

WIRESHARK KULLANICI REHBERİ



Hazırlayan :
e-m@il :

Haktan AKPINAR
r3db4r0n@hotmail.com

İÇİNDEKİLER

<u>Wireshark Hakkında.....</u>	<u>3</u>
<u>Özellikler.....</u>	<u>3</u>
<u>Kurulum.....</u>	<u>6</u>
<u>Kaynak kodundan Kurulum.....</u>	<u>6</u>
<u>Paket Yöneticileriyle Kurulum.....</u>	<u>7</u>
<u>Windows altında Kurulum.....</u>	<u>7</u>
<u>Kullanıcı arayüzü.....</u>	<u>9</u>
<u>Menüler.....</u>	<u>10</u>
<u>File</u>	<u>10</u>
<u>Edit</u>	<u>12</u>
<u>View.....</u>	<u>14</u>
<u>Go.....</u>	<u>15</u>
<u>Capture.....</u>	<u>16</u>
<u>Anaylze.....</u>	<u>19</u>
<u>Statistic.....</u>	<u>22</u>
<u>Filtreler..</u>	<u>26</u>
<u>Capture Filter.....</u>	<u>26</u>
<u>Display Filter.....</u>	<u>28</u>
<u>Karşılaştırma Değerleri.....</u>	<u>30</u>
<u>Görüntüleme Filtresi Mantıksal Operatörleri.....</u>	<u>31</u>
<u>Örnek İfadeler.....</u>	<u>33</u>
<u>Paketlerle Oynamak.....</u>	<u>34</u>
<u>Örnek Sorunlarda Wireshark Kullanımı.....</u>	<u>35</u>
<u>Torrent Sorunu.....</u>	<u>35</u>
<u>Wireshark ile Veri Madenciliği.....</u>	<u>36</u>
<u>Arp Poisoning Tespiti.....</u>	<u>37</u>
<u>Blaster Wormu Tespiti.....</u>	<u>38</u>
<u>Port Tarama Tespiti.....</u>	<u>39</u>
<u>Syn Flooding Tespiti.....</u>	<u>39</u>
<u>Bağlantı Sorunu Tespiti.....</u>	<u>40</u>
<u>Casus Yazılım Tespiti.....</u>	<u>41</u>
<u>Os Fingerprinting Tespiti (Icmp Tabanlı).....</u>	<u>42</u>
<u>Neden SSH.....</u>	<u>42</u>
<u>Messenger Üzerine.....</u>	<u>44</u>
<u>TCP Oturumuna Müdahale Üzerine</u>	<u>44</u>
<u>FTP Saldırı Tespiti.....</u>	<u>47</u>
<u>Son Sözler</u>	<u>49</u>
<u>Kaynaklar.....</u>	<u>49</u>

HAKKINDA

Wireshark GNU General Public Licence (GPL) altında yayımlanan çok güçlü özelliklere sahip bir açık kaynaklı paket analiz yazılımıdır. Wireshark ağınıza karşı herhangi bir saldırı durumunda sizi uyaracak saldırı tespit sistemi değildir. Farklı bir durum olduğunda sorunun ne olduğunu farketmenize yardımcı olur.

ÖZELLİKLER

- *Windows ve *nix sistemlerde çalışabilir.
- *Ağ arabiriminden eş zamanlı paket yakalayabilir.
- *Paketleri çok ayrıntılı bir şekilde protokol bilgileriyle görüntüler
- *Yakaladığı paketleri kaydetme imkanı vardır
- *Kriterlere göre paket filtreleme mevcuttur
- *Kriterlere göre paket arama mevcuttur
- *paket görünümüleri renklendirilerek kullanım kolaylaştırılabilir.
- *Çeşitli istatistikler yapabilir
- *...ve daha birçoğu

Wireshark farklı ağ türlerinde yakalama yapabilir. Desteklenen medya türleri <http://wiki.wireshark.org/CaptureSetup/NetworkMedia> adresinde gösterilmektedir.

Wireshark diğer paket yakalama yazılımlarının farklı formatlardaki dosyalarını açabilir.

- libpcap, tcpdump ve tcpdump formatındaki diğer araçlar.
- Sun snoop ve atmsnoop
- Shomiti/Finisar
- Novell LANalyzer
- Microsoft Network Monitor
- AIX's iptrace
- Cinco Networks NetXray
- Network Associates Windows-tabanlı Sniffer ve Sniffer Pro
- Network General/Network Associates DOS-tabanlı Sniffer (sıkıştırılmış yada sıkıştırılmamış)
- AG Group/WildPackets
- EtherPeek/TokenPeek/AiroPeek/EtherHelp/PackageGrabbe
- RADCOM's WAN/LAN Analyzer
- Network Instruments Observer version 9
- Lucent/Ascend router debug çıktısı
- HP-UX nettl
- Toshiba ISDN routers dump çıktısı
- ISDN4BSD i4btrace aracı
- EyeSDN USB S0
- Cisco Secure Intrusion Detection System' den IPLog formatı
- pppd logs (pppdump format)
- VMS's TCPIPtrace/TCPtrace/UCX\$TRACE araçları çıktısı

- DBS Etherwatch VMS utility çıktısı
- Visual Networks' Visual UpTime traffic
- CoSine L2 debug çıktısı
- Accellent's 5Views LAN agents çıktısı
- Endace Measurement Systems' ERF formatı
- Linux Bluez Bluetooth stack hcidump -w traces
- Catapult DCT2000 .out dosyası

---Wireshark diğer paket yakalama yazılımlarının açabileceği formatta çıktılar üretir.

- libpcap, tcpdump ve tcpdump formatındaki diğer araçlar (*.pcap,*.cap,*.dmp)
- 5Views (*.5vw)
- HP-UX nettl (*.TRC0,*.TRC1)
- Microsoft Network Monitor - NetMon (*.cap)
- Network Associates Sniffer - DOS (*.cap,*.enc,*.trc,*.fdc,*.syc)
- Network Associates Sniffer - Windows (*.cap)
- Network Instruments Observer version 9 (*.bfr)
- Novell LANalyzer (*.tr1)
- Sun snoop (*.snoop,*.cap)
- Visual Networks Visual UpTime traffic (*.*)

Microsoft Windows

- Windows 2000, XP Home, XP Pro, XP Tablet PC, XP Media Center, Server 2003 yada Vista
- 32-bit Pentium (tavsiye olunan: 400MHz ve üstü), 64-bit WoW64
- 128MB RAM (tavsiye olunan: 256MBytes ve üstü)
- 75MB boş disk alanı
- 800*600 (1280*1024 ve üstü tavsiye olunur) çözünürlük 65536 (16bit)

Desteklenen network kartları

- Ethernet: Windowsun tanıdığı herhangi bir kart olabilir.
- WLAN: MicroLogix destek listesine bakınız,
- Diğer türler için: <http://wiki.wireshark.org/CaptureSetup/NetworkMedia> adresine bakınız

64-bit işlemcilerde Wireshark 32 bit emulasyonu olarak çalışır .Bunun için WinPcap 4.0 gereklidir

---Trafik yoğun olan ağlarda yüksek işlemci gücü, fazlaca ram ve disk alanına sahip olunması tavsiye olunur.

---**Wiresharkın çökmesi durumunda ayrıntılar için**

<http://wiki.wireshark.org/KnownBugs/OutOfMemory> adresine bakınız.

Unix / Linux

Wireshark birçok unix platformunda çalışmaktadır.

- Apple Mac OS X
- Debian GNU/Linux
- FreeBSD
- Gentoo Linux
- HP-UX
- Mandriva Linux
- NetBSD
- OpenPKG
- Red Hat Fedora/Enterprise Linux
- rPath Linux
- Sun Solaris/i386
- Sun Solaris/Sparc

Sizin platformunuza uygun paket mevcut değilse kaynak kodu indirip kurabilirsiniz.

Programın son versiyonunu Wireshark websitesinden indirebilirsiniz:

<http://www.wireshark.org/download.html>

Problem bildirme ve yardım almak için :

Eğer wiresharkla ilgili probleminiz varsa ve yardım istiyorsanız..

*Birçok kullanışlı bilgiyi wireshark web sayfasında bulabilirsiniz

<http://www.wireshark.org>

*Wireshark wiki sayfası <http://wiki.wireshark.org> wireshark ve paket yakalama konularında geniş bir yelpazede bilgi sunmaktadır.

*FAQ : <http://www.wireshark.org/faq.html>

*Posta listeleri

wireshark-announce : Wireshark için yeni sürüm duyuruları yapar

wireshark-users : Kurulum ve kullanım sorunları ve çözümlerini içerir.

wireshark-dev : Geliştiriciler içindir

--Unix/linux platformlarında hata bildirmek için programın backtrace çıktısını bir dosyaya yazıp wireshark-dev@wireshark.org adresine postalayın.

KURULUM

Kurulum için gereken dosyaları <http://www.wireshark.org> adresinden elde edebilirsiniz.

Kaynak kodundan kurulum için :

GTK+ ve GLib nin kurulu olduğundan emin olun (eğer yoksa <http://www.gtk.org> adresinden temin edebilirsiniz)

Kaynaktan GTK+ kurulumu için:

```
#gzip -dc gtk+-1.2.10.tar.gz | tar xvf -  
<çıktı,çıktı,çıktı..>  
#cd gtk+-1.2.10  
#./configure  
<çıktı,çıktı,çıktı..>  
#make  
<çıktı,çıktı,çıktı..>  
#make install  
<çıktı,çıktı,çıktı..>
```

*nix platformlarında libpcap kütüphanelerinin kurulu olduğundan emin olun.Son versiyonunu <http://www.tcpdump.org> adresinden indirebilirsiniz.

Kaynaktan libpcap kurulumu için:

```
#gzip -dc libpcap-0.9.4.tar.Z | tar xvf -  
<çıktı,çıktı,çıktı..>  
#cd libpcap-0.9.4  
#./configure  
<çıktı,çıktı,çıktı..>  
#make  
<çıktı,çıktı,çıktı..>  
#make install  
<çıktı,çıktı,çıktı..>
```

--Wireshark klasörü içinde ./configure komutunu verin.

--make komutunu verin.

--Son olarak make install koutunu verin.

Configure basamağında sorun olursa sebebi için kaynak dizin içinde configure.log dosyasına bakınız.Muhtemelen sisteminizde GTK+ yoktur yada uygun versiyonda değildir. Sisteminizde libpcap bulunmadığı durumda yine configure basamağında

hata verecektir.

Eğer sorunu tanımlayamassanız wireshark-dev e-posta listelenerine config.log çıktısıyla danışabilirsiniz.

Paket yöneticileriyle kurulum

Debian için: # apt-get install wireshark

Redhat için: # rpm -ivh wireshark-0.99.5.i386.rpm

Gentoo için: # USE="adns gtk ipv6 portaudio snmp ssl kerberos threads
selinux" emerge wireshark

FreeBSD için: # pkg_add -r wireshark

Debian için bağımlı olduğu paketler:

b0x:/etc/init.d# apt-cache depends wireshark

wireshark

Depends: libadns1

Depends: libatk1.0-0

Depends: libc6

Depends: libcairo2

Depends: libcomerr2

Depends: libgcrypt11

Depends: libglib2.0-0

Depends: libgnutls26

Depends: libgtk2.0-0

Depends: libkrb53

Depends: libpango1.0-0

Depends: libpcap0.8

Depends: libpcre3

Depends: libportaudio2

Depends: wireshark-common

Depends: zlib1g

Recommends: gksu

Conflicts: ethereal

Replaces: ethereal

Windows altında kurulum:

<http://www.wireshark.org/download.html#releases> adresinden Windows setup
ını indirerek kurabilirsiniz.

http://www.mesutsariyar.com/index.php?view=article&catid=1%3Alatest-news&id=52%3Awireshark-ile-msn-capture-ve-ip-bulmak&tmpl=component&print=1&page=&option=com_content&Itemid=50

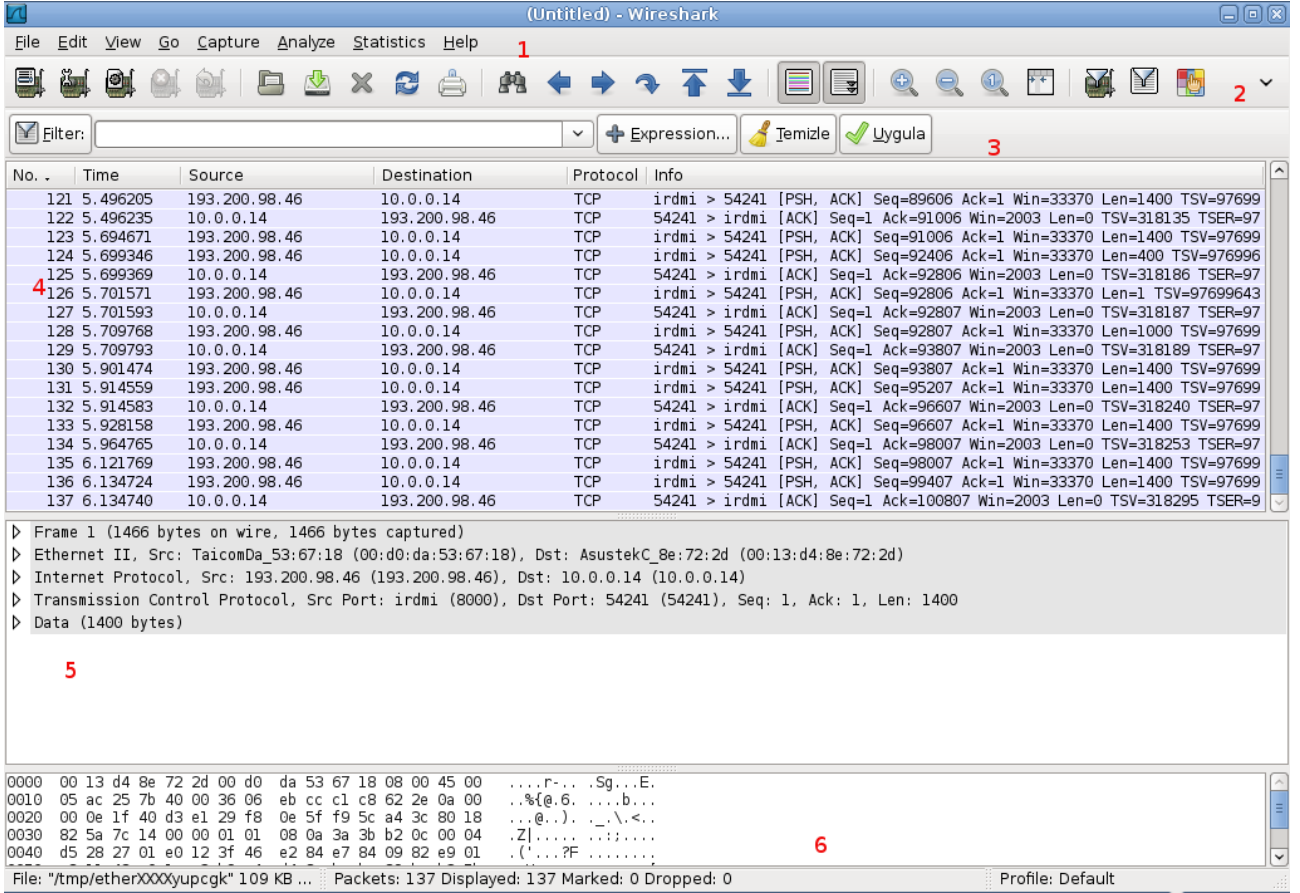
Adresinde windows altında kurulumu ayrıntılı şekilde anlatılmıştır.

