[TCP/IP Ağlarda Parçalanmış Paketler]

[Parçalanmış Paketler ve Güvenlik Sistemlerine Etkileri]

Huzeyfe ÖNAL <huzeyfe@lifeoverip.net>

5/27/2009

[Internetin temelini oluşturan TCP/IP kötü amaçlı kullanılabilecek çeşitli özelliklere sahiptir. Bunlardan biri de ağlar arası iletişimde çok sık kullanılan parçalanmış paketlerdir. Bu yazı parçalanmış paketlerin ne olduğu ve güvenlik yönünden ne gibi sakıncalar içerebileceğini anlatmaktadır.]

Contents

Parçalanmış Paketler

Parçalanmış paketler(Fragmented Packets) konusu çoğu network ve güvenlik probleminin temelini teşkil etmektedir. Günümüzde tercih edilen NIDS/NIPS(Ağ tabanlı Saldırı Tespit ve Engelleme Sistemleri) sistemleri bu konuya bir çözüm getirse de hala eksik kalan, tam anlaşılmayan kısımlar vardır. Bu sebepledir ki günümüzde hala parçalanmış paketler aracılığıyla yapılan saldırılara karşı korunmasız olan popüler IPS yazılımları bulunmaktadır[1].

Bu yazıda parçalanmış paketlerin nasıl çalıştığı, ne gibi tehlikeler oluşturabileceği ve basit koruma yöntemlerinden bahsetmeye çalışacağım.

Yazı iki bölümden oluşmaktadı; ilk bölümde IP parçalamanın ne olduğu, hangi durumlarda nasıl gerçekleştiği, ikinci bölümde IP parçalamanın ne tip güvenlik zaafiyetlerine sebep olabileceği konuları üzerinde durulacaktır.

IP (Internet Protocol) Yapısı

Parçalanmış paketler konusunun iyi anlaşılabilmesi için öncelikle IP(Internet Protocol) paketinin temel yapısının bilinmesi gerekmektedir. IP paketinin yapısını analiz etmek Sniffer olarak adlandırılan çeşitli araçlar vasıtasıyla olur. Bu araçlardan bazıları aşağıdaki gibidir;

- Tcpdump
- Wireshark
- Snort
- Tshark
- Snoop

Linux ortamında paket analizi için en sık kullanılan araçlar tcpdump ve daha görsel bir araç olan Wireshark'dır.

Windows ortamları için windump, thsark ya da daha görsel bir yazılım olan Wireshark tercih edilmektedir.

Tcpdump ile paket analizi yaparken dikkat edilmesi gereken en önemli nokta tcpdump'ın öntanımlı değerleri ile bir pakete ait 68/96 byte'ı göstermesidir. Bu değer bir ip paketinin başlık bilgilerini göstermeye yetecektir fakat paketin payload kısmı incelenmek istenirse bu değerden daha fazlasına ihtiyaç duyulur.

Tcpdump'la analiz yaparken –s 0 parametresini kullanarak bir pakete ait tüm alanları görmek mümkündür. Temel Tcpdump kullanımı için http://www.enderunix.org/docs/tcpdump.html adresinden faydalanabilirsiniz.

MTU (Maximum Transfer Unit)

MTU değeri bir ağa girişteki maksimum kapasiteyi belirtir. Mesela Ethernet ağları için **MTU** değeri 1500 byte, FDDI için 4500 byte'dır. Bu demek oluyor ki ethernet ağa giren bir paketin boyutu maksimum 1500 byte, FDDI ağa giren bir paketin boyutu en fazla 4500 byte olabilir.

MTU değerleri farklı iki ağ arasında geçişlerde eğer ilk ortamın MTU değeri daha büyükse IP paketlerinde yeni girilecek ortama göre parçalama işlemi yapılır.

