MSN PROTOKOLÜNÜ ANALİZ ETMEK

BELGE HAKKINDA

Bu belgedeki eksik, yanlış ya da geliştirilmesi gerektiğini düşündüğünüz yerleri e-posta yoluyla bildirebilirsiniz.

Belgenin ilk oluşturulma tarihi: 10 Haziran 2007

/***
* Cihan KÖMEÇOĞLU *
* cihan [at] akademi.enderunix.org *
* *
* EnderUNIX Yazılım Geliştirme Takımı *
* *
* http://www.enderunix.org *
* *
* Sürüm : 1.0 *
* *
* Tarih : 10.06.2007 *
* *
* Etiketler : msn, sniff,protokol *
* *
* Seviye : Başlangıç - Orta Seviye *
* *
* Makalenin en yeni versiyonu : http://www.enderunix.org/docs/msnsniff.pdf adresinden elde edilebilir. *
* *
***/

MSN PROTOKOLÜNÜ ANALİZ ETMEK

1.Giriş

Msn sunucusuna yapılan tüm bağlantılar TCP/IP üzerinden olmaktadır.Msn Messenger 1863. portu kullanmaktadır.Sunucuya yapılan bağlantı asenkrondur. Yaptığınız bir işlem için sunucudan cevap beklemek zorunda değilsiniz.Mesajlaşmalar sunucu üzerinden yapılmaktadır.Sunucu ile soket üzerinden bağlantı olduğu gibi http protokolü ile de bağlantı olabilmektedir.Mesajlaşmalar tcp paketleri ile olmaktadır.

2. Msn Paketlerini dinlemek için yapılması gereken işlemler

Burada anlatılanlar ağınızda bulunan bir ağgeçidinden geçen paketleri dinleyebilmek içindir.Bir dinleme işlemini yapabilmeniz için pakelerin dinleme yapılacağı makineden geçmek zorundadır.

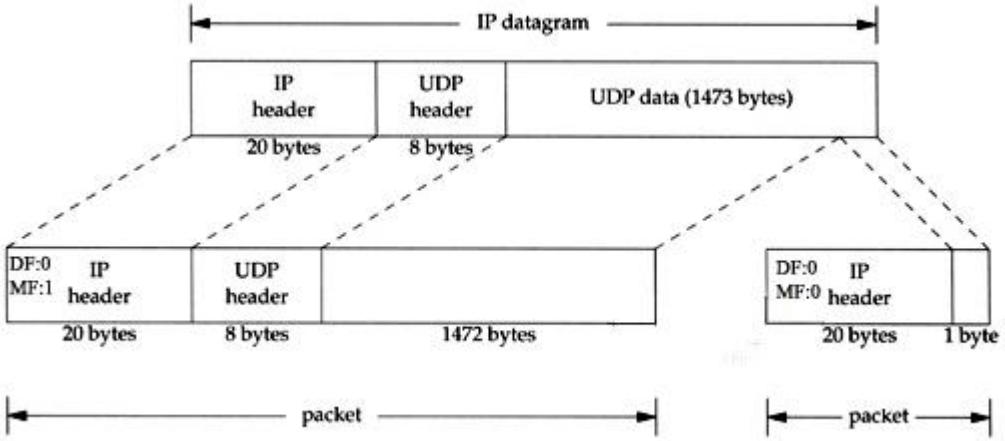
Her bir paketi yakaladıktan sonra bu paketlerin tcp ve ip başlıklarının uzunluğuna bakmamız gerekir. Normal bir paketin ip ve tcp başlık uzunluğu minimum 20 byte olması gerekir. Yani 20 bytedan daha düşük olan paketleri dikkate almamamız gerekir. Tcp ve ip başlık uzunluklarının tutulduğu alan 4 bit'dir ve uzunluğun ¼'ü kadar tutulmaktadır.20 byte'lık bir paket için bu alana 5 yazılmaktadır.Biz buradaki değeri 4 ile çarptığımızda paketin başlık uzunluğunu elde ederiz.

