not set

.... ..0. = Urgent: Not set

.... ...0 = Acknowledgment: Not set

.... 0... = Push: Not set

....0.. = Reset: Not set

....1. = Syn: Set

Çoğu DDOS engelleme sistemi bu anormalliği yakalayarak Hping tarafından üretilecek paketleri işlemeyen çöpe atmaktadır

DDoS Test Amaçlı Kullanılan Yazılımlar

DoS/ DDoS testlerinde yasal olmayan yollarla elde edilmiş Botnet yazılımları kullanılamaz. Botnet'lerin oluşturacağı trafiğin benzerini oluşturabilecek kapasitede açık kaynak kodlu ve ticari yazılımlar bulunmaktadır.

- Hping3
- Nping
- Juno
- T50
- ab
- Apache Jmeter
- DoSHTTP
- Mz
- Hyanae
- DDoSim

- Bonesi

Not: HTTP ve TCP üzerinden çalışan diğer uygulama seviyesi protokollerde ip spoofing yapılamayacağı için internet üzerinden yapılacak ddos testlerinde bu protokollere ait paket üretimleri gerçekleştirilemez.

Yerel ağda lab ortamı kurarak http ve benzeri protokoller için ip spoofing yapılarak tam bir botnet/zombi ordusu simülasyonu gerçekleştirilebilir. Internet üzerinden çeşitli bulut bilişim çözümleri kullanılarak 100-500-1000 ip adreslik uygulama seviyesi ddos saldırıları simüle edilebilir.

Ücretsiz yazılımlar genellikle kısıtlı özelliklere sahiptir. Geliştirme programlama dili olarak C ve Perl kullanıldığı için kod tarafı incelenerek eklemeler yapılabilir. Mesela hping paketleri gönderirken sadece bir ip adresinden ya da tamamen rastgele ip adreslerinden gönderebilir fakat belirli ip aralığından paket gönderme özelliği yoktur(botnet simülasyonu için).

Testlerde Kullanılan Kaynak Sistemlerin Konumlandırılması

Bazı DDoS koruma sistemleri internetten gelecek ataklara karşı başarılı bir koruma sağlarken eksik/hatalı yapılandırma nedeniyle ISP'nin kendi iç ağından gelebilecek ataklara karşı etkinlik sağlayamamaktadır. Bu nedenle ddos testlerinin ikisi yurt dışında ikisi yurt içindeki ana ISP'lerden olmak üzere en az 4 farklı lokasyondan gerçekleştirilmesi gerekir. Aksi halde ddos testlerinin sonuçları başarısız olacaktır.

URPF Koruması

URPF, çoğu ISP'nin kullandığı ve IP spoofing'i engelleme amacıyla kullanılan bir yöntemdir.

URPF'in aktif olup olmadığını anlamak için internete bağlı ayrı bir makineye ihtiyaç vardır.

```
tcpdump -i eth0 -tn tcp port 8080
```

URPF testi yapılması istenen makine üzerinden

```
hping3 --spoof 5.6.7.8. -p 8080 -S IPADRESİ_B
```

yazılır. Ardından tcpdump çıktısına bakılarak spoof edilmiş ip adreslerinden paketlerin çıkıp çıkmadığı belirlenebilir.

DDoS testleri hakkında detay bilgi için BGA DDoS Pentest Framework taslağı incelenebilir.

EK EKE EK

Hping Nedir?

Hping, istenilen türde TCP/IP paketleri oluşturmak için kullanılan harikulade bir araçtır. Her ne kadar adı ping komutundan esinlenilse de klasik ping uygulamasından çok daha gelişmiş bir uygulamadır.

Hping, oluşturulacak paketlerde tüm alanları kendimize özgü belirlenebilmesi, dinleme modu ile hostlar arası dosya transferi ve komut çalıştırma özelliği(Truva atı özelliği), IDS/IPS testleri için özel veri alanı belirtilebilmesi(ids imzalarının testi) gibi ileri düzey özelliklere sahiptir.