GTK2 grafik arayüzde sorun olursa GTK1 kullanabilirsiniz.256 renk windows GTK2 ile düzgün çalışmayacaktır.Buna karşılık GTK1 ile ileri analiz ve istatistiksel özellikleri kullanamayacaksınız.

Plugins / Extensions (Wireshark ve TShark inceleme motoru için)

- **Dissector Plugins** - eklentilerle birlikte bazı genişletilmiş incelemeler içerir.
- **Tree Statistics Plugins** - eklentilerle birlikte bazı genişletilmiş istatistikler içerir.
- **Mate - Meta Analiz ve Tracing engine (deneysel)** – Ayarlanabilir ilave görüntüleme filtresi motoru, ayrıntı için: <http://wiki.wireshark.org/Mate>
- **Daha ayrıntılı SNMP incelemesi için** SNMP MIBs - SNMP MIBs.
- **Tools** (yakalanan dosyalarla çalışmak için ilave komut satırı aracı):
- **Editcap** - Editcap yakalanan dosyaları okur ve başka dosyalara paket yazar.
- **Text2Pcap** - Text2pcap ASCII hex dump okur ve veriyi libpcap-style yakalanan dosya içine yazar.
- **Mergecap** - Mergecap çoklu kayıt dosyalarını tek çıktı dosyasında birleştirir.
- **Capinfos** - Capinfos yakalanan dosyalar hakkında bilgi sağlar.

Wireshark kullanıcı arayüzüne kısaca değinirsek;



1 Menü çubuğu

2 Araç çubuğu

3 Görüntüleme filtresi çubuğu

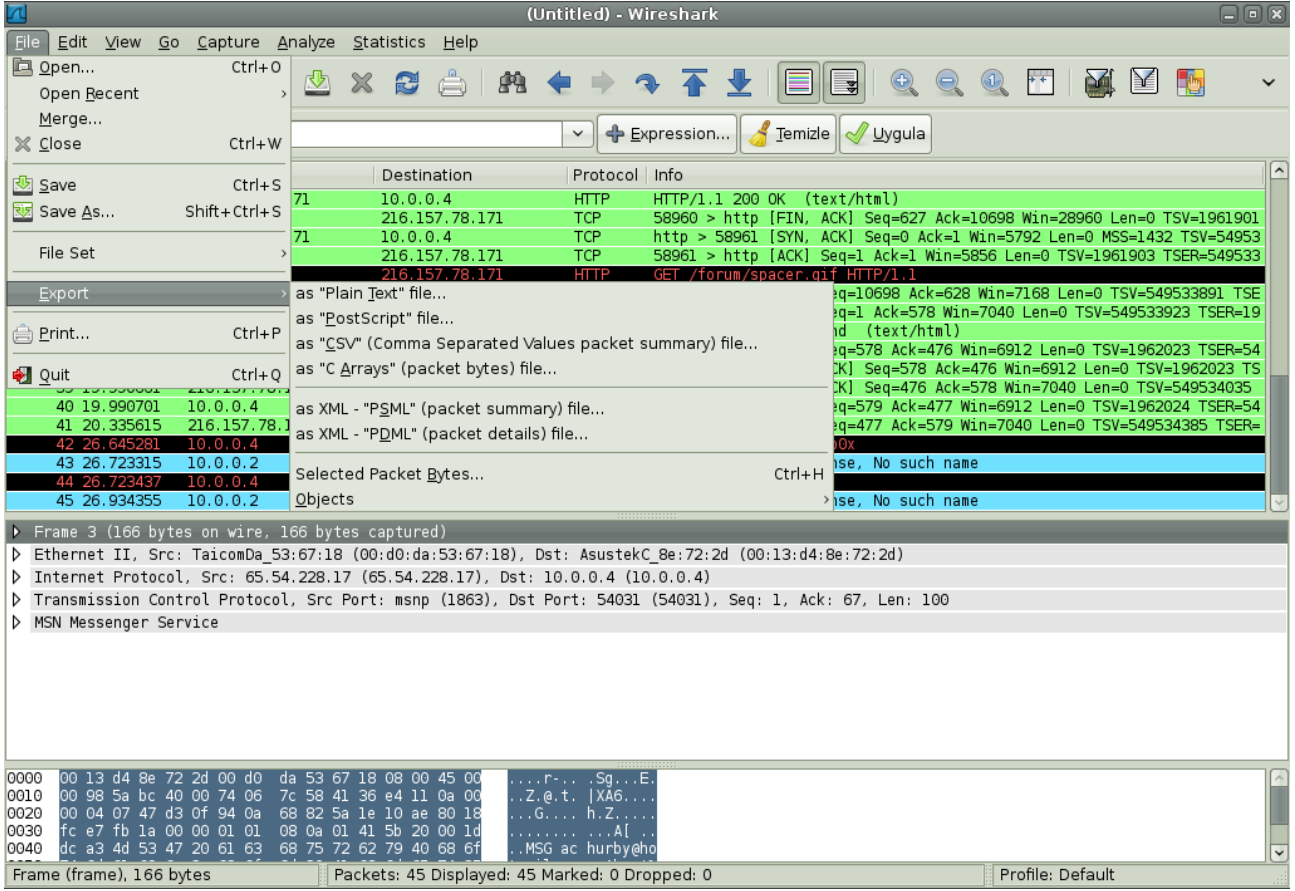
4 Özet alanı

5 Protokol ağacı alanı

6 Data alanı

MENÜLER

Dosya Menüsü



Open (ctrl+O): Hazırda var olan önceden kaydedilmiş wireshark ya da başka desteklediği paket analiz yazılımlarının ürettiği dosyaları görüntülemek için kullanılır.

Open Recent: Son kullanılan dosyaları açmada kolaylık sağlar.

Merge: Kaydedilmiş dosyaları birleştirmede kullanılır.

Close (ctrl+W): Açık olan dosyadan çıkar.

Save(ctrl+S): Görüntülenmekte olan paketleri kaydeder.

Save As(ctrl+shift+S): Farklı kaydeder.

File Set:

List Files: Dosya listesini oluşum tarihi, son değişim tarihi, boyutu şeklinde dosya dizisi içerisinde gösterir

Next File: Dosya dizisi içinde varolanı kapatıp sonrakine atlar

Previous File: Dosya dizisi içinde varolanı kapatıp bir öncekine atlar
Export:

As "Plain Text" File: Toplanan paketleri metin dosyası olarak dışa aktarmaya yarar.Özet ve ayrıntı bölümlerini aktarır.

As "PostScript" File: İstenen paketleri postscript dosyası olarak dışa aktarmaya yarar.Wireshark seçilen ayrıntı bölümü bilgilerini aktarır.

As "CSV" (Comma Separated Values packet summary) File: Wireshark özet bölümündeki bilgileri virgülle ayrılmış şekilde düz metin dosyası olarak dışa aktarır.

As "C Arrays" Paket Bytes File: Paket veri değerlerini hex byteleri olarak aktarır.

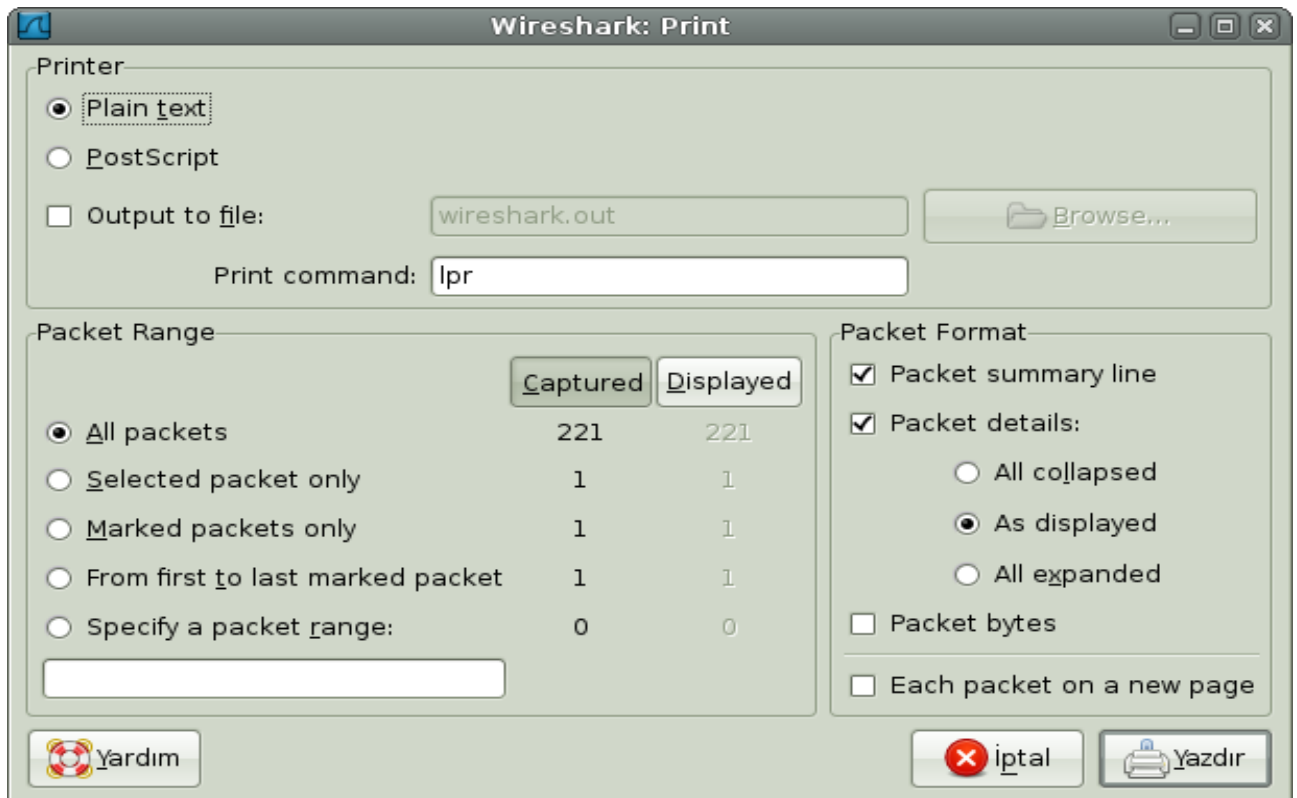
As XML -"PSML" (Packet Summary) File: Paketleri PSML (packet summary markup language) XML dosya formatında dışa aktarmaya yarar.

As "PDML" File: Paketleri PDML (packet details markup language) XML dosyası olarak aktarmaya yarar.

Selected Packet Bytes :

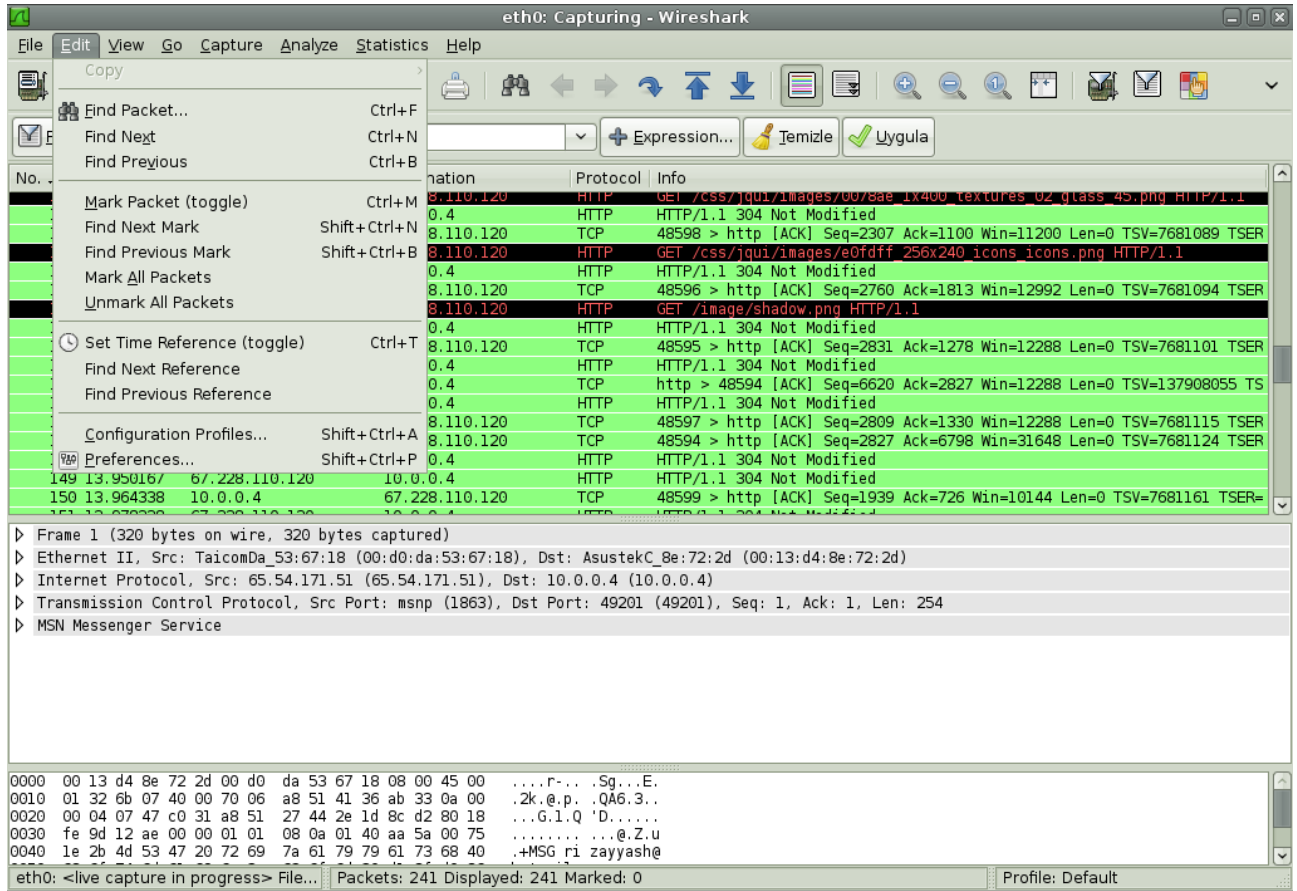
Objects --> http : Paketler içerisinde http protokollü paketleri ve objelerini dışarı aktarmaya yarar.