Paketleri yakaladıktan sonra bu paketlerin hangilerinin msn paketleri olduğunu belirlememiz gerekir. Bunu da port numarasına bakarak yapabiliriz. Aslında port numarasına bakarak bunu tam olarak belirleyemeyiz.Çünkü aynı portu kullanan başka uygulamalarda olabilir.İp numarasına göre de yapamayız çünkü her bağlantı için hemen hemen ip numarası farklı olabilmektedir. Bu yüzden performans için en iyi yöntem port numaralarına bakarak dinleme yapmaktır.Bunun içinde msn sunucusunun kullandığı 1863. port numarasını kullanarak yapacağız.Msn messenger tcp portokolünü kullandığı için tcp katmanındaki hedef ve kaynak portların 1863 olup olmadığına bakacağız. Kaynak portu 1863 olan paketler msn sunucusundan gelen paketler, hedef portu 1863 olan paketler ise istemciden sunucuya giden paketlerdir.

Paketlerin port numarasına baktıktan sonra bunların bir tcp paketi olup olmadığını araştıracağız. Bunun içinde ip katmanındaki protokol bilgisine bakmamız gerekir. Bu bize ip katmanından sonraki bir üst protokolün hangi protokol olduğunu göstermektedir.

Paketlerin tcp paketi olduğuna karar verdikten sonra bu paketlerin veri taşıyıp taşımadığını bulmamız gerekir.Bunu yapmak için paketlerin büyüklüğünden yararlanacağız. Ip katmanında paketin toplam uzunluğunun tutulduğu 16 bitlik bir alan bulunmaktadır. Bu alan paketin ip başlık uzunluğu ,tcp başlık uzunluğu ve verinin büyüklüğünün toplamıdır.Yani minimum başlık uzunlukları olan ve 12 byte bir veri taşıyan paket için toplam uzunluk alanındaki değer 52'dir. 20 byte ip başlık uzunluğu, 20 byte tcp başlık uzunluğu ve 12 byte verinin toplamıdır. Burada bu paketin veri taşıyıp taşımadığını anlamak için tcp ve ip başlık uzunluklarını toplayıp ip katmanındaki toplam uzunluk değerinden çıkarmamız gerekir. Yani az önceki örnekte 52 byte'lık bir paket için 20+20 =40 byte ve 52-40=12 byte veri olduğunu buluruz.
Bu işlemden sonra çıkan değer sıfırdan büyük ise bu paketin veri taşıdığını anlarız ve buradaki veriyi alarak msn protokolüne ait bilgiyi elde etmiş oluruz. Yalnız Bu fragmentation olan paketler için ek işlemler yapmamız gerekir.Makalenin devamında Ip fragmentation ve reordering işlemlerini anlatacağım.