Hping'i tüm özellikleriyle efektif kullanabilmek, çıktılarını yorumlamak için orta düzey TCP/IP bilgisi gerekir. Klasik otomatize araçlardan farklı olarak hping ile tamamen kendi oluşturduğunuz (tcp/ip bilgisi burada işe yarıyor) paketleri ağa gönderirsiniz. Mesela XMAS Scan için nmap'de nmap -SX komutu verilirken hping'de XMAS scanin ne olduğunu, hangi TCP bayrakları ile gerçekleştirildiğini bilmeniz ve ona göre parametreleri oluşturmanız gerekir (hping -FUP hedef_sistem gibi). Kısacası hping maharetli ellerde kaliteli bir hamur işlevi görmektedir.

Hping'in kullanılacağı alanlar

Hping'i iyi bir şekilde öğrenip kullanma TCP/IP'nin geçerli olduğu yerlerde(tüm iletişim dünyası) avantaj sağlayacaktır. Kısa kısa hping'in somut olarak nerelerde ne amaçla kullanılacağını listeleyecek olursak;

- İsteğe göre düzenlenmiş TCP, UDP, ICMP, Raw-IP paketleri üretme
- Güvenlik duvarı işlevsellik ve performans testleri
- DOS engelleme sistemleri testleri
- Saldırı Tespit ve Engelleme Sistemleri işlevsellik ve performans testleri
- Gelişmiş port tarama
- Gelişmiş dosya transferi
- TCP/IP protokolleri üzerinden hedef sistemlerden bilgi toplama
- Geçmiş TCP/IP zaafiyetlerinin lab. Ortamında tekrar edilmesi

Nasıl Edinebilirim?

Hping Linux/UNIX/Windows sistemler üzerinde sorunsuzca kullanılabilir ve kullanım için herhangi bir ücret istenmemektedir. Hping.org adresinden indireceğiniz kaynak kodları sisteminizde derleyerek hping'i kullanmaya

başlayabilirsiniz.

Kurulum için kaynak koddan derleme yerine kullandığınız Linux dağıtımlarının paket yönetim sistemleri de kullanılabilir.

```
#yum install hping3 / Fedora için
```

```
#apt-get install hping3 / Debian için
```

Aynı sitede Windows sistemler için hazır kurulum paketleri de bulunmaktadır.

```
C:\Documents and Settings\root\Desktop\hping2.win32>hping -v
hping version 2.0.0-b1 Support for XP SP2 (Fri March 17 2006)
libpcap based binary

C:\Documents and Settings\root\Desktop\hping2.win32>
```

www.hping.org'dan indirdiğiniz paketlerde problem yaşarsanız http://downloads.sourceforge.net/sectools/hping2.win32.tar.gz?modtime=1163676368&big_mirror=0 adresindeki sürümü denemenizi tavsiye ederim..

Not: Windows sistemlerde hping'in bazı özellikleri sağlıklı çalışmamaktadır. Hping'in gerçek gücünü görmek için mutlaka Linux/UNIX tabanlı bir sistemde denenmelidir.

Temel Hping Kullanımı

Hping 'in paket göndermek için çeşitli modları ve komut satırı parametreleri vardır. Temel olarak hping ile raw IP, ICMP, TCP ve UDP paketleri üretilebilir. Üretilen paketlere ait tüm özellikler komut satırından belirtilebilir. Hping ile birlikte kullanılacak seçenekleri görmek için **-h** parametresi kullanılır.

```
[root@mail ~]# hping -h
usage: hping host [options]
  -h --help          show this help
  -v --version       show version
  -c --count         packet count
  -i --interval      wait (uX for X microseconds, for example -i u1000)
  -f --fast          alias for -i u10000 (10 packets for second)
  -n --numeric       numeric output
  -q --quiet         quiet
  -I --interface     interface name (otherwise default routing interface)
  -V --verbose       verbose mode
  -D --debug         debugging info
  -z --bind          bind ctrl+z to ttl (default to dst port)
  -Z --unbind       unbind ctrl+z
Mode
```

Hping versiyonu öğrenme

Hping'in yaygın kullanılan iki sürümü vardır. Bunlar; hping2 ve hping3. Her iki sürümde de bazı özellikleri diğer sürüm tarafından desteklenmemektedir. Hping3, hping2'e göre daha fazla özellik barındırdığı için tercih edilebilir. Kullandığınız sistemde hangi sürüm hping'in kurulu olduğunu öğrenmek için –v parametresi kullanılabilir.