Print (ctrl+P):Seçilen paketleri yazdırmaya yarar.



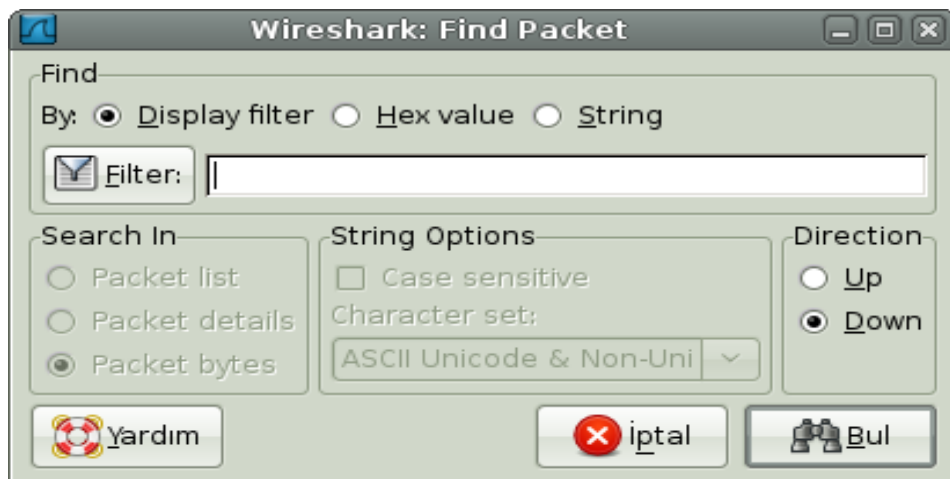
Quit(ctrl+Q):Programdan çıkar.

EDIT



Copy (Shift+ctrl+C): Veri bölmesinden tıklanan değeri filtre ifadesi olarak kopyalar. İstenilen bölüm seçilerek sağ fare menüsünden de yapılabilir .

Find Packet (ctrl+F): Birçok kritere göre arama yapmanıza imkan sağlar.Display filter seçeneği seçiliyse wireshark filtreleme kriterlerine göre arama yapar.Basit protokol taramalarından kuvvetli filtreleme ifadelerine kadar birçok türde etkin arama yapabilirsiniz.



Hex Value: Paket veri kümesi içerisinde belirtilen hex değerlerinde arama yapar.

String: Girilen string i liste,ayrıntı ya da veri alanlarında belirlenen kriterlere göre arar.String options alanından arama büyük küçük harf duyarlı ve karakter seti belirtilerek yapılabilir.Direction alanında ise taramanın aşağı yada yukarı yapılacağı belirtilir.

Find Next (ctrl+N): Belirlenen kriterde bir sonraki paketi bulur.

Find Previous (ctrl+B) : Belirlenen kriterde bir önceki paketi bulur.

Mark Packet (ctrl+M) : Seçilen paketi işaretler

Find Next Mark (shift+ctrl+N) : Bir sonraki işaretli paketi bulur.

Find Previous Mark (shift+ctrl+B) : Bir önceki işaretli paketi bulur.

Mark All Packet: Bütün paketleri işaretler.

Unmark All Paket: Bütün işaretleri kaldırır.

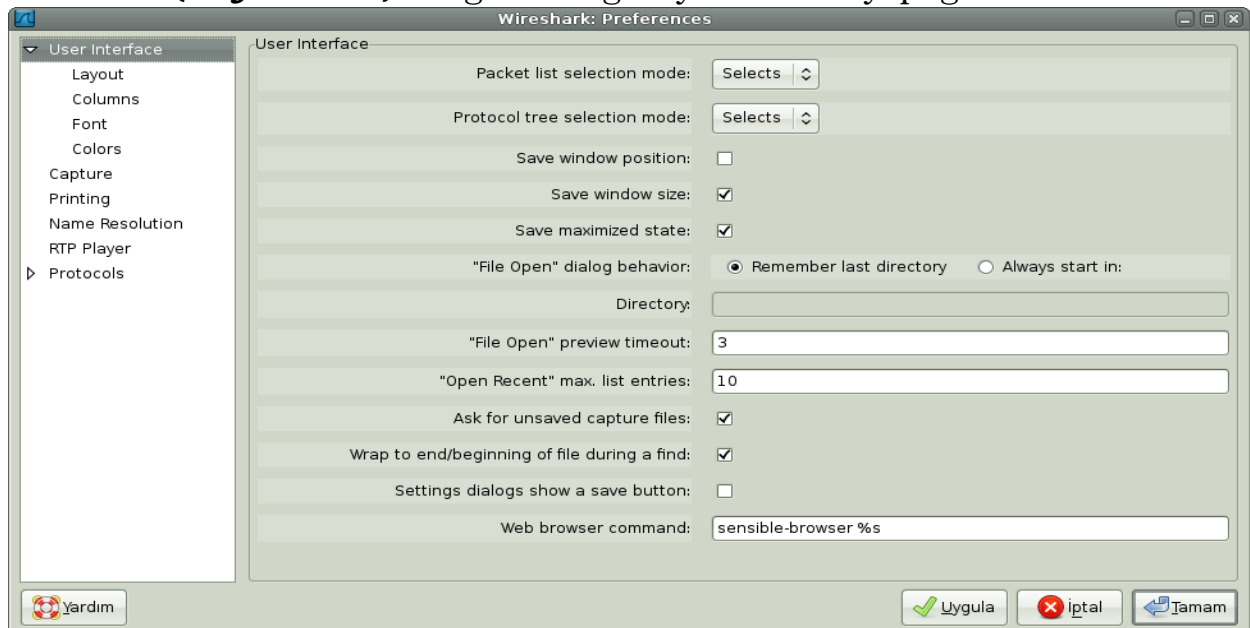
Set Time Reference (ctrl+T): Seçilen paketi zaman referansı olarak alır ve sonraki paketlerde o pakete göre zaman değerleri alır.

Find Next Reference: Bir sonraki referans alınan paketi bulur.

Find Previous Reference: Bir önceki referans alınan paketi bulur.

Configuration Profiles (shift+ctrl+A): Profil ekle-sil işlemlerini yapar.

Preference (shift+ctrl+P): Programla ilgili ayarlamaları yaptığımız bölümdür.



User Interface bölümünde program için pencere düzeni, renk, font ayarlamaları ve bunlar gibi görünümü kişiselleştirmeye yarayan seçenekler bulunur.

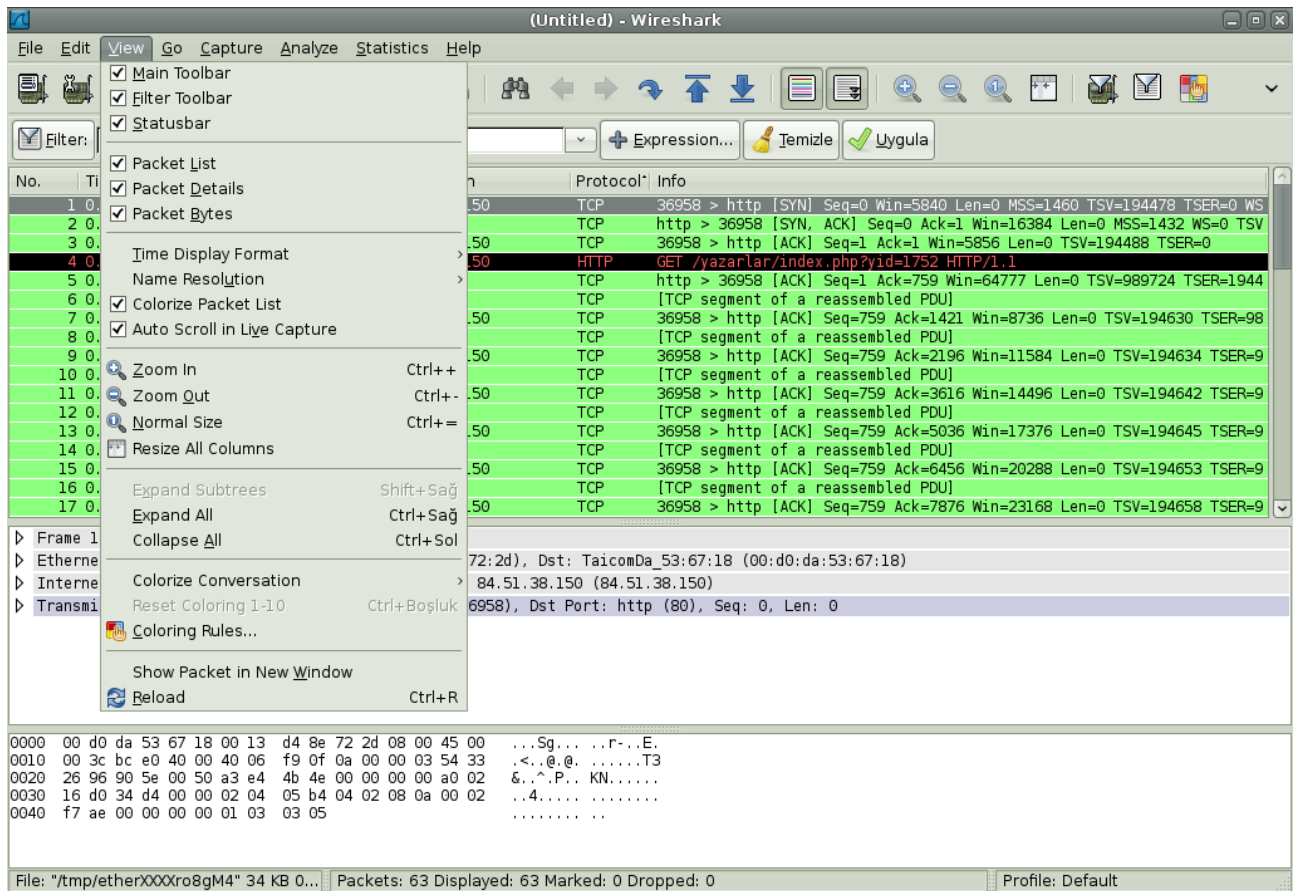
Capture : Paketleri yakalamak için kullanılacak default ağ arabirimi, eş zamanlı paket görüntüleme ve promiscuous mod seçimi (promiscuous mod : Yönlendirme olmadan bütün paketlerin bütün istemcilere dağıtıldığı durumda paketin hedefine bakmadan bütün paketlerin takibi olayıdır.Root yetkisi gerektirir.), otomatik kaydırma çubuğu hareketi, ve yakalanan paketlerin türlerine göre sayıları ve % oranlarını veren info penceresinin saklanması seçenekleri bulunur.

Printing: Yazdırmak için gereken ayarlar bulunmaktadır.Dosya çıktısı konumu, yazdırma komutu ve çıktı türü “düz metin ya da post script seçenekleri” bulunmaktadır.Varsayılan yazdırma komutu lpr dir.

Name Resolutions: Adres dönüşüm işlemlerini etkinleştirebileceğiniz alandır.

Protocols: ihtiyaca göre wireshark üzerinde paketlerin protokollere göre kullanım ayarlamalarını yapabileceğiniz bölümdür.

VIEW

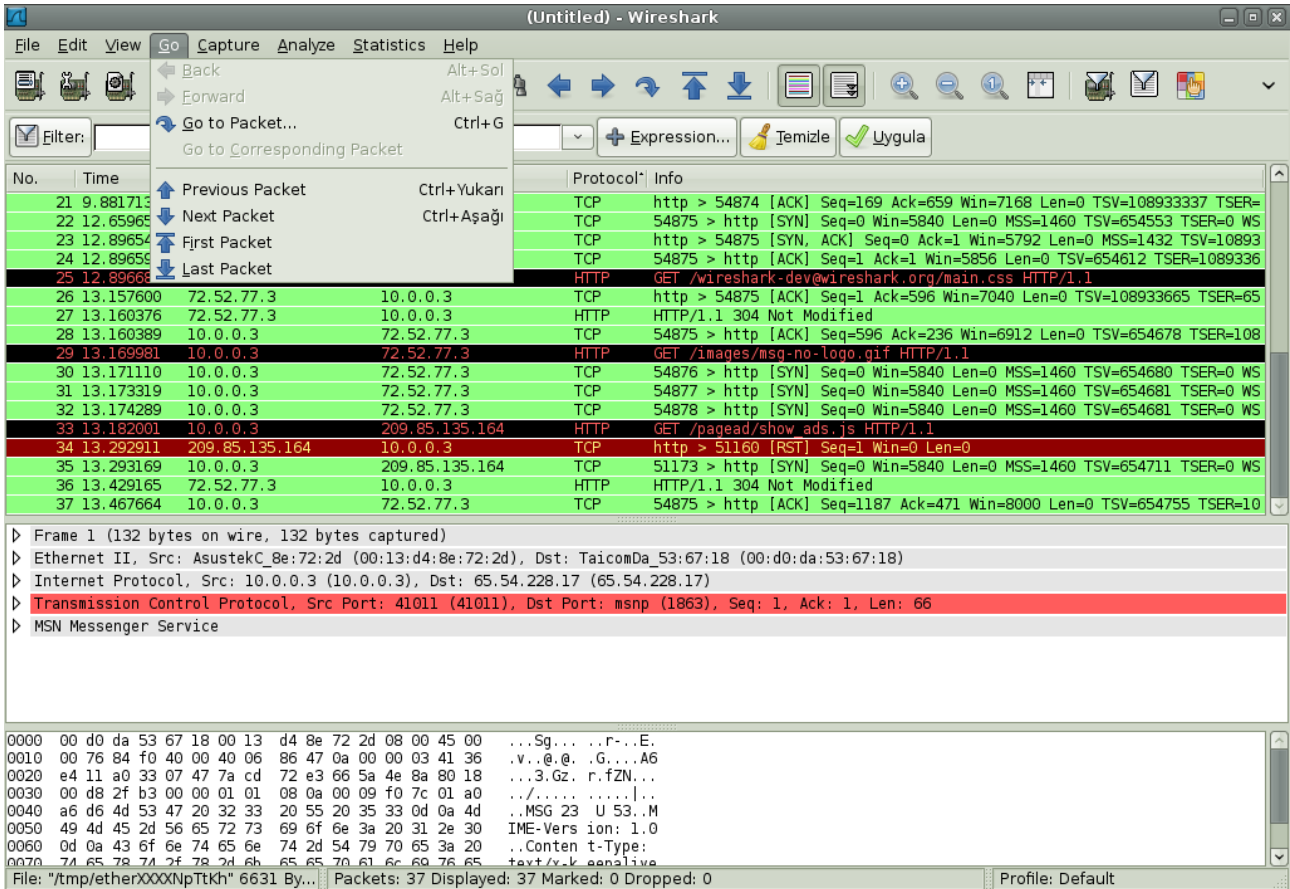


View menüsü wireshark programının görünümüyle ilgili ayarları yapabileceğiniz bölümdür.Araç çubuklarını gösterip saklayabilir, boyutlandırma yapabilirsiniz. Adres çözümleme seçeneği burada da karşımıza çıkıyor.Default olarak ağ adres çözümleme seçeneği seçili gelmemektedir.Eğer preference seçeneğinden ya da