Ip fragmentation

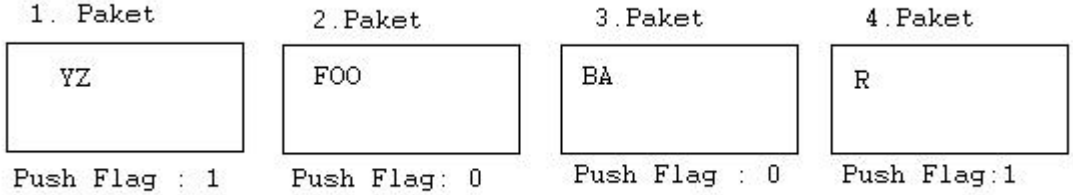


Fragmentation olan paketlerdeki verileri birleştirebilmemiz için ip katmanındaki more fragment biti ,dont fragment biti ve identification kısmına bakmamız gerekir. İlk olarak dont fragment bitine bakacağız. Eğer bu dont fragment biti 1 ise bu verinin fragmentation yapılmadığını gösterir. Eğer 0 ise verinin fragmentation yapıldığını göstermeyebilir. Bu sırada more fragment bitine de bakmamız gerekir.Eğer more fragment biti 1 ise bunun fragmentation yapıldığını gösterir.Ama verinin son parçasında more fragment biti 0 olacağı için identification kısmına da bakmamız gerekir.Bir identification lisetesi tutup her bir fragmantation parçasının identification kısmına bakarak karşılaştırmamız gerekir. Böylelikle fragmentation olan paketleri anlayabiliriz. Verileri birleştirirken dikkat etmemiz gereken şey ise verinin ilk parçası taşıyıcı katmanından (transport layer) sonra geldiği, bundan sonraki parçalarda verinin bulunduğu kısım hemen ip katmanından sonra gelmektedir.

Tcp paket reordering & reassembly

Msn mesajları parçalara ayrılarak yollanabilir. Bizim yapmamız gereken bunları birleştirmektir. Bunun içinde tcp katmanındaki push flag'inden yararlanacağız. Tcp push flag'i bize pakette veri olup olmadığını göstermektedir . Yalnız bu reordering & reassembly gerektiren paketler için geçerli değildir. Yani verinin parçalarını taşıyan ilk pakette push flag 1 olmakta sonraki parçalarında ise push flag 0 olmaktadır. Biz bunu kullanarak verileri birleştireceğiz. Yapmamız gereken sadece ikinci bir push flagi 1 olan paketleri görene kadar verileri buffer etmektir.

Aşağıdaki şekilde olduğu gibi 1. paketin push flagi '1' , 2. ve 3. paketin push flagi '0' , 4. paketin push flagi ise '1' değerindedir. Yukarıda anlatıldığı gibi ikinci bir push flagi set edilmiş paket görene kadar verileri birleştirmemiz gerekir.Yani 2,3,4 nolu paketlerin verileri birleştirilmesi gerekir. 1. paket başka bir verinin parçasını taşımaktadır. 2,3,4 nolu paketleri birleştirdiğimizde biz "FOOBAR" değerini elde edeceğiz.



3. Msn Protokolü

Kısaca dinlemek için yapılması gereken adımları anlattıktan sonra kendiniz basit bir sniffer yazacak kadar msn protokolünden bahsedeceğim.

CALL: Bir kişi ile sohbet etmek istediğiniz zaman CALL mesajı yollanır. Karşı tarafta buna JOI mesajı ile cevaplar. Bu durumda iki kişi sohbete başlayabilir.Çağrılmak istenen kişinin mail adresi parametre olarak yollanmaktadır.Aşağıdaki mesajın en sonunda \r\n karakteri görülmektedir. Bu her mesajın sonunda bulunmaktadır. İlgili mesajdaki satırın bittiğini göstermektedir.

Örnek Bir CALL mesajı

```
CAL 2 ornek@hotmail.com
```

JOI: Call mesajına karşılık cevap olarak verilir. JOI mesajında parametre olarak sohbete katılan kişinin mail adresi bulunmaktadır.Böylelikle sohbet başlayabilir.

```
JOI ornek@hotmail.com Name_123\r\n
```

OUT: Sunucu ile bağlantı kapatılmak istendiğinde yollanır. Herhangi bir paramteresi bulunmamaktadır.

```
OUT \r\n
```

BYE:İki türlü BYE mesajı bulunmaktadır.Birincisi bir kişi out mesajını yollayıp sunucu ile bağlantısını kopardığı zaman kişinin oturumunda bulunan online kişilere yollanarak offline olduğu bildirilir.Bu mesaj aşağıdaki gibidir.

```
BYE ornek@hotmail.com \r\n
```

Diğer bye mesajı ise sohbet sonlandırıldığında yollanır. Diğer bye mesajından farklı olarak en son parametresi 1 olmaktadır.

```
BYE ornek@hotmail.com 1 \r\n
```