Hping2 kurulu bir sistemden alınacak çıktı

```
# hping -v  
  
hping version 2.0.0-rc3 (Mon May 3 10:56:19 CEST 2004)  
libpcap based binary
```

Hping3 kurulu sistemden alınacak çıktı

```
$ hping -v  
  
hping version 3.0.0-alpha-1 ($Id: release.h,v 1.4 2004/04/09 23:38:56 antirez  
Exp $)  
This binary is TCL scripting capable
```

Hping Çalışma Modları

Hping çeşitli türde tcp/ip paketleri üretip bunları kullanabilir demiştik. Öntanımlı olarak hping TCP paketleri üretir, bunu değiştirmek için(udp, icmp veya ip yapmak için) aşağıdaki parametreler kullanılabilir.

-0 --rawip Raw ip paketleri kullanmak için

-1 --icmp Icmp Paketi oluşturmak için.

-2 --udp UDP Paketleri oluşturmak için.

-8 --scan Klasik Tarama modu.

-9 --listen Dinleme modu

Hping ile paket gönderimi

Hping kullanarak ilk paketimizi gönderelim. Öntanımlı olarak hping icmp

yerine TCP paketlerini kullanır. Yine öntanımlı olarak boş(herhangi bir bayrak set edilmemiş) bir tcp paketini hedef sistemin 0 portuna gönderir ve gelen cevabı ekrana basar. Dolayısıyla hping’de komut satırı parametreleri çok önemlidir. 0. Porta gönderilecek null bayraklı bir TCP paketi tüm Firewall/IPS cihazları tarafından engellenecektir.

```
# hping 192.168.1.1
HPING 192.168.1.1 (eth0 192.168.1.1): NO FLAGS are set, 40 headers + 0
data bytes
Ctrl^C
--- 192.168.1.1 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Tcpdump Çıktısı

```
# tcpdump -i eth0 -tttnn tcp port 0
IP 192.168.1.5.1894 > 192.168.1.1.0: . win 512
IP 192.168.1.5.1895 > 192.168.1.1.0: . win 512
```

TCP Paketleriyle Oynama

Bir TCP paketinde hangi alanlar vardır, öncelikle buna biraz değinelim sonra hping ile tcp başlığındaki alanlar ile oynayarak neler yapabiliyoruz görelim.


TCP oturumunda en önemli bileşen bayrak (flags) larıdır. Oturumun kurulması, veri aktarımı, bağlantının koparılması vb. gibi işlerin tamamı bu bayraklar aracılığı ile yapılır.

Hping kullanarak paket oluşturacağımız diğer protokollerde(IP, ICMP, UDP) bayrak tanımı yoktur.

```

Transmission Control Protocol, Src Port: 1168 (1168), Dst Port: 80 (80), Seq: 0, Len: 0
  Source port: 1168 (1168)
  Destination port: 80 (80)
  Sequence number: 0 (relative sequence number)
  Header length: 28 bytes
  Flags: 0x02 (SYN)
    0... .. = Congestion Window Reduced (CWR): Not set
    .0.. .. = ECN-Echo: Not set
    ..0. .... = Urgent: Not set
    ...0 .... = Acknowledgment: Not set
    ....0... = Push: Not set
    ....0... = Reset: Not set
    ....1. = Syn: Set
    ....0... = Fin: Not set
  Window size: 16384
  Checksum: 0xca99 [correct]
    [Good Checksum: True]
    [Bad Checksum: False]
  Options: (8 bytes)
    Maximum segment size: 1460 bytes
    NOP
    NOP

```



TCP'deki bayraklar ve hping parametreleri

TCP'de 6+2 bayrak vardır. Yoğun olarak 6 tanesi kullanılır ve hping ile aşağıdaki gibi belirtilir(komut satırı parametreleri)

SYN (hping -S)

FIN (hping -F)

RST (hping -R)

ACK (hping -A)

PUSH (hping -P)

URG (hping -U)

İlk oluşturacağımız paket her TCP oturumunun kurulmasında ilk adım olan SYN bayraklı bir paket. Hping'e **-S** parametresi vererek SYN bayraklı paketler gönderebiliriz.