capture options penceresinden enable network name resolution aktif hale getirilmediyse paket yakalama işleminden sonra buradan name resolution seçeneği altında ağ katmanını için adres çözümlemesi seçeneğini seçip reload (ctrl+R) ardından çözümlemenin yapıldığını görebilirsiniz. **Coloring Rules** seçeneği ile varolan renk ayarlarını değiştirebileceğiniz gibi belirlediğiniz filtre ifadesine istediğiniz rengi atayarak işinizi büyük ölçüde kolaylaştırabilirsiniz.



GO



Back(ctrl+so): Bir önceki baktığınız pakete atlar.

Forward (ctrl+sağ): Ziyaret edilen bir sonraki pakete zıplar.

Go To Packet (ctrl+G): Paket numarasına göre istenilen pakete zıplar.



Go To Corresponding Packet: Seçilen pakete karşılık gelen pakete zıplar .

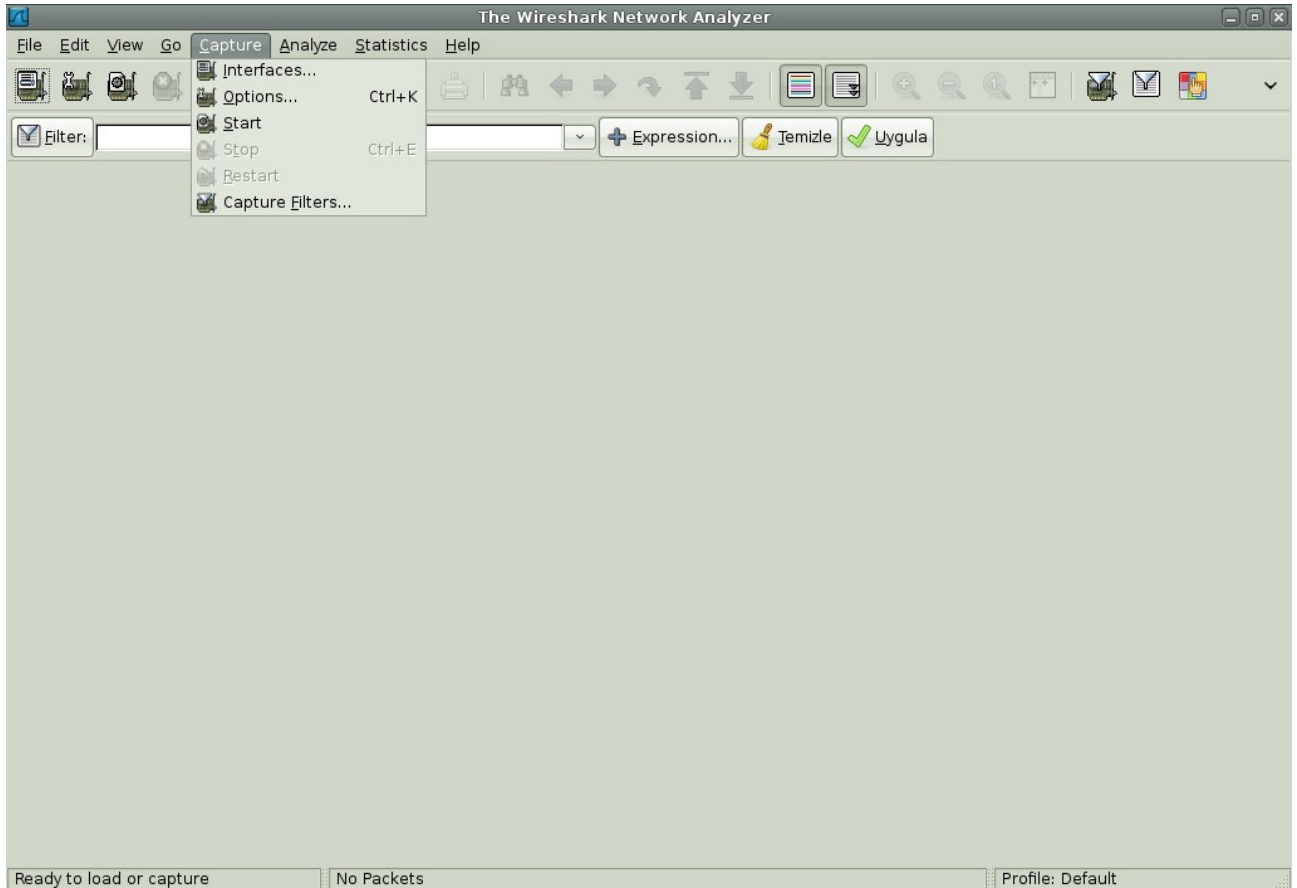
Previous Packet (ctrl+yukarı): Seçili paketten önceki pakete zıplar.

Next Packet(ctrl+aşağı): Seçili paketten sonraki pakete zıplar.

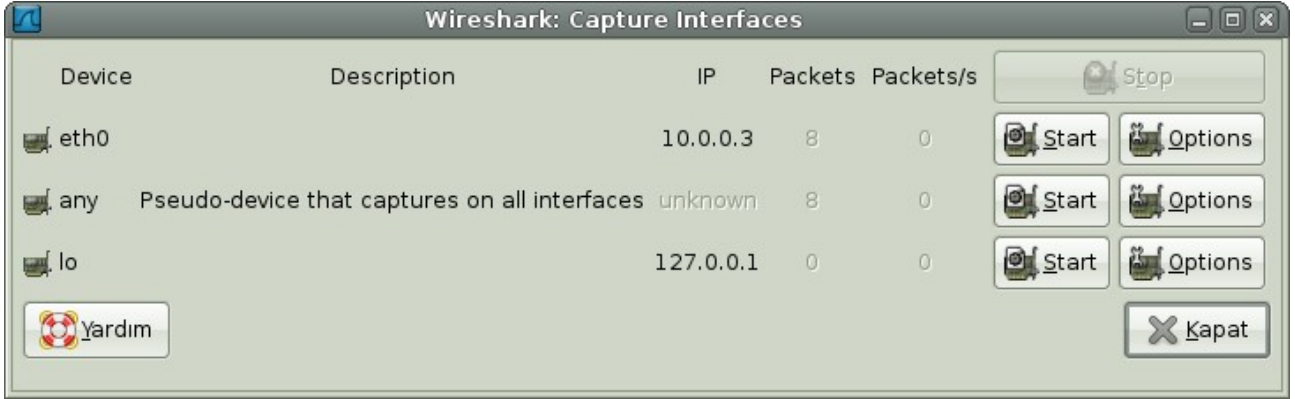
First Packet: Yakalanan ilk pakete zıplar.

Last Packet: Yakalanan son pakete zıplar.

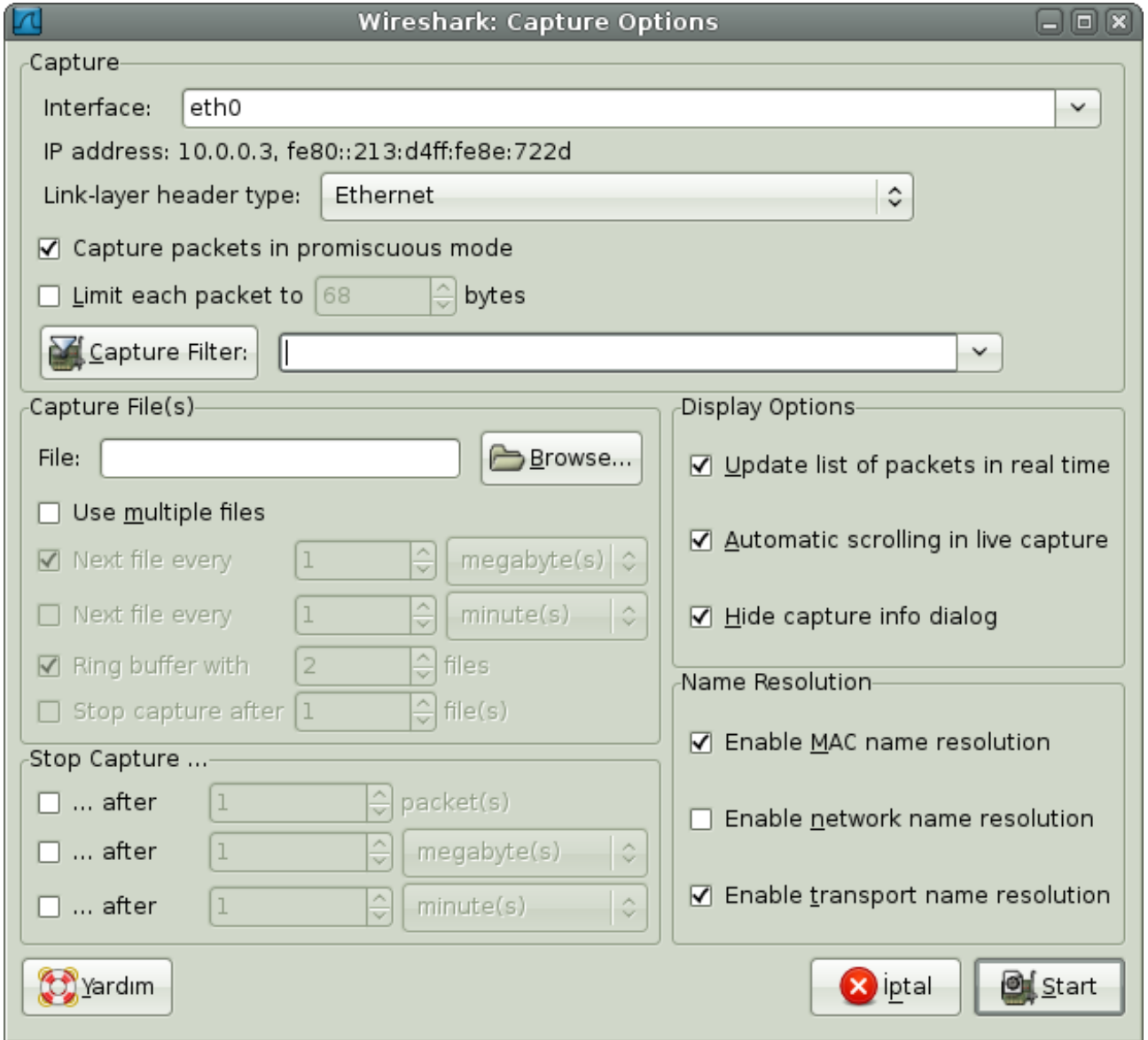
CAPTURE



Interfaces: Wiresharkın kullanacağı ağ arabirimi ve özellikleri ayarlanır.



Options: Uygulama sırasında kullanılacak ağ arabirimi seçimi adres çözümleme özellikleri, görünüm özellikleri uygulama durdurmak için ayarlanacak özellikler gibi bir çok ayarlanabilir bölüm içermektedir.



IP address:Seçilen ağ arabirimin sahip olduğu ip adresidir.

Limit each packet to n bytes : Paket yakalama işlemi sırasında uyulacak tampon sınırdır.Seçili olmadığı durumda default değeri 65535 bytes tır.

-Default değerde bırakmanız önerilir.

Capture packets in promiscuous mode : Hub kullanılan ağlarda yalnızca kullanılan makine ile ilgili paketleri değil gelen bütün paketleri hedeflerine bakmadan toplama özelliğidir.

Capture Filter: Paket yakalama sırasında filtreleme özelliği sunar.İstenmeyen paketlerin yakalanmasını engelleyerek hem analiz işlemini kolaylaştırır hemde programın çalışması sırasında daha az paket ile sistem kaynaklarını idareli kullanır.Filtreleme bölümünde ayrıntılı anlatılacaktır.

Capture File(s) Alanı:

----File: Yakalama dosyası olarak kullanılacak dosya ismi belirtmene yarar.Default olarak boştur.

----Use multiple files: Tek dosya kullanımı yerine wireshark otomatik olarak yeni bi dosyayla yer değiştirir.

----Next file every n megabyte(s): Belirtilen boyutta (kilobyte,megabyte,gigabyte) paket yakalandıktan sonra bir diğer dosyaya geçer.

----Next file every n minute(s): Belirtilen süre geçtikten sonra diğer dosyaya geçer.

----Ring buffer with n files: Belirtilen sayıda dosya aşıldığında en eski dosyayı siler.

----Stop capture after n file(s):Belirtilen sayıda dosya değiştikten sonra yakalama işlemini durdurur.

Stop Capture. Alanı:

... after n packet(s) : Belirtilen sayıda paket yakalandıktan sonra yakalama işlemini durdurur.