Paket Parçalama(Fragmentation)

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

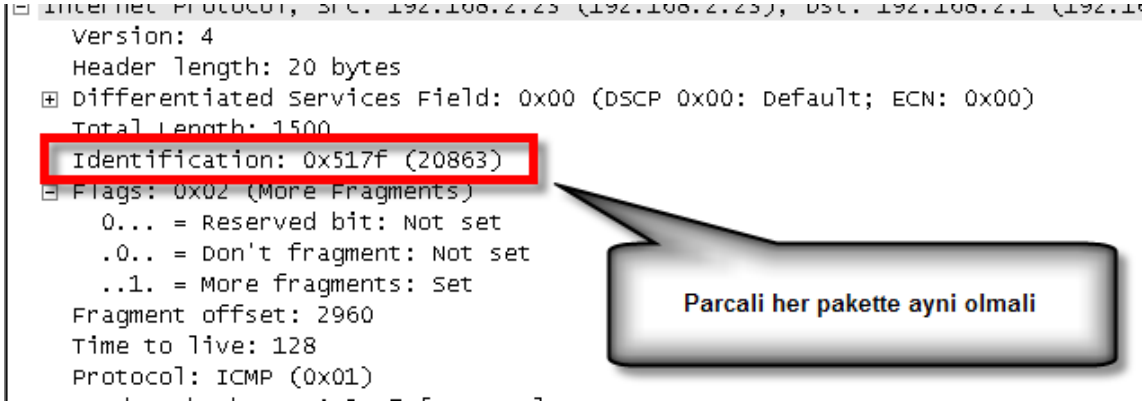
Mesela Ethernet ağlarının MTU değeri 1500 byte'dır. Bizim IP datagramımızın değeri 1560 byte olsun. Bu paket ethernet ağının girişindeki router'a geldiğinde router diğer tarafında ethernet ağı olduğunu ve bunun mtu değerinin 1500 byte olduğunu bilir ve 1560 byte'lik gelen paketi Ethernet ağına parçalayarak gönderiyor.

Paketimiz artık hedefine ilk parça 1500 byte, ikinci parçası 60 byte olmak üzere iki parça olarak ulaşır ve birleştirilir.

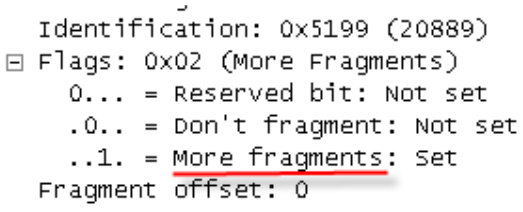
Paketlerin Birleştirilmesi

Parçalanmış paketlerin hedefe ulaştığında doğru sırada birleştirilmesi gerekir. Paketler hedefe ulaştığında tekrar birleştirilip orjinalinin elde edilmesi için her pakette bulunması gereken bazı alanlar vardır. Bunlar;

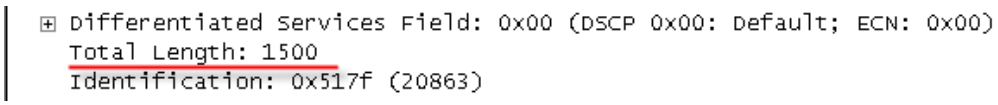
Fragmentation ID, diğer bir isimle IP ID. Bir IP datagramına ait parçalanmış tüm paketlerde bu değer aynı olmalıdır.



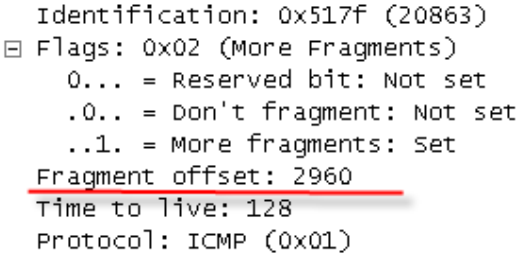
• Parçalanmış her paket datagramın hangi kısmını taşıdığını (Offset değeri) ve sırasını bilmelidir. Kendisinden sonra ek parça paket varsa bu alan flags[+], paketin kendisi son paket ise değer flags [none] olur.



• Parçalanmış her paket taşıdığı veri boyutunu ve hangi byte'dan itibaren taşıdığını bilmelidir. Ne kadarlık bir veri taşıdığı Total Lenght ile belirtilir.