```
# hping -S 192.168.1.1
```

```
HPING 192.168.1.1 (eth0 192.168.1.1): S set, 40 headers + 0 data bytes
```

```
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=2.5 ms
```

```
len=46 ip=192.168.1.1 ttl=255 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=0.9
ms
--- 192.168.1.1 hping statistic ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.9/1.7/2.5 ms
```

Hping tarafından oluşturulan paket detayı

```
# tcpdump -i eth0 -tttnn tcp and host 192.168.1.1
2007-07-05 19:44:30.096849 IP 192.168.1.4.2244 > 192.168.1.1.

2019758107:2019758107(0) win 512
2007-07-05 19:44:30.097393 IP 192.168.1.1.0 > 192.168.1.4.2244: R 0:0(0)
ack
2019758108 win 0
```

Yukarıdaki çıktıda görüleceği üzere hping ile oluşturulan SYN bayraklı TCP paketi hedef sistemin 0. Portuna gitmeye çalışmış ve hedef işletim sistemi tarafından RST paketiyle düşürülmüştür. 0. Port yerine daha farklı portlara paketler gönderilirse farklı sonuçlar elde edilecektir.

Hping çıktısını yorumlama

Hping çıktısı gönderilen pakete dönen cevabı içerir. Eğer paket gönderilen hedef sistem cevap dönmüyorsa ekran boş kalacaktır.

Gönderilen paket:

```
# hping -S vpn.lifeoverip.net -p 80
HPING vpn.lifeoverip.net (bce1 91.93.119.80): S set, 40 headers + 0 data
bytes
```

Dönen cevap

```
len=46 ip=91.93.119.80 ttl=64 DF id=48348 sport=80 flags=SA seq=0
win=65535 rtt=0.1 ms
```

len => dönen paketin boyutu

ip => paketi gönderen ip adresi(hedef sistem)

ttl => paketin yaşam süresi

DF => Parçalama biti aktif durumda

İd => IP paketine ait tanımlayıcı biricik(uniq) bilgi

Sport => paketin gönderildiği kaynak port

Flags => aktif TCP bayrakları

seq => paketin sıra numarası

win => paketin pencere boyutu

rtt => Round trip time süresi(milisaniye)

Hping kullanımında port belirtimi

-p parametresi kullanılarak hedef sisteme gönderilen paketlerin hangi porta gideceği belirtilir. Default olarak bu değer 0 dır.

-s parametresi ile kaynak TCP portu değiştirilebilir, ön tanımlı olarak bu değer rastgele atanır.

80.porta SYN bayraklı paket göndermek için

```
#hping -S -p 80 localhost
```

Komutu yeterli olacaktır.

-c parametresi ile kullanılmazsa hping durdurulana kadar(CTRL^c) paket göndermeye devam eder, **-c** ile kaç adet paket göndereceği belirtilir.

RST Bayraklı TCP paketleri oluşturmak

```
# hping -R -c 3 192.168.1.1 -p 80
HPING 192.168.1.1 (eth0 192.168.1.1): R set, 40 headers + 0 data bytes
--- 192.168.1.1 hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Benzer şekilde **-R** yerine diğer TCP bayrak tipleri konularak istenilen türde TCP paketi oluşturulabilir.

