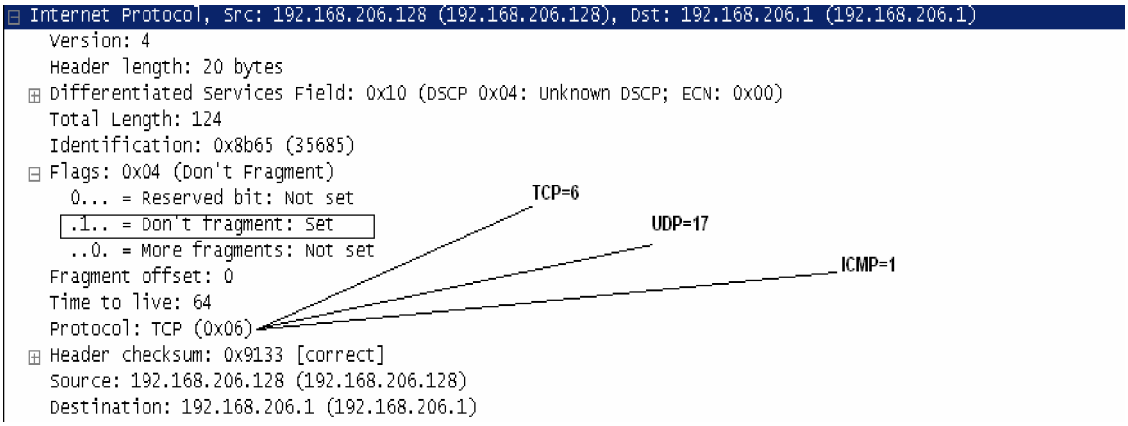
Parçalanmış(fragmented) Paketler

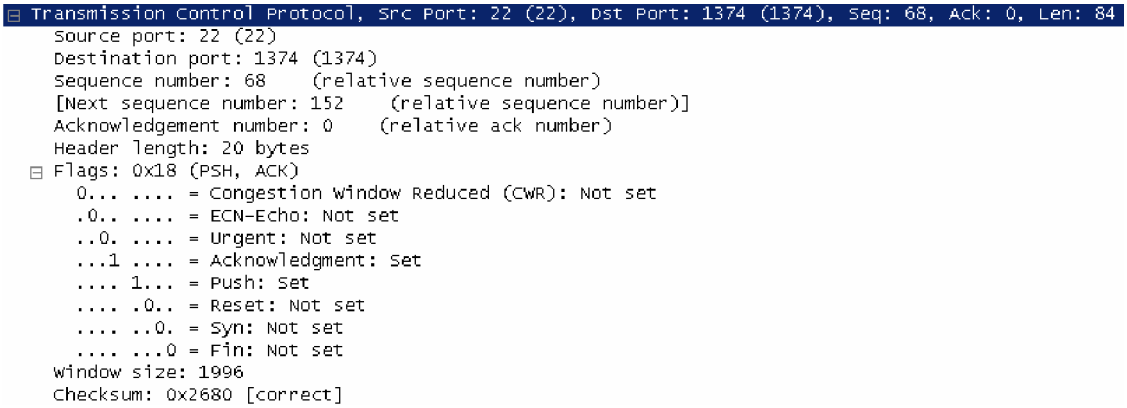
Parçalanmış paketler çoğu network ve güvenlik probleminin temelini teşkil etmesine rağmen genellikle gözardı edilen ve pek söz edilmeyen bir konudur. Bu kısa yazıda parçalanmış paketlerin nasıl çalıştığı, ne gibi tehlikeler oluşturabileceği ve basit koruma yöntemlerinden bahsedeceğim.

Parçalanmış paketleri anlamak için öncelikle bir IP paketinin temel yapısının bilinmesi ve analiz etmek için gerekli araçlara sahip olunması gerekir. Linux ortamında paket analizi için kullanılan en basit araç tcpdump(windows ortamları için windump kullanılabilir)dır, biraz daha görsel bir araç isterseniz Ethereal(yeni adı ile Wireshark) deneyebilirsiniz.

Tcpdump ile paket analizi yaparken bilinmesi gereken önemli nokta tcpdump'ın öntanımlı değerleri ile bir pakete ait 68/96 byte'ı gösterdiğidir , bu değer bir ip paketinin başlık bilgilerini göstermeye yetecektir . Fakat paketin payload kısmını görmek isterseniz muhtemelen bu değerden daha fazlasına ihtiyaç duyacaksınız. Tcpdump'a pakete ait değerin nekadarını görmek istediğinizi –s byte şeklinde bildirebilirsiniz. Temel Tcpdump kullanımı için http://www.enderunix.org/docs/tcpdump.html adresinden faydalanabilirsiniz.