Hangi Byte'dan itibaren bu verinin ekleneceği de "Fragment Offset" değeri ile belirtilir. Yani önceki paket 2960 byte tasımıstır, biz de buna ek 1500 byte yapıp göndereceğiz, bir sonraki pakette offset değeri 2960+1500 olacaktır(aslında 2960+1480)



Ip Parçalama Örneği

Bir IP paketinin nasıl parçalandığını görmenin en kısa yolu ethernet ağında 1500 bytedan büyük paket göndermektir. Bunu da windows/Linux ortamındaki ping komutu ile yapabiliriz. Daha detaylı paket oluşturma için hping aracı incelenebilir.

```
C:\Documents and Settings\redlabs>ping -l 5000 192.168.2.1 -n 1

Pinging 192.168.2.1 with 5000 bytes of data:

Reply from 192.168.2.1: bytes=5000 time=3ms TTL=255

Ping statistics for 192.168.2.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 3ms, Average = 3ms
```

Yukarıdaki komutla 192.168.2.1 sistemine gönderilmek üzere 5000 byte uzunluğunda bir adet paket(-n 1) hazırlamış olduk.

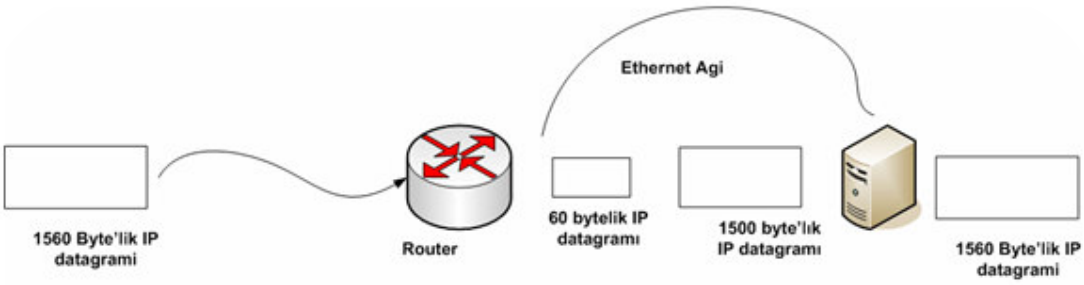
Bu komut çalıştırıldığında ağ arabirimi sniffer(Wireshark) aracılığıyla izlenirse aşağıdaki gibi bir çıktı alınacaktır. Çıktıda dikkat edilmesi gereken tek bir paket olarak gönderilen icmp paketinin 3 parçaya ayrılarak hedefe gönderildiğidir.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.23	192.168.2.1	IP	Fragmented IP protocol (proto=ICMP 0x01, off=0)
2	0.000016	192.168.2.23	192.168.2.1	IP	Fragmented IP protocol (proto=ICMP 0x01, off=1480)
3	0.000023	192.168.2.23	192.168.2.1	IP	Fragmented IP protocol (proto=ICMP 0x01, off=2960)
4	0.000033	192.168.2.23	192.168.2.1	ICMP	Echo (ping) request

Detay İnceleme

Amacımız 1560 byte olarak gönderilen bir paketin nasıl parçalandığı ve parçaların ne içerdiğini incelemek. Bunun için yine Windows komut satırından ping aracını kullanıyoruz.

Windows komut satırından ping –l 1560 komutunu verdiğimizde 1560 bytelik bir buffer alanı vermiş oluyoruz (bir nevi icmp için veri kısmı). Bu pakete 20 byte IP, 8 byte icmp başlığı eklendiği için toplam paket boyutu 1588 byte oluyor.




```
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
```

Yukarıdaki resimde dikkatimizi çeken bir nokta var. **IP datagram**'ımız 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 parçada **icmp** başlık bilgileri (8) byte da taşındığı için ilk paketin orjinal veri boyutu aslında 1472'dir.

Parçalanmış paketlerde sadece ilk paket protokol başlık bilgisini(TCP, UDP, ICMP vs) taşır.

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 Çıktısı

```
#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
```

Buraya kadar IP paketlerinin parçalanması, parçalanmanın neden ve nasıl olduğunu inceledik. Konunun detayının anlaşıldığını düşünerek parçalanmış paketlerin güvenlik yönünden incelenmesine geçebiliriz.

Parçalanmış Paketler ve Güvenlik Zaafiyetleri

Öncelikle paket parçalamanın olağan bir durum olduğunu belirtmek gerekir. İyi niyetlerle düşünülmüş bu özellik bugüne kadar çeşitli ciddi güvenlik sorunlarına sebep olmuştur. Bunların başında denial of service saldırısı yapmak için kullanılan Ping Of Death ve Tear Drop gelir.