Aynı pakette birden fazla bayrak kullanımı

Hping ile TCP paketleri oluştururken tek bayrak kullanılması zorunlu değildir. İstenirse tüm bayrakları set edilmiş TCP paketleri de üretilebilir (tabi bu paket firewalllar tarafından düşürülecektir). Özellikle durum korumalı olmayan sistemleri test etmek için SYN/ACK, RST/ACK bayraklı paketler kullanılabilir.

hping -S -A localhost -p 80

```
HPING localhost (lo0 127.0.0.1): SA set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=54904 sport=80 flags=R seq=0 win=0
rtt=0.0 ms
len=40 ip=127.0.0.1 ttl=64 DF id=54955 sport=80 flags=R seq=1 win=0
rtt=0.0 ms
^C
```

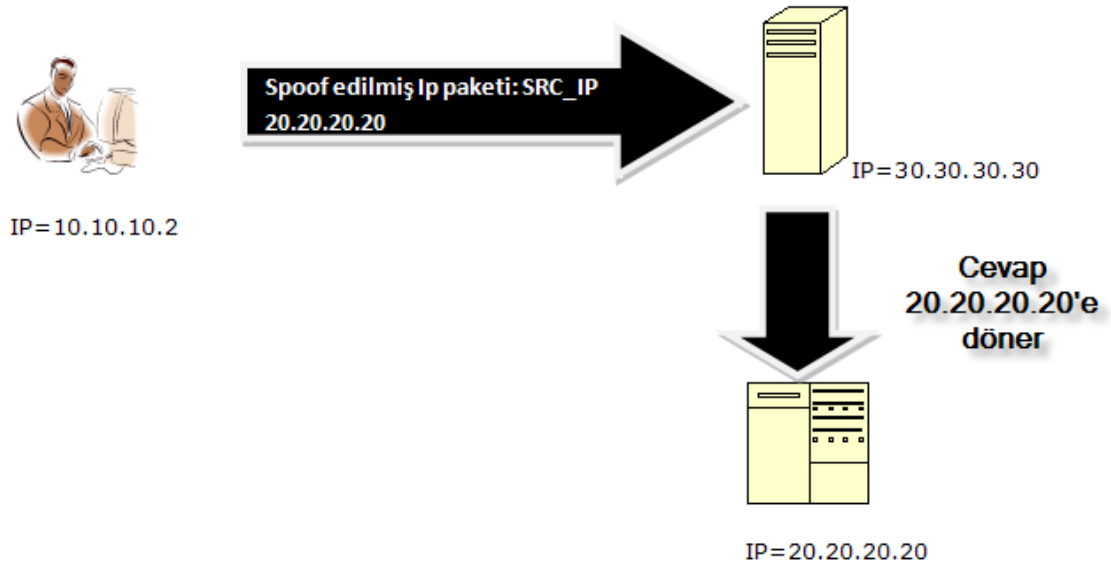
IP Paketleriyle Oynama

Hping ile IP paketlerine ait istenilen alanlar düzenlenebilir. IP başlığına bakılırsa en önemli alanların kaynak IP adresi, hedef_IP adresi, paket parçalama opsiyonu ve ip id numarası olduğu görülecektir.

```
Internet Protocol, Src: 91.93.119.80 (91.93.119.80), Dst: 192.168.2.27 (192.168.2.27)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 108
  Identification: 0xfb7e (64382)
  Flags: 0x04 (Don't Fragment)
    0... = Reserved bit: Not set
    .1.. = Don't fragment: Set
    ..0. = More fragments: Not set
  Fragment offset: 0
  Time to live: 56
  Protocol: TCP (0x06)
  Header checksum: 0xb19c [correct]
    [Good: True]
    [Bad : False]
  Source: 91.93.119.80 (91.93.119.80)
  Destination: 192.168.2.27 (192.168.2.27)
```

IP Başlığı

Hping ile spoof edilmiş paketler oluşturma(IP Spoofing)



Hping kullanarak istenilen ip adresinden geliyormuş gibi paketler üretilebilir. Burada dikkat edilmesi gereken husus kaynak ip adresini spoof ederek gönderdiğimiz paketler hedefe ulaştıktan sonra dönecek cevabın bize değil spoof edilmiş ip adresine döneceğidir.