IP Başlık Bilgileri

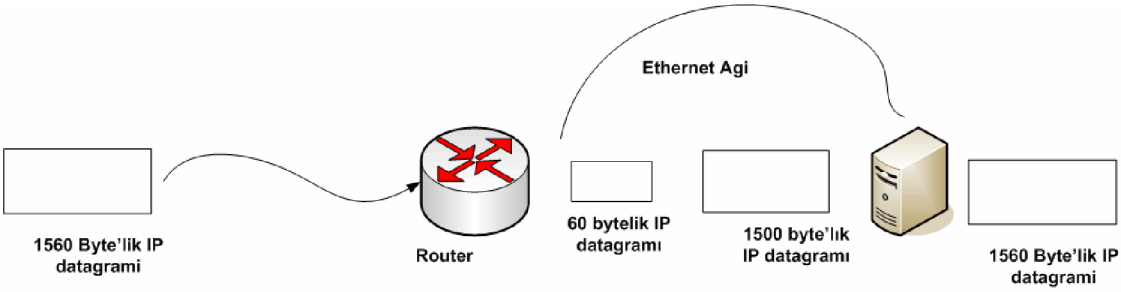


TCP Başlık Bilgileri

MTU(Maximum Transfer Unit): Bir ağa girişteki maksimum kapasiteyi belirtir. Mesela Ethernet ağları için MTU değeri 1500Byte'dır. Bu demek oluyorki bir Ethernet ağa giren bir paketin boyutu maksimum 1500 byte olabilir..

Fragmentation (parçalama) bir IP datagramının ağlar arasında dolaşırken kendi boyutundan daha düşük kapasitede bir ağa/ağ geçidine geldiğinde yaşadığı durumdur, yani parçalanma, bölünmedir.

Mesela Ethernet ağlarının MTU değeri 1500 byte'dır. Bizim IP datagramımızın değeri 1560** byte olsun, bu paket Ethernet ağının girişindeki router'a geldiğinde router diğer tarafında Ethernet ağı olduğunu ve bunun mtu değerinin 1500 byte olduğunu biliyor ve 1560 byte'lik gelen paketi Ethernet ağına parçalayarak gönderiyor. Ve paketimiz artık hedefine iki parça olarak ulaşıyor ve birleştiriliyor. İlk parça 1500 byte, sonraki parça 60 byte olmak üzere hedefine ulaşıyor.




```
C:\Documents and Settings\rapsodi>ping snort-home -n 1 -l 1560

Pinging snort-home [192.168.206.128] with 1560 bytes of data:

Reply from 192.168.206.128: bytes=1560 time=2ms TTL=64
```

```
[root@Snort ~]# tcpdump -ttttnn icmp -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
2006-11-12 19:22:26.565615 IP (tos 0x0, ttl  64, id 14620, offset 0, flags [+], proto 1, length: 1500)
192.168.206.1 > 192.168.206.128: icmp 1480: echo request seq 3328
2006-11-12 19:22:26.572492 IP (tos 0x0, ttl  64, id 14620, offset 1480, flags [none], proto 1, length: 108)
192.168.206.1 > 192.168.206.128: icmp
2006-11-12 19:22:26.574002 IP (tos 0x0, ttl  64, id 4095, offset 0, flags [+], proto 1, length: 1500)
192.168.206.128 > 192.168.206.1: icmp 1480: echo reply seq 3328
2006-11-12 19:22:26.574044 IP (tos 0x0, ttl  64, id 4095, offset 1480, flags [none], proto 1, length: 108)
192.168.206.128 > 192.168.206.1: icmp
```

Not: Windows komut satırından ping –l 1500 komutunu verdiğimizde 1500 bytelik bir buffer alanı vermiş oluyoruz(bir nevi icmp için veri kısmı) bir de bu pakete 20 byte IP+8 byte icmp başlığı eklendiği için 1588 oluyor.

Yukarıdaki resimde dikkatimizi çeken bir nokta var. IP datagramımız 1560 byte ve biz bunun 1500 ve 60 byte olmak üzere iki pakete parçalanarak gitmesi gerektiğini düşünüyoruz fakat tcpdump çıktısında ilk paket 1500, ikinci paket 108 byte olarak gözüküyor. Bunun sebebi parçalanmış her paketin de bir IP paketi olduğu ve her IP paketinin de 20 bytelik bir başlık bilgisi taşıdığıdır. Ilk parça paketde icmp başlık bilgileri(8) byte da taşındığı için ilk paketin orjinal veri boyutu aslında 1472'dir.

Elimizde 108 bytelik bir eksiklik var bunu bir sonraki pakete veriyoruz, bir sonraki paketin boyutu 60 olmalıydı buna bir de IP başlığı ekliyoruz(dikkat: icmp başlığı sadece ilk pakette var!) 60+28+20=108 etti. Yani ikinci paketin toplam boyutu 108 byte olmalı ki tcpdump çıktıları da bunu doğruluyor.