MSG: Mesaj yollanmak istendiğinde server'a yollanan mesaj tipidir. Parametre olarak kişinin yolladığı mesaj, msn client programının adı, karakter kodlaması, kişinin yazdığı mesaj bulunmaktadır.MSG mesajlarının bazıları mail adresi içerebilmektedir. Mail adresi içeren msg mesajları sohbet yapılan kişiden gelen mesajlardır.Buradaki mail adresi mesajı yollayan kişinin mail adresidir. Mail adresi içermeyen msg mesajları ise sohbet yaptığımız kişiye yollanılan mesajlardır. Aşağıdaki örnekten birincisi sohbet yapığımız kişiye yollanılan mesajdır.İkincisi ise cevap olarak gelen mesajdır.Mesajın en sonunda \r\n\r\n karakterleri görülmektedir.Bu yazılan mesajın başlayacağını göstermektedir.\r\n olanları ise satırın bittiğini göstermektedir.

Örnek bir MSG mesajı:

```
MSG 4 N 100\r\n
MIME-Version: 1.0\r\n
Content-Type: text/plain; charset=UTF-8\r\n
X-MMS-IM-Format: FN=Arial; EF=I; CO=0; CS=0; PF=22\r\n\r\n
Merhaba Nasılsın
```

MSG ornek@hotmail.com isim 133\r\n
MIME-Version: 1.0\r\n
Content-Type: text/plain; charset=UTF-8\r\n
X-MMS-IM-Format: FN=Arial; EF=I; CO=0; CS=0; PF=22\r\n

\r\nİyilik

Aşağıdaki mesaj ise kullanıcının bir mesaj yazdığını göstermektedir.

MSG 7 U 91\r\n
MIME-Version: 1.0\r\n
Content-Type: text/x-msmsgscontrol\r\n
TypingUser: ornek@hotmail.com\r\n

NLN: Msn messenger'daki kullanıcının durum bilgisini göstermektedir.

NLN AWY ornek@hotmail.com ornek%20display%20name 268435492

- **NLN** - *Available*
- **IDL** - *Idle*
- **PHN** - *On the Phone*
- **LUN** - *Out to Lunch*
- **BSY** - *Busy*
- **AWY** - *Away*

XFR: Notification server'dan switchboard server'a yönlendirme yapabilmek için istemcinin gönderdiği mesajdır. Server bu mesaja yine XFR ile cevap verir.

- Mesajın ilk parametresi switchboard server'a yönlendirme yapılacağını bildirmektedir.
- İkinci parametresi switchboard server'ın ip adresidir.
- Üçüncü parametresi kimlik doğrulama tipidir. Her zaman CKI olur.
- Dördüncü parametre ise kimlik doğrulama stringidir.Switchboard server'a bağlantı kurulacağı zaman kimliğini kanıtlayabilmesi için bu string gereklidir.

>>> XFR 15 SB\r\n

<<< XFR 15 SB 207.46.108.37:1863 CKI 17262740.1050826919.32308\r\n

IRO: Switchboard server ile bağlantı kurulduktan sonra size listenizdeki kişiler IRO mesajı ile yollanır.

- IRO mesajının ilk parametresi size gönderilen kaçıncı IRO mesajı olduğunu göstermektedir.Birinci IRO mesajı için 1, ikinci IRO mesajı için ise 2 olmaktadır.
- İkinci parametresi ise toplam IRO mesajının sayısıdır.
- Üçüncü parametresi listenizdeki kişinin hesap adıdır.
- Dördüncü parametresi ise kişinin görünen adıdır.

<<< IRO 1 1 2 example@passport.com Mike\r\n

<<< IRO 1 2 2 myname@msn.com My%20Name\r\n

Kaynaklar

[1.]Richard Stevens TCP/IP Illustrated, Volume1: The Protocols

[2.]http://www.hypothetic.org/