Örnek: www.lifeoverip.net adresine 10.10.10.10 ip adresinden geliyormuş gibi SYN paketleri gönderelim.

```
# hping -a 10.10.10.10 -S -p 80 www.lifeoverip.net
HPING www.lifeoverip.net (r10 91.93.119.80): S set, 40 headers + 0 data
bytes

--- www.lifeoverip.net hping statistic ---
3 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Ekrandaki sonuç incelenirse geriye hiç paket dönmediği(**100% packet loss**) görülecektir. Bunun sebebi gönderdiğimiz paketlere dönen cevapların 10.10.10.10 ip adresine gitmesidir.

10.10.10.10 ip adresi de –eğer varsa böyle bir adres- kendisine gelen bu paketlere RST bayraklı TCP paketleriyle cevap dönecektir.

Rastgele Spoof edilmiş ip adreslerinden paket gönderme

Özellikle DOS/DDOs saldırılarının simülasyonlarında faydalı olan bir özelliktir. Hedef sisteme milyonlarca farklı ip adresinden geliyormuş gibi istek gönderilebilir.

```
# hping -rand-source -S -p 80 www.lifeoverip.net
```

Paketlerin TTL değeriyle oynanması

Paket oluştururken ip seviyesinde belirlenebilecek diğer bir özellik de paketlerin yaşam süresini belirleyen TTL değeridir. Hping ile istediğimiz ttl değerini pakete atayabiliriz.

```
#hping -t 10 www.google.com -p 80 -S
```

Burada dikkat edilmesi gereken husus TTL değerleri düşükse paketimizin hedefe ulaşmadan zaman aşımına uğramasıdır.

```
# hping -t 10 www.google.com -p 80 -S
HPING www.google.com (rlo 209.85.229.104): S set, 40 headers + 0 data
bytes
TTL 0 during transit from ip=209.85.255.176 name=UNKNOWN
TTL 0 during transit from ip=209.85.255.176 name=UNKNOWN
TTL 0 during transit from ip=209.85.255.176 name=UNKNOWN
TTL 0 during transit from ip=209.85.255.178 get hostname...^C
--- www.google.com hping statistic ---
4 packets tramitted, 3 packets received, 25% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Çıktıdan da görüleceği gibi ttl değerini 10 yapıp gönderdiğimiz paketler google.com'a ulaşmadan aradaki bir Router tarafından düşürülüyor ve bize bilgi mesajı olarak icmp paketleri dönüyor.