... after n megabytes(s): Belirtilen kb,mb,gb miktardan sonra yakalama işlemini durdurur.

... after n minute(s): Belirtilen süre sonunda (saniye, dakika, saat, gün) yakalama işlemini durdurur.

Display Options Alanı:

Update List Of Packets In Real Time: Yakalanan paketleri eşzamanlı olarak anında ekranda görmenize yarar.

Automatic Scrolling in Live Capture: Kaydırma çubuğu otomatik olarak son yakalanan pakete göre iner.

Hide Capture info Dialog: Yakalanan paketlerin protokollere göre sayı ve oranını veren bilgi penceresini saklar.

Name Resolution Alanı: Adres dönüşümü işlemlerini yapar.

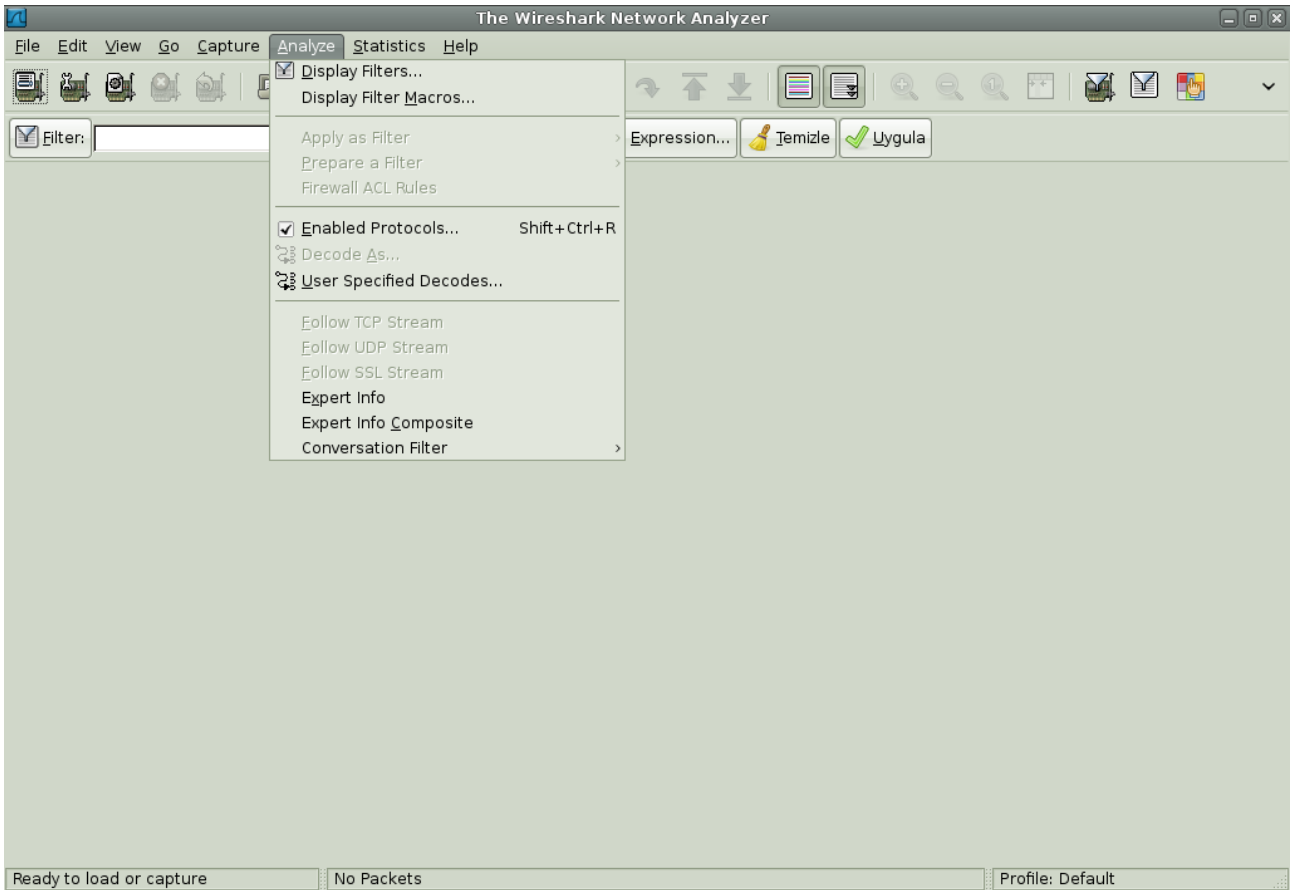
Start: Yakalama işlemini başlatır.

Stop: Yakalama işlemini durdurur.

Restart: Belirlenen seçeneklere göre yakalama işlemini tekrar başlatır.

Capture Filters : Paket yakalama işlemini belirtilen filtrelere göre yapar.

ANALYZE

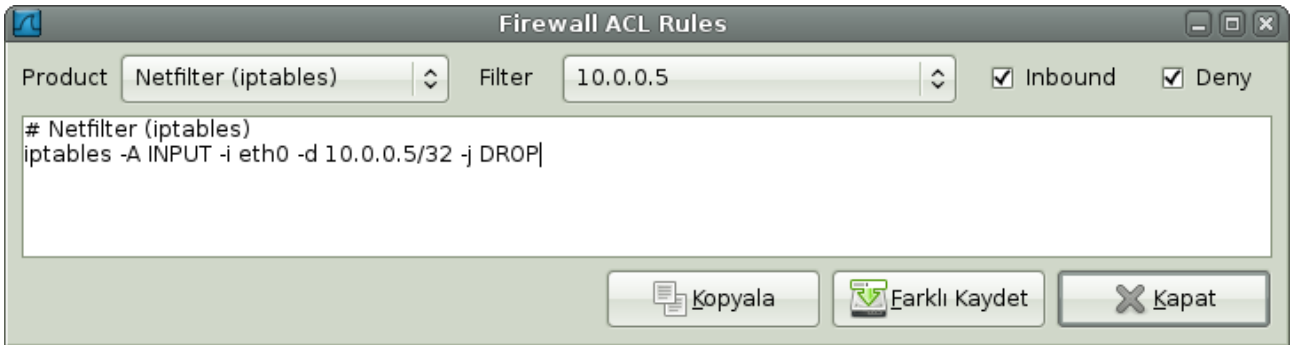


Display Filter: Yakalanan paketleri belirtilen ifadelerle göre sıralar.

Apply As Filter: Seçilen paketin kaynak ve hedef adresine göre filtreleme yapar.And (&&), or(| |), and not (&& !) ve or not (| | !) eklemeleriyle ifade güçlendirilir ve daha özelleşmiş arama yapılabilir.

Prepare a Filter: Filtre ifadesini değiştirir ama hemen uygulamaz.Üstteki filtre uygulaması koşulları bunun içinde geçerlidir.

Firewall ACL Rules :Cisco IOS, Linux Net- filter (iptables), OpenBSD pf ve Windows Firewall (via net- sh) için firewall kural ifadesi oluşturur.Yeni kullanıcılar için mükemmel ötesi bi özelliktir.



Decode As: Paketleri belirli protokollere göre decode eder

Enabled Protocols(shift+ctrl+R): Yakalama işlemi sırasında istenmeyen protokollerin kaldırılmasına imkan verir.Capture filter gibi düşünülebilir.

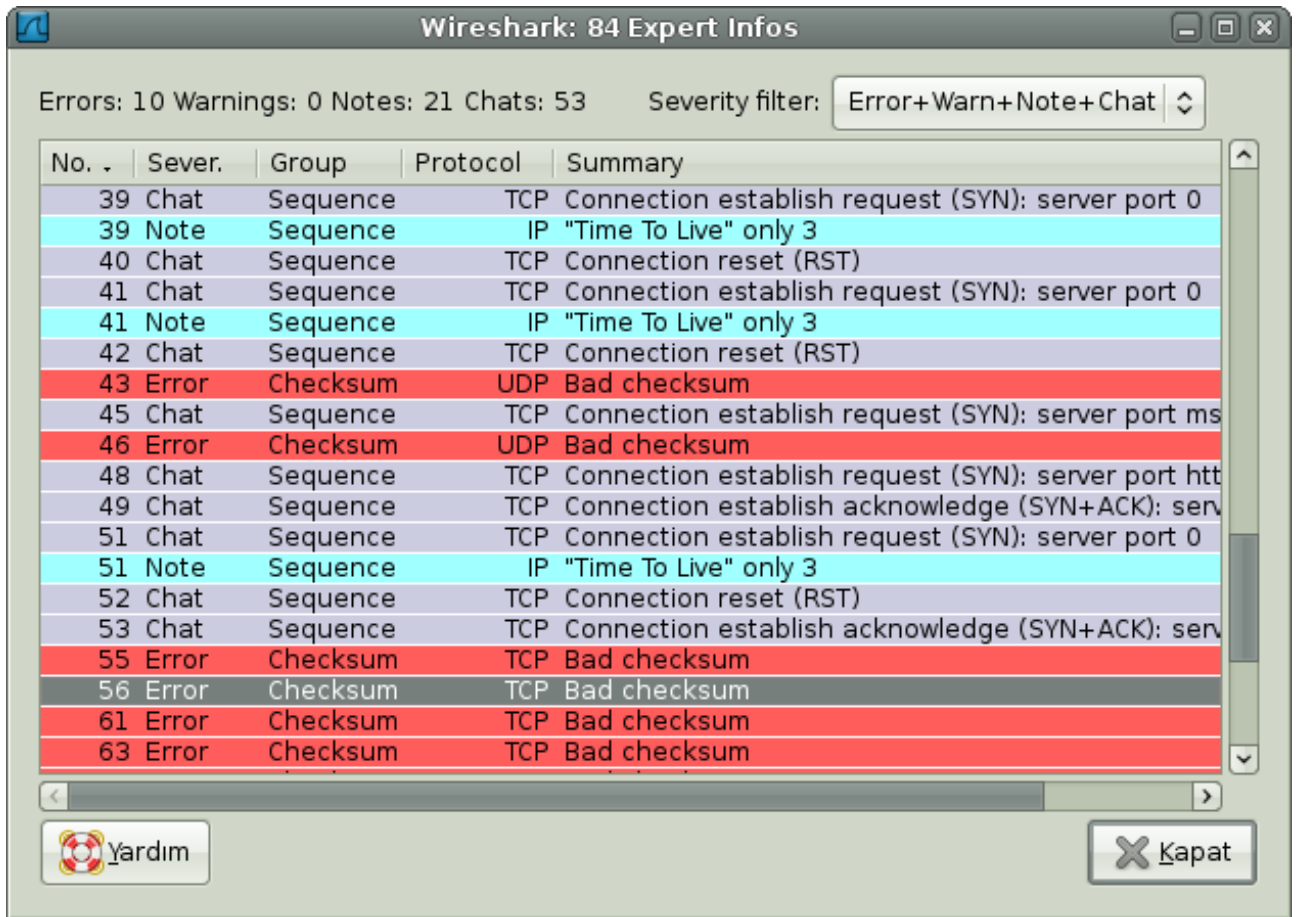
Decode As: Geçici olarak protokol çevrim işi yapar.

User Specified Decodes: Hali hazırda var olan çevrimleri görüntüler.

Follow TCP Stream: Seçilen paketle ilgili tcp bağlantılarının tüm tcp segmentlerini ayrı bir pencerede gösterir.

Follow SSL Stream: Follow TCP stream ile aynı özelliktedir fakat SSL stream için çalışır.

Expert Info: İletişimde meydana gelen olayların kayıt sistemidir.Yakalanan paketleri errors,notes,warnings,chats kriterlerine göre ayırır.



Expert Info Composite: Expert info ile aynı özelliktedir fakat ayrı gruplar halinde olduğundan daha hızlı analize imkan sağlar

STATISTIC

The image shows the Wireshark Statistics window. The left pane displays a list of 17 packets with their respective times and source addresses. The middle pane shows the details of the selected packet (No. 17), including the Hypertext Transfer Protocol section. The right pane shows the packet bytes, including the HTTP request and response.

No.	Time	Source
1	0.000000	10.0.0.4
2	0.545795	64.4.36.33
3	0.903741	10.0.0.4
4	0.989622	74.125.43.164
5	1.133643	10.0.0.4
6	1.287911	10.0.0.2
7	1.288021	10.0.0.4
8	1.394670	10.0.0.2
9	1.394772	10.0.0.4
10	1.413426	10.0.0.2
11	1.413588	10.0.0.4
12	4.411743	10.0.0.4
13	4.497935	85.13.139.89
14	4.497983	10.0.0.4
15	4.498070	10.0.0.4
16	4.558245	85.13.139.89
17	4.558276	10.0.0.4

Packet 17 details:

- Window size: 7424 (scaled)
- Checksum: 0xe0a7 [correct]
- Options: (12 bytes)
- [SEQ/ACK analysis]
- [This is an ACK to the segment]
- [The RTT to ACK the segment was ...]
- Hypertext Transfer Protocol

Packet bytes:

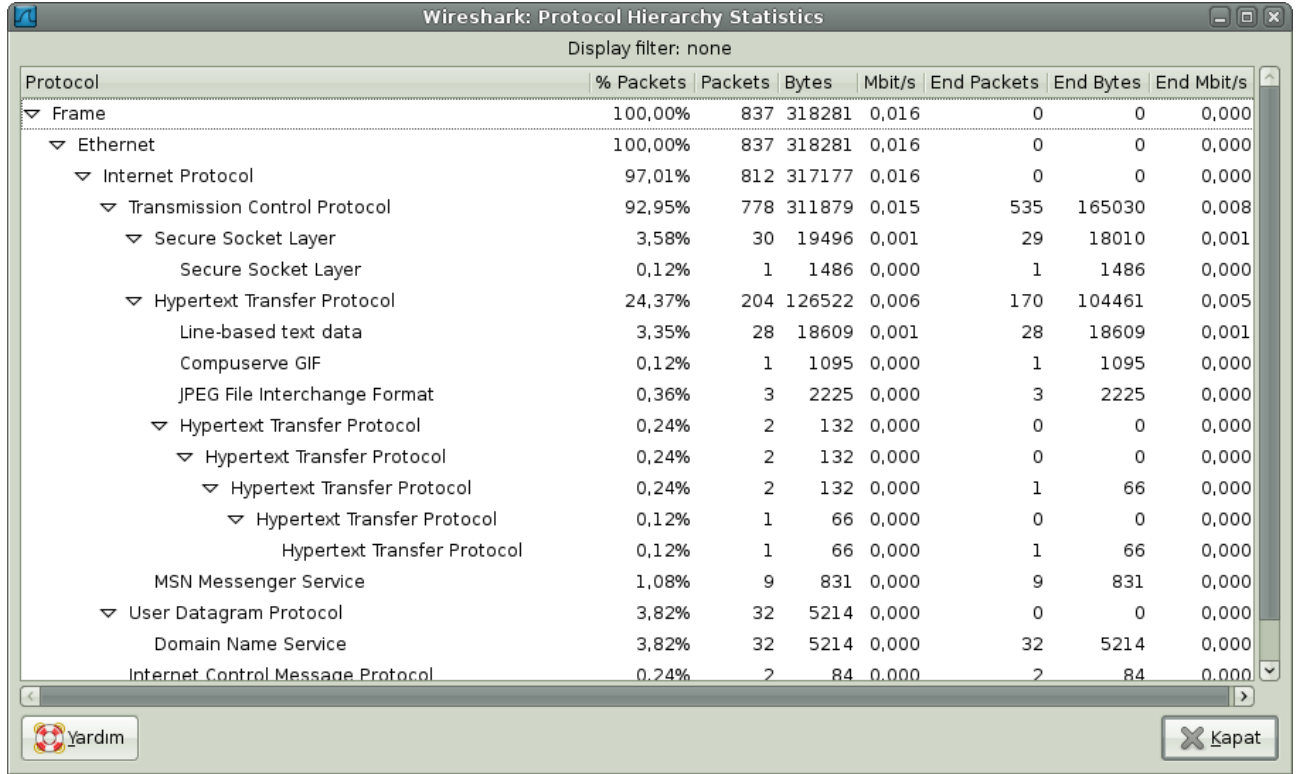
```
...E.  
...Y..  
...RG..  
...form  
...d,thi  
...cumen  
...d,for
```

Summary: Açık olan yakalama dosyasında dosya formatı, paket sayısı, boyut, ilk ve son paket yakalama zamanları , filtre ve yakalama arabirimine ilişkin verileri içerir.