C:\>ping snort-home -n 1 -l 3000

#tcpdump -ttttnn icmp -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

2006-11-12 19:52:37.567038 IP (tos 0x0, ttl 64**, id 15332**, **offset 0,** flags [+], proto 1, length: 1500) 192.168.206.1 > 192.168.206.128: icmp 1480: echo request seq 4352
2006-11-12 19:52:37.567040 IP (tos 0x0, ttl 64, **id 15332**, offset 1480, flags [+], proto 1, length: 1500) 192.168.206.1 > 192.168.206.128: icmp
2006-11-12 19:52:37.567042 IP (tos 0x0, ttl 64, id **15332**, **offset 2960**, **flags [none**], proto 1, length: 68) 192.168.206.1 > 192.168.206.128: icmp

Parçalanan paketlerin hedefe ulaştığında tekrar birleştirilip orjinalinin elde edilmesi için her pakette bulunması gereken bazı alanlar vardır. Bunlar;

- Fragmentation ID, yukarıdaki resimde ID olarak gözüküyor. Bir IP datagramına ait parçalanmış tüm paketlerde bu değer aynı olmalıdır.
- Parçalanmış her paket datagramin hangi kısmını taşıdığını(Offset değeri) ve sırasınıbilmelidir kendisinden sonra paket varsa bu alan flags[+] kendisi son paket ise değer flags[none] olur.
- Parçalanmış her paket taşıdığı veri boyutunu bilmelidir.

Peki parçalama bazı durumlarda zorunlu ise bunun ne gibi zararları olabilir ? Bu soruya yine adım adım basit bir örnek üzerinden ilerleyerek cevap verelim.

Güvenlik duvarınız stateful özelliğine sahip değil(bu devirde kalmamıştır böyle bir firewall ama örnek olsun) ve paketleri protokollerine gore yasaklıyor. Sıradan bir paket geldiğinde onu başlık bilgilerine bakarak filtreleyebilir fakat eğer paket parçalanmış bir paket ise sadece ilk parça paketi filtreleyebilecektir, diğer parça paketler firewalldan süzülerek geçecektir. Ya da firewallunuz parçalanmış paketlerden anlamıyorsa gelen parçalı her paketi ayrı ayrı değerlendirecektir ve ilk paket sonrasındakilerden birşey anlamayacaktır.

Fragmented paketler için sağlam çözüm kullandığımız güvenlik duvarının parçalanmış paketleri gördüğü zaman(bunu ip başlığındaki More Fragment değerinden anlayabilir) bunları hedefe ulaştıurmadan kendisi birleştirebilmeli, kurallarını control edip eğer izin verilmişse hedef sisteme gönderebilmelidir.

OpenBSD PF güvenlik duvarındaki scrub özelliği kullanılarak parçalanmış paketlerin güvenlik duvarında tekrar birleştirilmesi ve hedefe bu şekilde ulaştırılması sağlanabilir. Yine benzer şekilde Snort saldırı tespit ve engelleme sistemi kullanılarak parçalanmış paketler birleştirilerek saldırı imzaları araması yapılır.

Unutulmaması gerekenler: parçalanmış paketlerde sadece ilk paket protocol bilgisini taşır.!

OpenBSD PF ve parçalanmış paketler

Scrub özelliği

fragment reassemble : Gelen parçalanmış paketleri hedefe iletmeden once birleştirerek göndermek için kullanılır. Bu seçeneğin yararı güvenlik duvarları paket tamamlanmadan kuralları tam uygulamayacağı için fragment paketlerin güvenlik duvarı kurallarına gelmeden birleştirilmesi gerekir. Ek olarka fragment crop, fragment drop-ovl , no-df seçeneklerine de gözatabilirsiniz.

Snort ve Parçalanmış Paketler

Snort ile birlikte gelen frag3 preprocessor'u kullanılarak IDS sistemine gelen parçalanmış paketler detection engine(Snort'da kural karşilaştirmasinin yapildiği kısım)gelmeden birleştirilerek yapilmaya çalışılan ids atlatma tekniklerini boşa çıkarmak mümkün olabilmekte.

Fragment paketlerle oynamak istiyorsanız Dug Song tarafından yazılan ve uzun yıllardır Fiewwall/IDS/IPS gibi sistemleri test etmede kullanılan fragtest(http://www.monkey.org/~dugsong/fragroute/) programını kullanabilirsiniz.

Kaynaklar:

- http://www.openbsd.org/faq/pf/scrub.html
- The Tao of Network Security Monitoring
- http://www.snort.org/docs/snort_htmanuals
- SANS Intrusion detection in-depth (kurs notlari)

Huzeyfe ÖNAL <huzeyfe[at]EnderUNIX.org>
http://www.enderunix.org/huzeyfe