Bu saldırılara sebep olan güvenlik açıklıkları uzun zaman önce işletim sistemi geliştirici firmalar tarafından kapatılmıştır fakat paket parçalama ile yapılan Firewall/IDS/IPS atlatma yöntemleri hala bazı sistemler üzerinde çalışabilmektedir.

Parçalanmış Paket Oluşturma Araçları

Paket parçalama işlemi normalde bizim (kullanıcılar) tarafımızdan yapılmaz. Ağlar arası geçişleri sağlayan yönlendirici sistemler(router) gerektiğinde bu işlemi gerçekleştirir. Fakat internette bulunan çeşitli araçlar kullanılarak kendi isteğimize göre paketleri parçalayıp gönderebiliriz.

Bu da bize kullandığımız network sistemlerini test etme imkanı sunar.

Hping ile Parçalanmış Paket Oluşturma

Hping TCP/IP ağlarda kullanılan ileri düzey bir paket oluşturma aracıdır. Hping kullanarak bir IP paketine ait tüm özellikleri kendimiz belirleyebiliriz. Aşağıdaki komut hping'in paket parçalama amaçlı kullanılabilecek seçeneklerini göstermektedir.

```
root@redlabs:~# hping3 --help|grep -i frag
  -f  --frag       split packets in more frag.  (may pass weak acl)
  -x  --morefrag   set more fragments flag
  -y  --dontfrag   set dont fragment flag
  -g  --fragoff    set the fragment offset
  -m  --mtu        set virtual mtu, implies --frag if packet size > mtu
```

Nmap Taramalarında Parçalanmış Paket Kullanımı

Nmap –f (--mtu) parametresi ile port taramalarında istenilen boyutlarda parçalanmış paketler kullanmaya izin verir.

Aşağıdaki taramada hedef sistem taranırken gönderilecek paketlerin boyutları 8 byte olarak düzenlenmiştir.

```
root@redlabs:~# nmap  --mtu  8 192.168.2.1 --packet_trace -n -p 80

Starting Nmap 4.85BETA7 ( http://nmap.org ) at 2009-05-27 16:03 EDT
SENT (0.0670s) TCP 192.168.2.22:49224 > 192.168.2.1:80 ?? ttl=42 id=52276
iplen=28 frag offset=0+  seq=1540756055 (incomplete)
SENT (0.0670s) TCP 192.168.2.22:?? > 192.168.2.1:?? S ttl=42 id=52276 iplen=28
frag offset=8+   option incomplete
SENT (0.0680s) TCP 192.168.2.22:?? > 192.168.2.1:?? ?? ttl=42 id=52276 iplen=28
frag offset=16  (incomplete)
RCVD (0.0680s) TCP 192.168.2.1:80 > 192.168.2.22:49224 SA ttl=64 id=0 iplen=44
seq=681563302 win=5840 ack=1540756056 <mss 1460>
Interesting ports on 192.168.2.1:
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)
```

Fragroute ve Fragrouter Araçları

Hping ve Nmap parçalanmış paketler oluşturmaya ve bunlarla bazı işlemler yapmaya izin verse de özellikleri kısıtlıdır. İleri düzey testler için her iki araç da yetersiz kalmaktadır. Gerçek ortamlarda test yapabilmek için bu işe özel yazılmış alternatif araçlar

kullanılmalıdır. Bunlar fragroute ve fragrouter'dir. Her iki araç da temelde aynı işi yapmaya yöneliktir. Aralarında basit farklar vardır.

Fragroute ile Parçalanmış paket çalışmaları

Fragroute halihazırda oluşturulmuş bir trafiği(bir web isteği) istenen özelliklere göre parçalamaya yarar. Yani siz bir yandan web sayfasını ziyaret ederken diğer yandan fragroute sizin web sayfanıza giden istekleri belirli boyut ve özelliklerde parçalayarak gönderir.