The image shows the Wireshark Summary window. It provides a summary of the capture file, including file name, length, format, and packet size limit. It also shows the time range of the capture, the interface used, and the number of packets captured. The traffic statistics section provides a detailed overview of the captured data, including the number of packets, bytes, and the average packet size.

Traffic	Captured	Displayed	Marked
Packets	837	837	0
Between first and last packet	161,830 sec		
Avg. packets/sec	5,172		
Avg. packet size	380,264 bytes		
Bytes	318281		
Avg. bytes/sec	1966,763		
Avg. MBit/sec	0,016		

Protocol Hierarchy : Yakalanan paketlerin ağaç şeklinde katman ve protokol hiyerarşisini gösterir. Her sıra bir protokole ait istatistiksel değerleri tutar. Seçilen sıra filtre ifadesi olarak kullanılabilir.



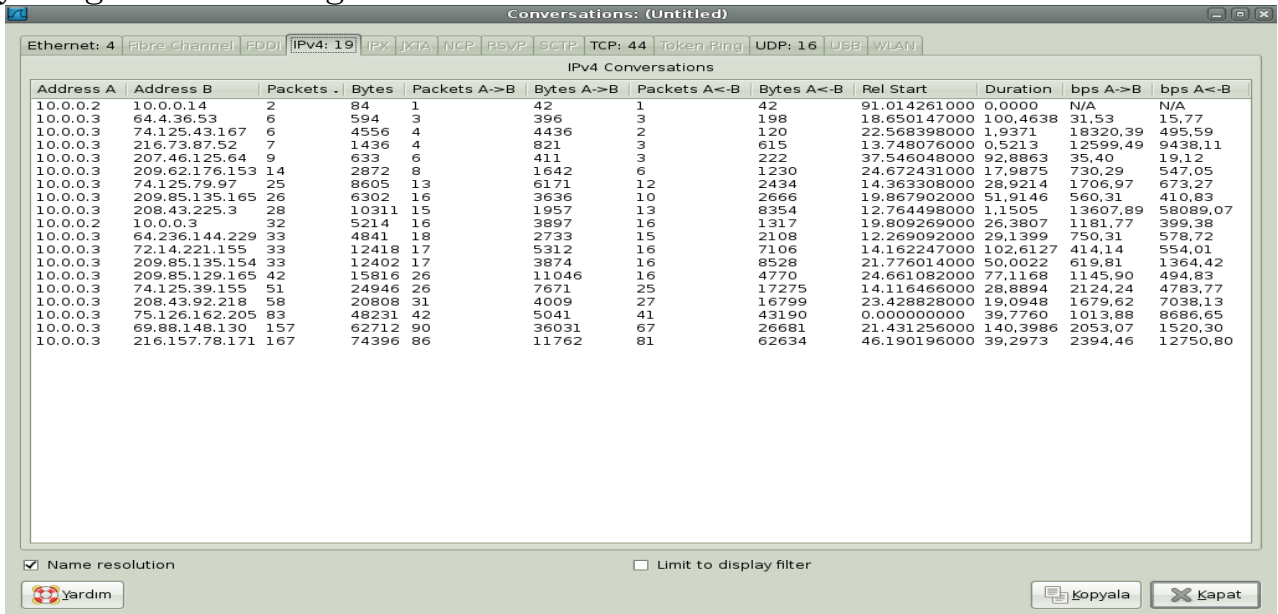
Wireshark: Protocol Hierarchy Statistics

Display filter: none

Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00%	837	318281	0,016	0	0	0,000
Ethernet	100,00%	837	318281	0,016	0	0	0,000
Internet Protocol	97,01%	812	317177	0,016	0	0	0,000
Transmission Control Protocol	92,95%	778	311879	0,015	535	165030	0,008
Secure Socket Layer	3,58%	30	19496	0,001	29	18010	0,001
Secure Socket Layer	0,12%	1	1486	0,000	1	1486	0,000
Hypertext Transfer Protocol	24,37%	204	126522	0,006	170	104461	0,005
Line-based text data	3,35%	28	18609	0,001	28	18609	0,001
Compuserve GIF	0,12%	1	1095	0,000	1	1095	0,000
JPEG File Interchange Format	0,36%	3	2225	0,000	3	2225	0,000
Hypertext Transfer Protocol	0,24%	2	132	0,000	0	0	0,000
Hypertext Transfer Protocol	0,24%	2	132	0,000	0	0	0,000
Hypertext Transfer Protocol	0,12%	1	66	0,000	0	0	0,000
Hypertext Transfer Protocol	0,12%	1	66	0,000	1	66	0,000
Hypertext Transfer Protocol	0,12%	1	66	0,000	1	66	0,000
MSN Messenger Service	1,08%	9	831	0,000	9	831	0,000
User Datagram Protocol	3,82%	32	5214	0,000	0	0	0,000
Domain Name Service	3,82%	32	5214	0,000	32	5214	0,000
Internet Control Message Protocol	0,24%	2	84	0,000	2	84	0,000

Yardım Kapat

Conversations: Kaynak ve hedef noktaları arasındaki trafiğin istatistik bilgisini verir. Noktalar arasındaki toplam-gelen-giden paket ve byte miktarı portlara göre listelenir. Conversations penceresi endpoint penceresiyle benzerdir. Listedeki her bir sıra bir diyalogun istatistiksel değerlerini verir. Adres çözümleme özelliği Conversations penceresi içinde, programın başlangıcında “capture options” bölümünden , preferences altında name resolutions bölümünden ya da view menüsü altında name resolutions bölümünden seçildikten sonra kullanılabilir. Limit to display filter özelliği ise herhangi bir filtreleme yönergesi tanımlandığı durumda kullanılabilir.



Conversations: (Untitled)

Ethernet: 4 Fibra Channel FDDI IPv4: 19 IPX JXTA NCP RSVP SCTP TCP: 44 Token Ring UDP: 16 USB WLAN

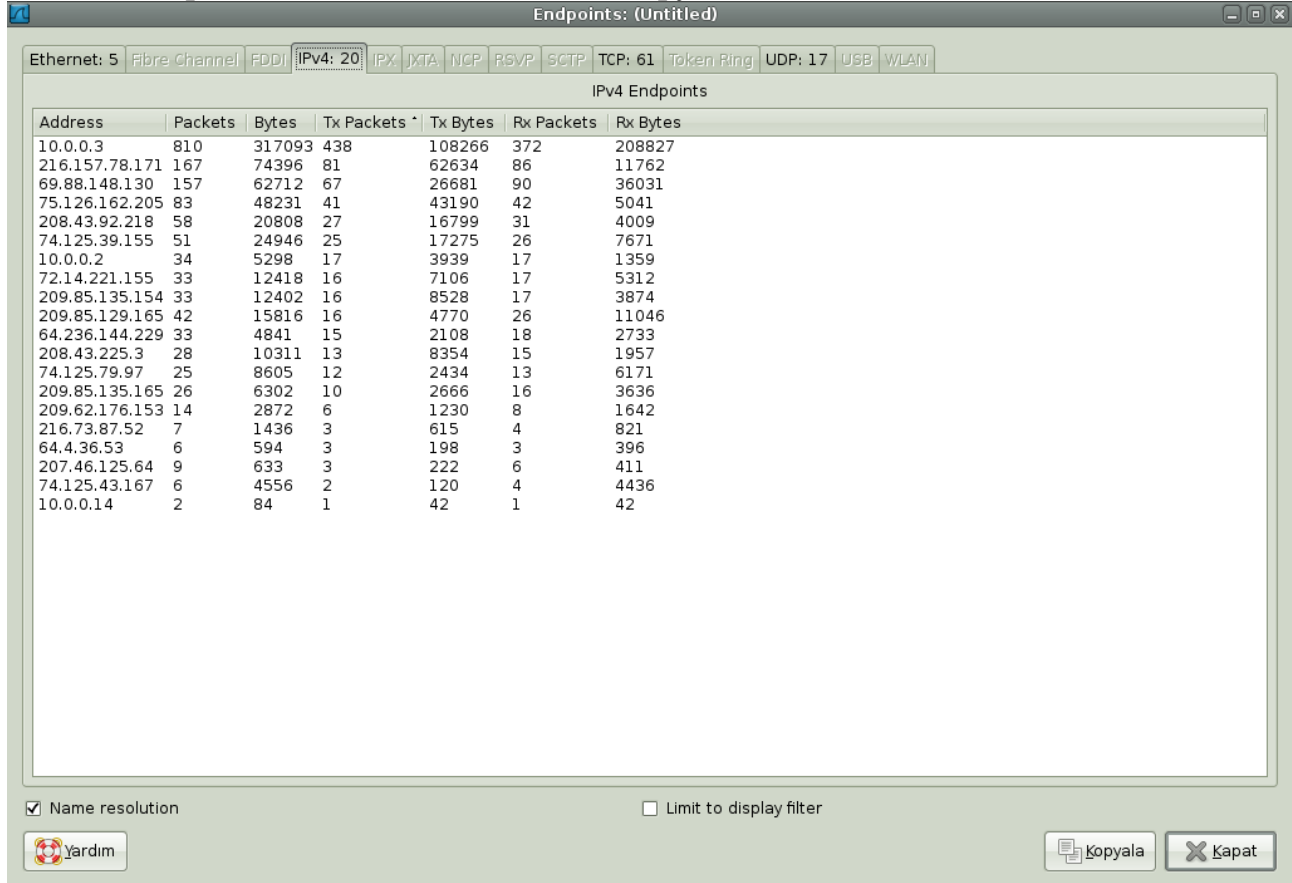
IPv4 Conversations

Address A	Address B	Packets .	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
10.0.0.2	10.0.0.14	2	84	1	42	1	42	91.014261000	0,0000	N/A	N/A
10.0.0.3	64.4.36.53	6	594	3	396	3	198	18.650147000	100,4638	31,53	15,77
10.0.0.3	74.125.43.167	6	4556	4	4436	2	120	22.568398000	1,9371	18320,39	495,59
10.0.0.3	216.73.87.52	7	1436	4	821	3	615	13.748076000	0,5213	12599,49	9438,11
10.0.0.3	207.46.125.64	9	633	6	411	3	222	37.546048000	92,8863	35,40	19,12
10.0.0.3	209.62.176.153	14	2872	8	1642	6	1230	24.672431000	17,9875	730,29	547,05
10.0.0.3	74.125.79.97	25	8605	13	6171	12	2434	14.363308000	28,9214	1706,97	673,27
10.0.0.3	209.85.135.165	26	6302	16	3636	10	2666	19.867902000	51,9146	560,31	410,83
10.0.0.3	208.43.225.3	28	10311	15	1957	13	8354	12.764498000	1,1505	13607,89	58089,07
10.0.0.2	10.0.0.3	32	5214	16	3897	16	1317	19.809269000	26,3807	1181,77	399,38
10.0.0.3	64.236.144.229	33	4841	18	2733	15	2108	12.269092000	29,1399	750,31	578,72
10.0.0.3	72.14.221.155	33	12418	17	5312	16	7106	14.162247000	102,6127	414,14	554,01
10.0.0.3	209.85.135.154	33	12402	17	3874	16	8528	21.776014000	50,0022	619,81	1364,42
10.0.0.3	209.85.129.165	42	15816	26	11046	16	4770	24.661082000	77,1168	1145,90	494,83
10.0.0.3	74.125.39.155	51	24946	26	7671	25	17275	14.116466000	28,8894	2124,24	4783,77
10.0.0.3	208.43.92.218	58	20808	31	4009	27	16799	23.428828000	19,0948	1679,62	7038,13
10.0.0.3	75.126.162.205	83	48231	42	5041	41	43190	0.000000000	39,7760	1013,88	6686,65
10.0.0.3	69.88.148.130	157	62712	90	36031	67	26681	21.431256000	140,3986	2053,07	1520,30
10.0.0.3	216.157.78.171	167	74396	86	11762	81	62634	46.190196000	39,2973	2394,46	12750,80

☒ Name resolution ☐ Limit to display filter

Yardım Kopyala Kapat

Endpoints: Hedef ve kaynak adresi ayrımı yapmadan her son nokta için istatistik bilgisini verir. Desteklenen her protokol için ayrı bir sekme mevcuttur. Her sekmede yakalanan son nokta sayıları belirtilmektedir. Örneğin ethernet:5 Hali hazırda 5 tane ethernet son noktasının yakalandığını söylemektedir. Eğer protokolle ilgili yakalanmış son nokta yoksa ilgili sekme silik şekilde görünmektedir. Her sıra bir son nokta için istatistiksel değerleri göstermektedir. Name resolution ve limit to display filter özellikleri conversation bölümünde anlatıldığı gibi kullanabilmektedir. kopyala butonu değerleri CSV (Comma Separated Values) formatında kopyalamaktadır.