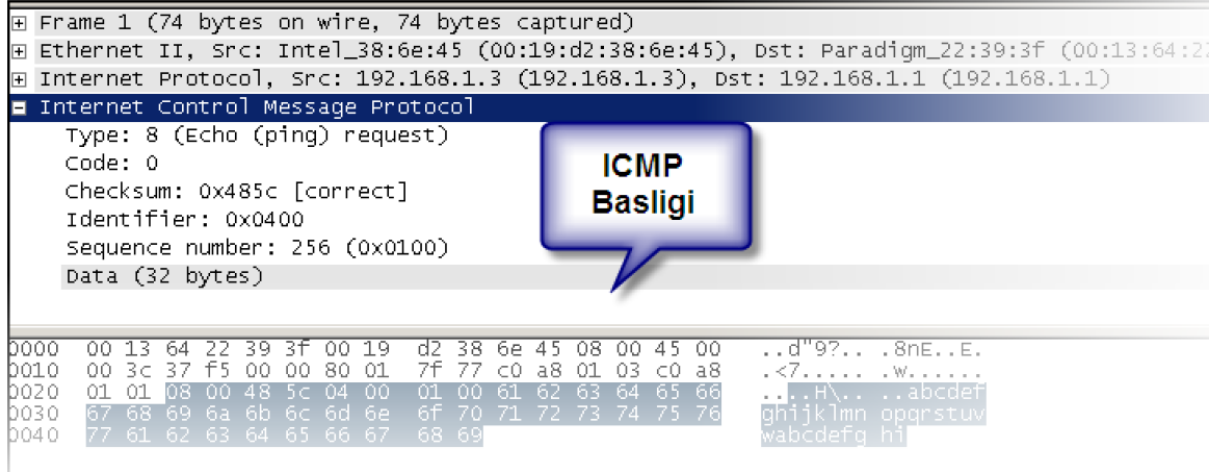
Ham IP Paketleri Oluşturma

Raw IP paketleri oluşturmak için hping'e ilk olarak **-rawip** parametresinin verilmesi gerekir. Özellikle network cihazlarının testlerinde bu tip paketler çok işe görmektedir.

ICMP Paketleriyle Oynama

ICMP diğer protokollere yardımcı olmak amacıyla tasarlanmış bir protokoldür. IP ve UDP paketlerinde herhangi bir hata mekanizmasının olmaması (ttl

expire olunca geriye ICMP mesajı dönmesi, kapalı udp portundan icmp mesajı dönmesi) ICMP'nin kullanımını kaçınılmaz kılmaktadır.



ICMP paketlerinde TCP ve UDP'deki gibi port değeri yoktur, bunlara benzer olarak icmp type ve icmp code değerleri vardır. Bir ICMP paketinin ne işe yaradığı bu değerlerle belirlenir.

Bazı icmp type degerleri ek olarak icmp code degerine de sahiptir. Mesela;

ICMP type 3 mesajı “Destination Unreachable”

Manasına gelmektedir fakat hedef ulaşılamaz(**Destination Unreachable**) mesajı da farklı anlamlar içerebilir. İşte burada icmp code değeri devreye girerek hangi kodun aslında ne manaya geldiğini söyler.

- 0 Net Unreachable
- 1 Host Unreachable
- 2 Protocol Unreachable
- 3 Port Unreachable
- 4 Fragmentation Needed and Don't Fragment was Set
- 5 Source Route Failed
- 6 Destination Network Unknown
- 7 Destination Host Unknown
- 8 Source Host Isolated
- 9 Communication with Destination Network is Administratively Prohibited
- 10 Communication with Destination Host is Administratively Prohibited
- 11 Destination Network Unreachable for Type of Service
- 12 Destination Host Unreachable for Type of Service
- 13 Communication Administratively Prohibited [RFC 1812]
- 14 Host Precedence Violation [RFC 1812]
- 15 Precedence cutoff in effect [RFC 1812]

Tüm icmp type/code degerlerine <http://www.iana.org/assignments/icmp-parameters> adresinden ulaşılabilir.

Hping ile ICMP tipi ve kodu belirtmek için kullanılan parametreler.

-C --icmp-type type
-K --icmp-code code

ICMP paket oluştururken kullanılabilecek diğer seçenekler için **hping -icmp-help** komutu kullanılabilir.

Klasik ping paketi (icmp echo request) oluşturmak

Hatırlatma: Her gün defalarca kullandığımız ping aracı ICMP paketleriyle çalışır.

```
# hping --icmp 10.10.10.2 -K 0 -C 8  
  
HPING 10.10.10.2 (r1 10.10.10.2): icmp mode set, 28 headers + 0 data bytes  
len=46 ip=10.10.10.2 ttl=64 id=23972 icmp_seq=0 rtt=0.2 ms  
len=46 ip=10.10.10.2 ttl=64 id=23981 icmp_seq=1 rtt=0.1 ms  
^C
```

UDP ve ICMP ilişkisi

UDP’de TCP benzeri bayrak mekanizması olmadığı için paketin durumuna ait bilgiler icmp mesajlarıyla iletilir. Mesela TCP’de kapalı porta gönderilecek bağlantı isteğine RST bayraklı cevap dönülecek ve gönderen işletim sistemi portun kapalı olduğunu anlayacaktır.