Fragroute'in sağlıklı çalışabilmesi için öncelikle Linux sistemlerde aşağıdaki komut çalıştırılmalıdır.

echo "0" > /proc/sys/net/ipv4/conf/all/rp_filter

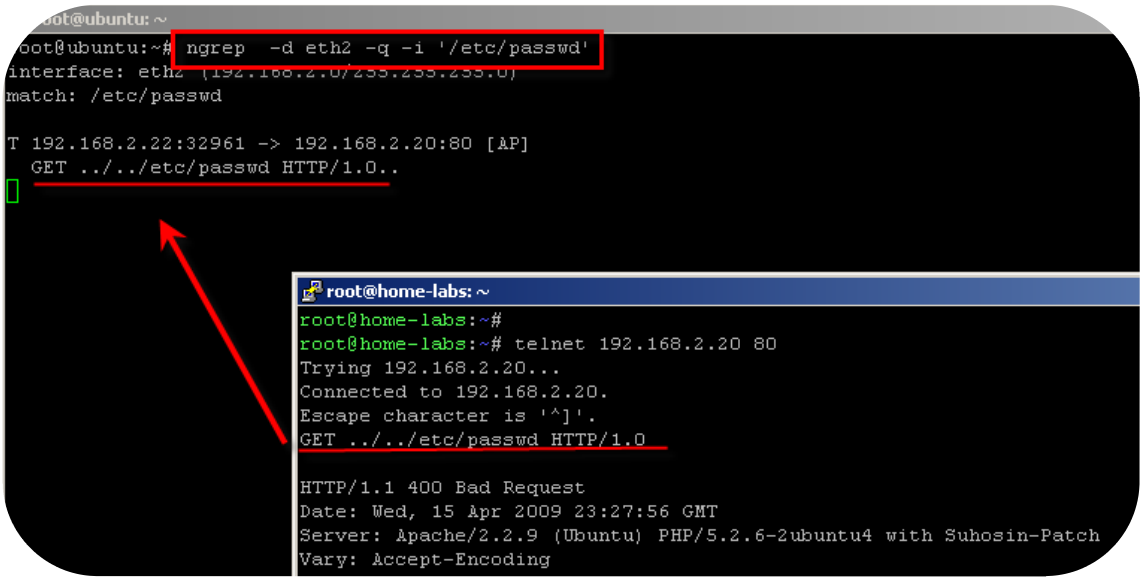
Fragroute ile Ngrep örneği;

Ngrep ağ trafiğinde string arama yazılımıdır. Mesela ngrep –d eth2 –i '/etc/passwd'

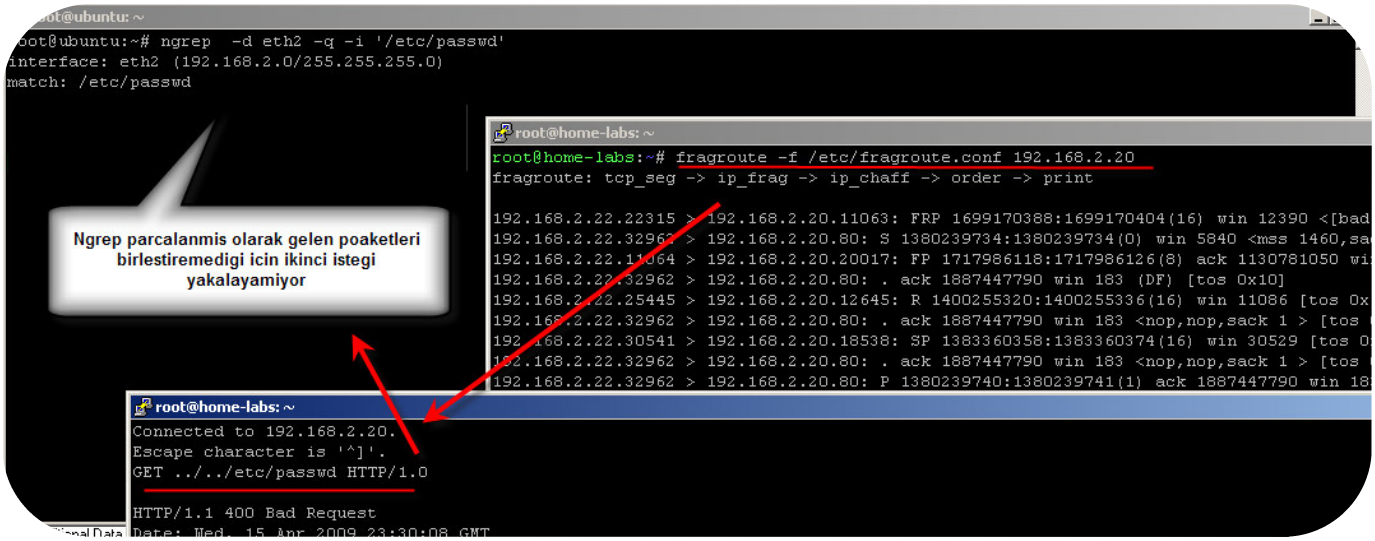
Komutu eth2 arabirimini dinleyerek trafikte geçen /etc/passwd stringini yakalar ve ekrana basar.