The screenshot shows a window titled "Endpoints: (Untitled)" with a tabbed interface. The "IPv4: 20" tab is selected. Below the tabs is a table titled "IPv4 Endpoints" with the following columns: Address, Packets, Bytes, Tx Packets, Tx Bytes, Rx Packets, and Rx Bytes. The table contains 20 rows of data. At the bottom of the window, there are checkboxes for "Name resolution" (checked) and "Limit to display filter" (unchecked). There are also buttons for "Yardım" (Help), "Kopyala" (Copy), and "Kapat" (Close).

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
10.0.0.3	810	317093	438	108266	372	208827
216.157.78.171	167	74396	81	62634	86	11762
69.88.148.130	157	62712	67	26681	90	36031
75.126.162.205	83	48231	41	43190	42	5041
208.43.92.218	58	20808	27	16799	31	4009
74.125.39.155	51	24946	25	17275	26	7671
10.0.0.2	34	5298	17	3939	17	1359
72.14.221.155	33	12418	16	7106	17	5312
209.85.135.154	33	12402	16	8528	17	3874
209.85.129.165	42	15816	16	4770	26	11046
64.236.144.229	33	4841	15	2108	18	2733
208.43.225.3	28	10311	13	8354	15	1957
74.125.79.97	25	8605	12	2434	13	6171
209.85.135.165	26	6302	10	2666	16	3636
209.62.176.153	14	2872	6	1230	8	1642
216.73.87.52	7	1436	3	615	4	821
64.4.36.53	6	594	3	198	3	396
207.46.125.64	9	633	3	222	6	411
74.125.43.167	6	4556	2	120	4	4436
10.0.0.14	2	84	1	42	1	42

IO Graphs: Belirtilen özelliklerde paketlerin zamana göre akış grafiğini verir. Ağda durum kontrolü için oldukça faydalı bir özelliktir. Bu özellik normal paket akış diyagramında ağda meydana gelecek herhangi bir anormallik hemen farkedilebilir. X eksenini için diyagramdaki işaret süreleri ayarı ve ayarlanan işaret çizelgesinde gösterilecek ayrıntı miktarı değiştirilebilir, saate göre çizim yapılabilir. Y eksenini için ise paket sayısı, byte, bit değerlerine göre çizim yaptırılabilir. Bunların yanında okunabilirliği kolaylaştırmak için farklı protokollere göre farklı renklendirme ve farklı arama kriterleri içinse filtreleme özelliği kullanılabilir.

Graphs

- **Graph 1-5:** Grafik içerisinde belirtilen özelliklerde 5 değişken belirtilebilir.
- **Color:** Belirtilen özelliklerde çizilen değişkenin rengini gösterir. Değiştirilemez.
- **Filter:** İhtiyaca göre istenilen özellikteki paketlerin grafiğinin

eldeste kullanılır

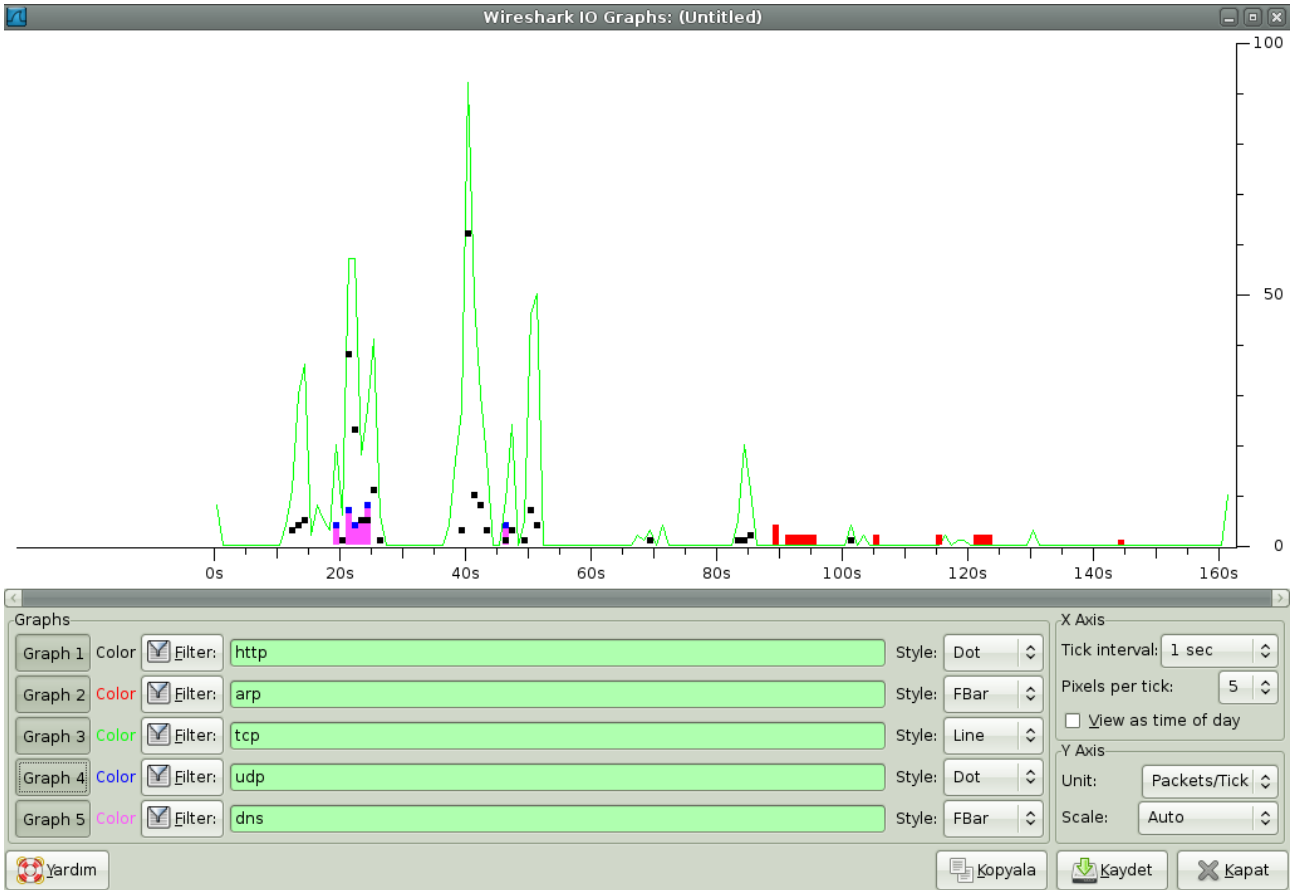
- **Style:** Grafik stilini seçmemize yarar. (Line/Impulse/FBar/Dot)

- **X Axis**

- **Tick interval:** X eksenini zaman aralığıdır. (10/1 dakika yada 10/1/0.1/0.01/0.001 saniye olarak)
- **Pixels per tick:** Her bir işaret arasında 10/5/2/1 pixel kullanabilir
- **View as time of day:** X ekseninde zaman belirteci olarak yerel saati kullanır.

- **Y Axis**

- **Unit:** Y eksenini değışkeni olarak Paket/işaret, Byte/işaret, Bit/işaret ve gelişmiş.. seçenekleri mevcuttur.
- **Scale:** Ölçek değeri (Logaritmik,otomatik,10,20,50,100,200,500,...)



Conversation List ve **Endpoint List** menüleri Conversations ve Endpoints pencerelerindeki sekmelerin ayrılmış halidir.

Service Response Time: İstek ve cevap arasındaki zamanı gösterir. Service response time istatistikleri aşağıdaki protokoller için kullanılmaktadır.

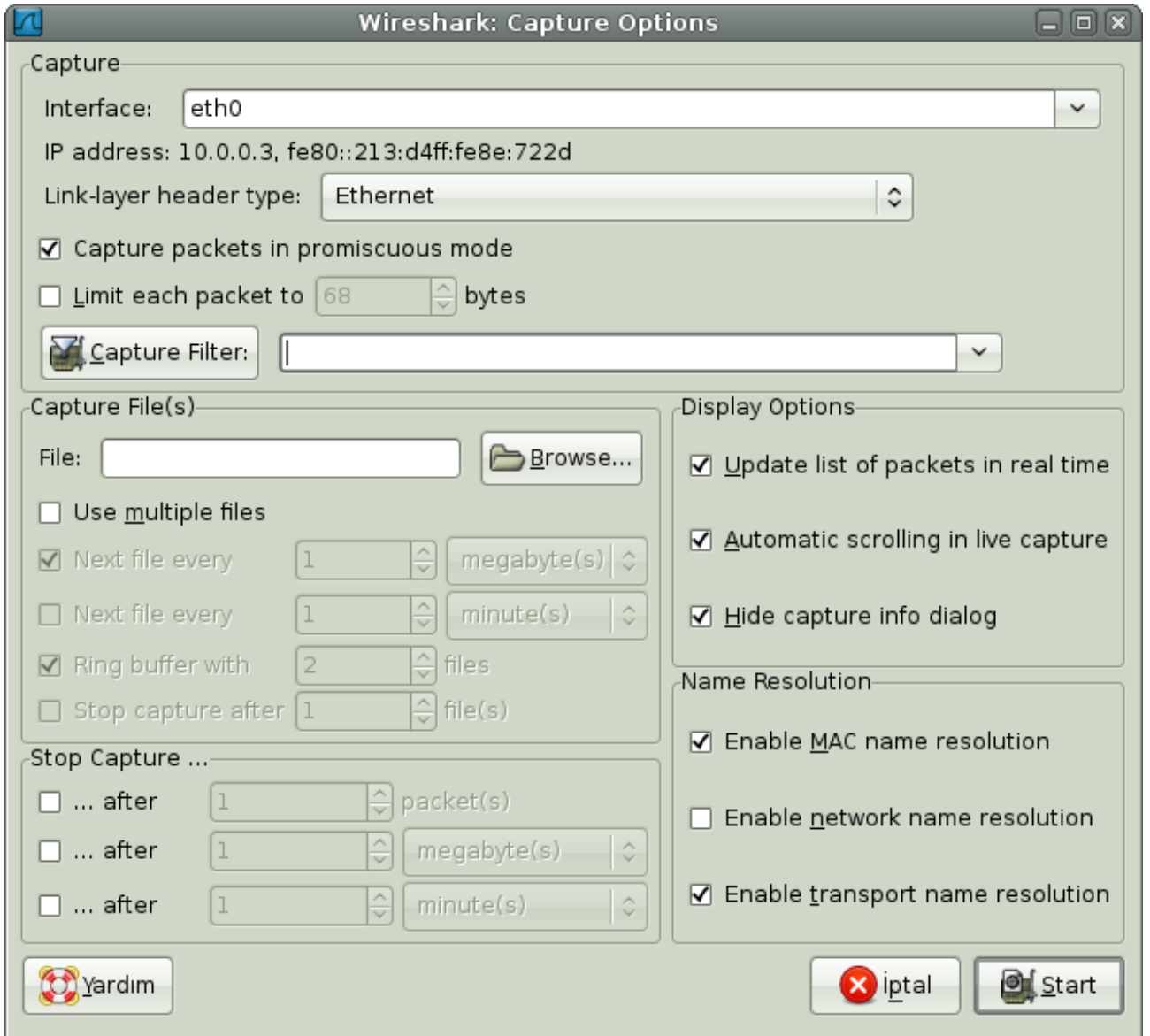
- **DCE-RPC**

- **Fibre Channel**
- **H.225 RAS**
- **LDAP**
- **MGCP**
- **ONC-RPC**
- **SMB**
- **ANSI**
- **GSM**
- **H.225**

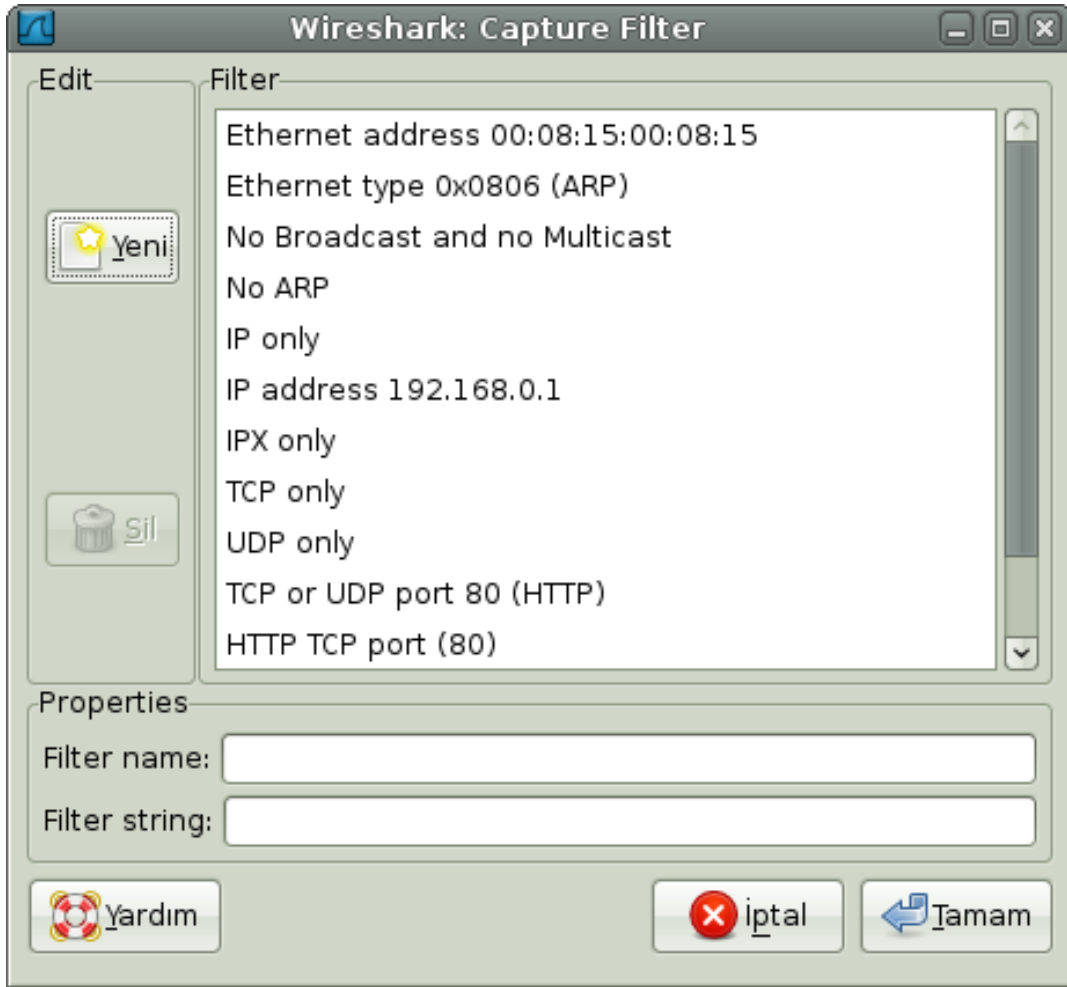
Wlan Traffic Statics: Yakalanan kablosuz ağ trafiğinin istatistik bilgisinin sunar.