UDP’de ise bayrak mekanizması olmadığı için bu işi ICMP yapar. Yani kapalı UDP portuna gönderilen cevaplar icmp dest. Port unreachable mesajıyla cevap verilir.

```
# hping --udp -p 80 10.10.10.2  
  
HPING 10.10.10.2 (r1 10.10.10.2): udp mode set, 28 headers + 0 data bytes  
ICMP Port Unreachable from ip=10.10.10.2 name=blog.lifeoverip.net  
--- 10.10.10.2 hping statistic ---  
1 packets transmitted, 1 packets received, 0% packet loss  
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Broadcast ICMP Paketleri

Broadcast paketler tek bir adrese gönderilip o adres altındaki tüm canlı sistemlere ulaşan paketlerdir. Mesela 192.168.2.0 networkünün broadcasti olan 192.168.2.255 adresine bir adet paket gönderirsek bu ağda açık olan tüm sistemler o paketi alır ve uygun cevabı döner(di).

ICMP paketleri broadcast tipte olabilir, bu da çeşitli smurf saldırılarında icmp'nin kullanılabileceğini gösterir. 2000'li yıllarda broadcast adreslere gönderilen icmp paketleriyle ciddi DOS/DDOS saldırıları gerçekleştirilmiştir. Bu saldırılardan edinilen tecrübeler ışığında işletim sistemi ve sınır güvenlik cihazları broadcaste gelen paketlere cevap dönmeyecek şekilde yapılandırılmaya başlandı.

Hping ile hem ip spoofing hem de icmp broadcast özelliği kullanılarak geçmişte yapılan DDOS/DOS saldırıları simüle edilebilir.

UDP Paketleriyle Oynama

TCP/IP ailesinin en basit protokollerinden biridir. Gönderici port numarası, alıcı port numarası, paket boyutu ve checksum değerlerinden oluşan başlık bilgisine sahiptir.

```

User Datagram Protocol, Src Port: 64736 (64736), Dst Port: domain (53)
  Source port: 64736 (64736)
  Destination port: domain (53)
  Length: 44
  Checksum: 0x3a73 [correct]
    [Good Checksum: True]
    [Bad Checksum: False]

```

Hping ile UDP paketi oluşturma

```
# hping --udp -p 54 localhost
```

```

HPING localhost (lo0 127.0.0.1): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=127.0.0.1 name=localhost
ICMP Port Unreachable from ip=127.0.0.1 name=localhost

```

Aynı paket açık bir porta gönderilirse cevap dönmeyecektir. Bunun sebebi UDP'nin açık portlara gelen sıradan isteklere cevap dönmemesidir.

```
# hping 10.10.10.1 --udp -p 53
```

```

HPING 10.10.10.1 (r1 10.10.10.1): udp mode set, 28 headers + 0 data bytes
--- 10.10.10.1 hping statistic ---
4 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

```

İstenirse “-s port_numarası” parametresiyle kaynak port değeri de belirtilebilir.

UDP kullanarak traceroute

Traceroute genellikle ağlar arası ulaşım yollarını ve bir ağa ulaşmada kullanılacak ağ cihazlarının keşfinde kullanılır. Klasik traceroute programları yüksek numaralı udp portlarına istek göndererek dönen cevapları analiz edip sonuç çıkarmaya çalışır.

Günümüzde güvenlik duvarları bu tip paketlere izin vermediği için genellikle traceroute denemeleri başarısızlıkla sonuçlanır. Hping ile istediğimiz udp portundan trace çekerek sonuca ulaşabiliriz. Özellikle UDP port 53(DNS istekleri) hemen her sistemde açık olacağı için bu port tercih edilebilir.

```
#hping --udp -p 53 195.175.39.49 -T
```

Broadcast UDP Paketleri

UDP paketleri de icmp paketleri gibi broadcast adreslere gönderilebilir. Bunun sonucu olarak yüksek derecede DDOS saldırıları oluşturulabilir. Hping ile hem ip spoofing özelliği hem de udp broadcast mesaj gönderme özelliği kullanılarak bu tip DOS saldırıları simule edilebilir.