Bizim yapacağımız test fragroute ile bir paketi parçalayıp göndermek ve Ngrep'in bunu yakalayamadığını görmek.

Önce paketleri parçalamadan Ngrep'i çalıştıralım ve HTTP isteğinde gönderdiğimiz /etc/passwd stringini yakaladığını görelim.



Yukarıdaki işlemi (HTTP isteğinde /etc/passwd gönderimi fragroute aracılığıyla yaparsak Ngrep'in bir şey yakalayamadığını görürüz.



Parçalanmış Paketler ve Güvenlik Duvarları

Güvenlik duvarına bir paket geldiğinde onu başlık bilgilerine bakarak filtreleyebilir fakat paket parçalanmış bir paket ise sadece ilk parça paketi filtreleyebilecektir, diğer parça paketler firewalldan süzülerek geçecektir.

Güvenlik duvarları gelip giden paketleri kural tablosu ile karşılaştırabilmesi için paketlerin parçalı olmaması gerekir. Bu da güvenlik duvarlarının paket birleştirme özelliğine sahip olmalarını zorunlu tutar.

OpenBSD PF güvenlik duvarındaki scrub özelliği kullanılarak parçalanmış paketlerin güvenlik duvarında tekrar birleştirilmesi ve hedefe bu şekilde ulaştırılması sağlanabilir.

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 olarak **fragment crop**, **fragment drop-ovl** , **no-df** seçeneklerine de inelenebilir.

Parçalanmış Paketler ve Saldırı Tespit Sistemleri

Parçalanmış paketler konusunda en sıkıntılı sistemler IDS/IPS'lerdir. Bunun nedeni bu sistemlerin temel işinin ağ trafiği inceleme olmasıdır. Saldırı tespit sistemleri gelen bir paketin/paket grubunun saldırı içerikli olup olmadığını anlamak için çeşitli kontrollerden geçirir. Eğer bu kontrolleri geçirmeden önce paketleri birleştirmezse çok rahatlıkla kandırılabilir.

Mesela HTTP trafiği içerisinde "/bin/bash" stringi arayan bir saldırı imzası olsun. IDS sistemi 80.porta gelen giden her trafiği inceleyerek içerisinde /bin/bash geçen paketleri arar ve bu tanıma uyan paketleri bloklar. Eğer IDS sistemimiz paket birleştirme işlemini uygun bir şekilde yapamıyorsa biz fragroute veya benzeri bir araç kullanarak /bin/sh stringini birden fazla paket olacak şekilde (1. Paket /bin, 2.paket /bash)gönderip IDS sistemini atlatabiliriz.

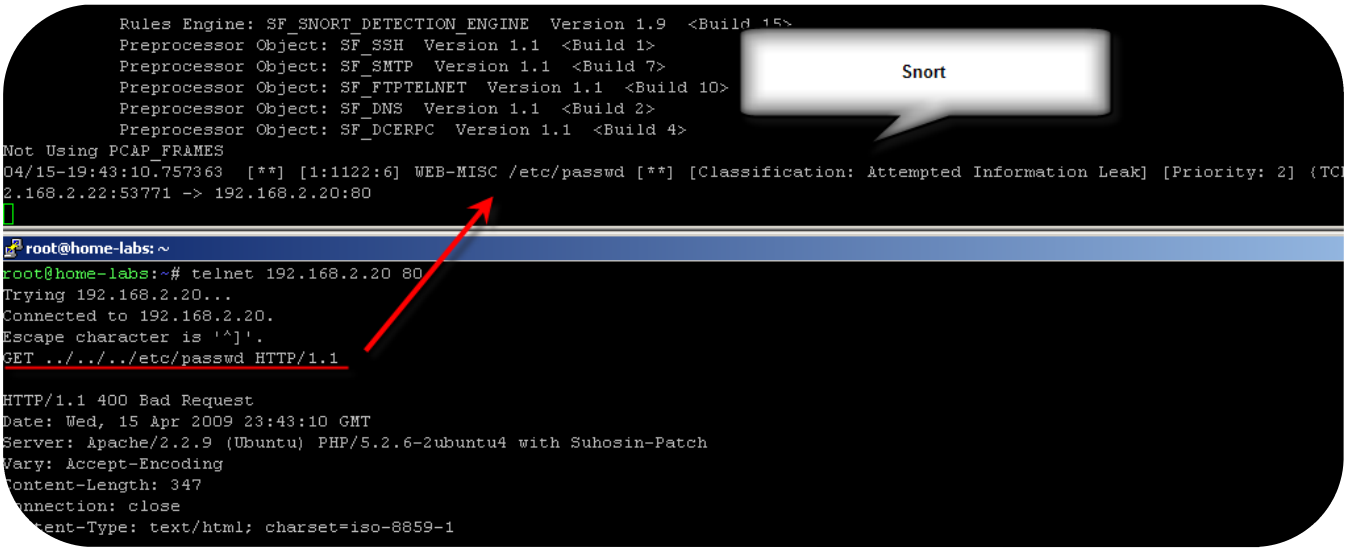
Snort ve Parçalanmış Paketler

Açık kaynak kodlu IDS/IPS yazılımı Snort parçalanmış paketler için sağlam bir çözüm sunmaktadır. Snort ile birlikte frag3 önişlemcisi kullanılarak IDS sistemine gelen parçalanmış paketler detection engine(Snort'da kural karşilaştirmasinin yapildiği kısım)gelmeden birleştirilerek yapilmaya çalışılan ids atlatma tekniklerini işe yaramaz hale getirilebilir. Frag3 önişlemcisine ek olarak Stream5 önişlemcisi de kullanılarak stateful bir yapı kurulur. Bu ikili sağlıklı yapılandırılmazsa Snort bir çok saldırıya açık hale gelir ve gerçek işlevini yerine getiremez.

Snort TCP reassembly modülü:

*preprocessor stream5_tcp: policy windows, use_static_footprint_sizes, *
ports client 21 23 25 42 53 80 135 136 137 139 143 110 111 445 465 513 691 1433 1521 2100 2301 3128 3306 8000 8080 8180 8888

Aşağıdaki çıktı Frag3 ve Stream5 önişlemcilerinin aktif olduğu durumda HTTP üzerinden yapılan *GET ../../../etc/passwd HTTP/1.1* isteğinin Snort tarafından yakalandığını gösteriyor. Aynı isteği parçalanmış paketlerle denersek Snort yine yakalar. Bunun nedeni frag3 eklentisinin networkten gelen paketleri önce birleştirip sonra Detection engine'e aktarmasıdır.



.

Sonuç

Parçalanmış paketler TCP/IP'nin en ilginç konularından biridir ve ilginç olduğu kadar da tehlikeli kullanımlara sahiptir. Bu yazıda parçalanmış paketler ve güvenlik sistemlerinin(Firewall/IPS) parçalanmış paketlere karşı nasıl davrandıklarını incelemeye çalıştım.

Bu yazıda anlatılan konuları uygulamalı olarak görmek ve kendiniz de denemek isterseniz www.guvenlikegitimleri.com adresinden Network Penetrasyon Testleri Eğitimine kayıt olmanızı tavsiye ederim.

Kaynaklar:

[1] TippingPoint IPS Signature Evasion through Packet Fragmentation (http://www.milw0rm.com/papers/302)

http://www.openbsd.org/faq/pf/scrub.html
The Tao of Network Security Monitoring
http://www.snort.org/docs/snort_htmanuals
http://citeseer.ist.psu.edu/ptacek98insertion.html
http://www.monkey.org/~dugsong/fragroute/)