FILTRELER

Wiresharkın en önemli özelliği filtrelerde sıra.Filtre özelliği, wiresharkta dinleme sırasında veya dinlemenin ardından paketler arasında istenilen özellikteki paketleri görüntülemekte kullanılabilir.



Capture options penceresinden belirtilen capture filter, paket **yakalama sırasında** wiresharkın uyacağı koşulları belirtir.



Görüldüğü gibi capture filter penceresinde Wiresharkın paket yakalama sırasında uyacağı kurallar için bir liste sunulmuştur.

Burada

Ethernet address 00:08:15:00:08:15 : Ethernet II altında kaynak veya hedef adreslerinde belirtilen mac adresine ait paketleri yakalar.

Ethernet type 0x0806 (ARP), 0x0800 (IP), 0x8035(RARP), 0x6003 (DECNET), 0x6004 (DEC LAT), 0x6002 (MOP RC), 0x6001 (MOP DL)

not broadcast and not multicast: Broadcasting ve multicasting paketlerini yakalamaz.

not arp: Arp pakelerini yakalamaz.

IP address 192.168.0.1 : Belirtilen ip adresini hedef yada kaynak adres kısımlarında barındıran paketler yakalanır.

IPX only : İlgili protokole ilişkin paketleri yakalar.

TCP only : İlgili protokole ilişkin paketleri yakalar.

UDP only : İlgili protokole ilişkin paketleri yakalar.

TCP or UDP port 80 (http) : Port 80 için tcp ve udp paketlerini yakalar.

HTTP TCP port (80) : Port 80 için http ve tcp paketlerini yakalar.

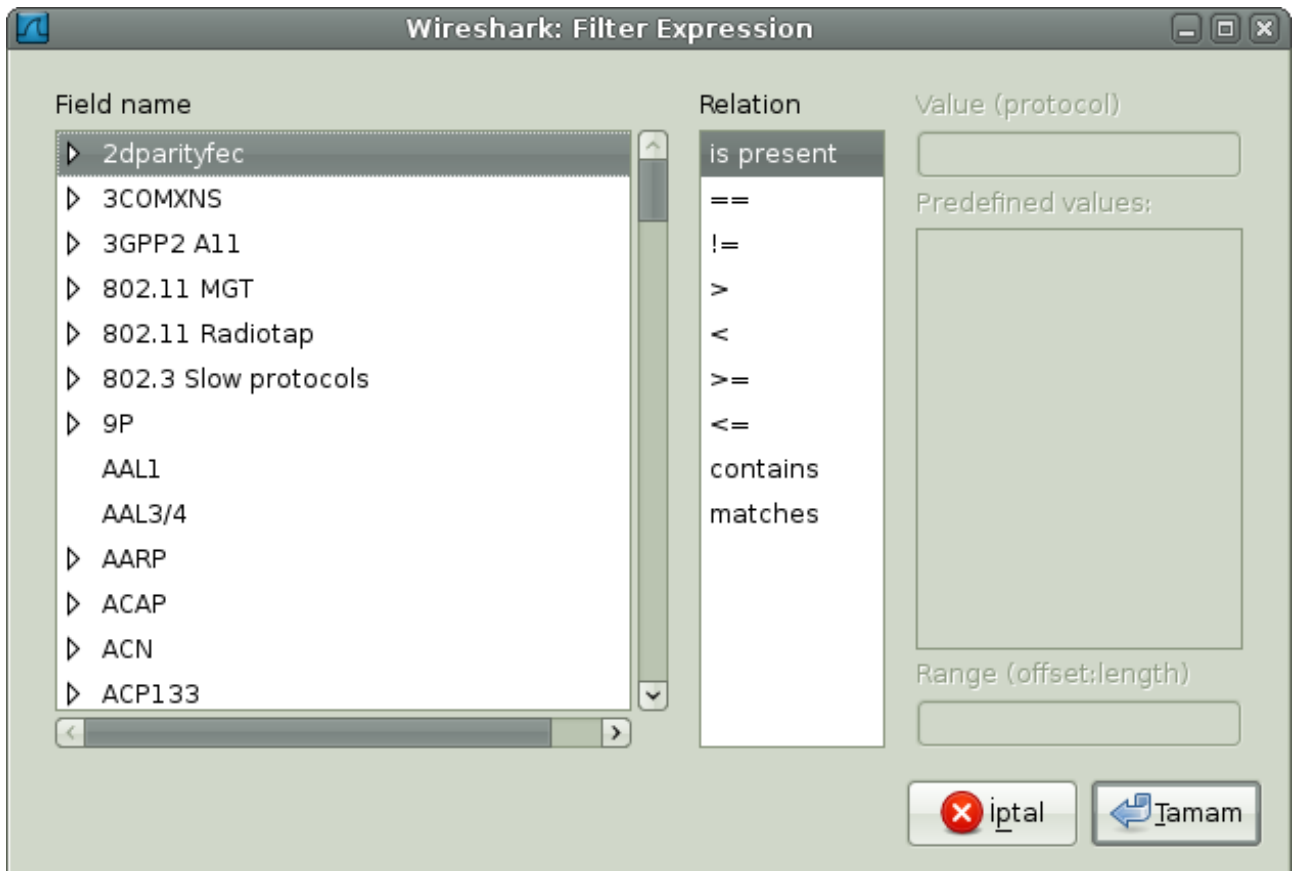
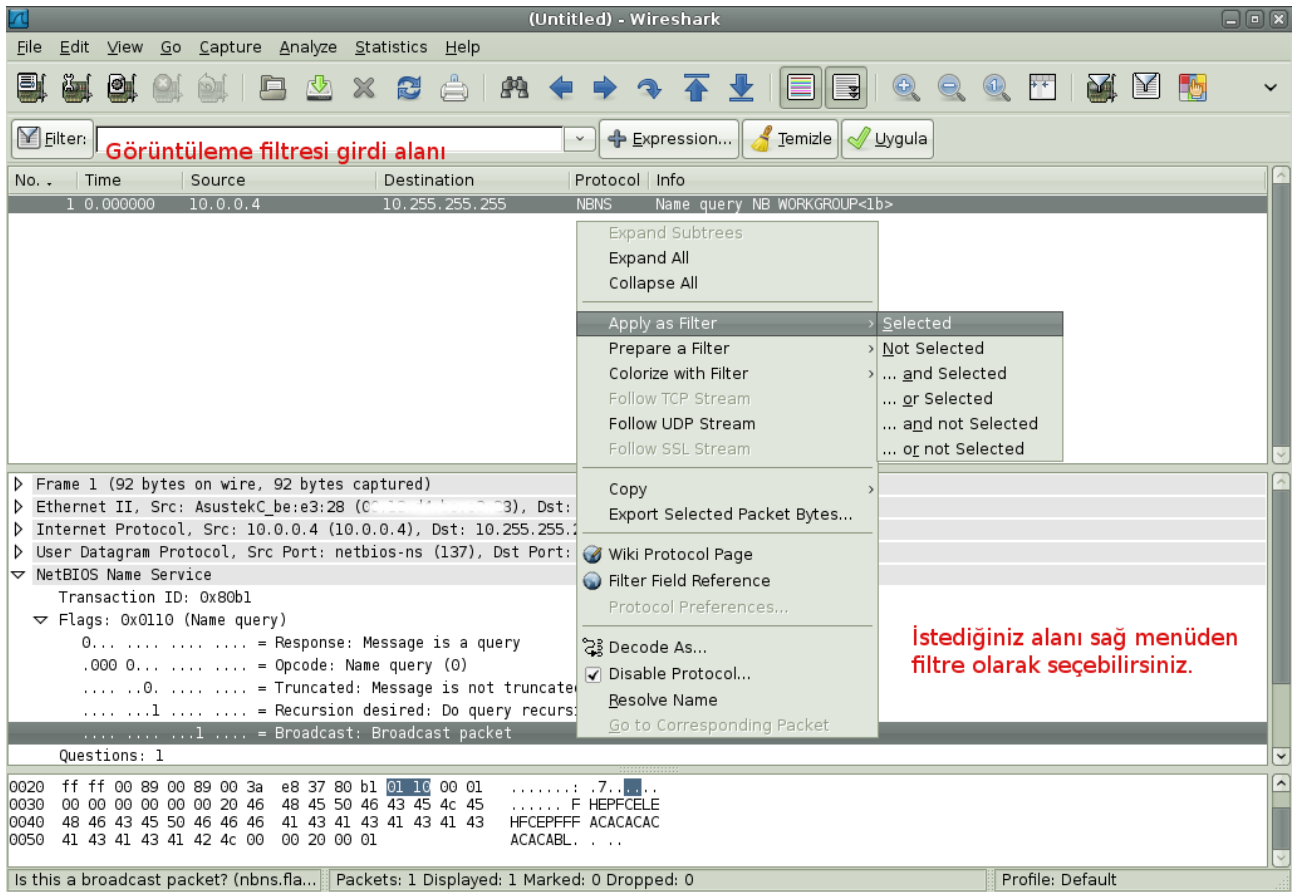
No ARP and no DNS : DNS ve ARP paketleri harici paketleri yakalar

Non-HTTP and non-SMTP to/from www.wireshark.org : Belirtilen adres için http ve smtp harici paketleri yakalar.

Örnek olarak capture filter **host 74.125.43.103** olarak belirttiğimizde wireshark ip paket başlığı altında kaynak ve hedef adreslerine bakarak sadece 74.125.43.103 barındıran paketleri yakalamaktadır.Aynı şekilde **ip src host 10.0.0.3** ifadesiyle wireshark kaynak adresi 10.0.0.3 olan paketleri dinlemektedir.

Display filter: Wiresharkın yakaladığı paketler içinden istenilen özellikteki paketlerin gösteriminde kullanılır.Mantıksal ve karşılaştırma operatörleri burada da kullanılmaktadır Araç çubuğunun altında filtre bölümü bulunmaktadır.Display filter in capture filter a göre en büyük üstünlüğü filtreleme ifadelerinin expression penceresi altından seçilebilmesidir.Bu özellik yeni kullanıcılar için çok faydalıdır.

Wiresharkın filtrelemedeki en büyük artışı imkan verdiği alanlarda istenilen özelliklerin sağ menü ile filtre olarak kullanılabilmesidir.Örneğin yakalanan paketler içersinden belirli bir kaynak adresine sahip paketleri görüntülemek için paket listesi bölümünden ilgili kaynak adresine sağ tıklayıp menüden **apply as filter** ile istediğimiz filtrelemeyi gerçekleştirmiş oluyoruz. Bunun gibi paket ayrıntıları bölümünden de istediğimiz özellikleri filtre ifadesi olarak kullanabiliyoruz.



Wiresharkın filtre sistemine alıştığınızda ve hangi etiketleri filtrelerinizde

kullanacağınıza karar verdiğinizde filtre dizgisi yazmak çok çabuk ve basit olmaktadır.Fakat wiresharkta yeniyseniz Ya da yabancı olduğunuz protokollerle çalışıyorsanız ne yazacağınızı kararlaştırmak çok karmaşık bir hal almaktadır.Bu durumda da **filter expression dialog** penceresi bize yardıma koşar.

not!

Filtre ifadesi diyalog penceresi wireshark görüntüleme filtre dizgileri yazımını öğrenmek için mükemmel bir yoldur.

Wireshark basit ama güçlü görüntüleme filtre dili ile karmaşık filtre ifadeleri oluşturmanıza imkan sağlar.Paketlerdeki değerleri karşılaştırabileceğiniz gibi ifadeleri birleştirerek daha karmaşık ifadeler oluşturabilirsiniz.

Wireshark yakalama ve gösterme filtrelerinin bir diğer güzel özelliği ise ifadelerde mantıksal ve karşılaştırma operatörlerinin kullanılabilmesidir.Bir ifadenin yanında başka bir ifade kullanmak için ve (&&), veya (!) bunların deęilleri ! koşulları birlikte kullanılabilir.

Karşılaştırma Deęerleri:

Farklı karşılaştırma operatörlerini karşılaştırma deęerleriyle birlikte kullanarak görüntüleme filtreleri oluşturabilirsiniz.

English	C-like	Örnek ve Açıklama
eq	==	Eşittir ip.src==10.0.0.5
ne	!=	Eşit değildir ip.src!=10.0.0.5
gt	>	Büyüktür frame.len > 10
lt	<	Küçüktür frame.len < 128
ge	>=	Büyük yada Eşittir

		frame.len ge 0x100
le	<=	Küçük yada Eşittir frame.len <= 0x20

Operatörleri ingilizce kısaltmalarıyla yada c-like terimleri şeklinde kullanabilirsiniz.

Görüntüleme filtresi mantıksal operatörleri

English	C-like	Açıklama ve Örnek
And	&&	Mantıksal ve ip.src==10.0.0.5 and tcp.flags.fin
Or		Mantıksal veya ip.scr==10.0.0.5 or ip.src==192.1.1.1
Xor	^^	Mantıksal xor tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29
Not	!	Mantıksal değildir not llc
[...]		Alt dizi operatörü Wireshark sekansın alt sekanslarını ayrıntılı yollarla seçmeye imkan sağlar. Etiketten sonra virgülle ayrılmış belirteç aralığının listesini içeren köşeli parantez yerleştirebilirsiniz.

eth.src[0:3] == 00:00:83

Üstteki örnekte tek bir aralık belirtmek için n:m formatı kullanılmıştır.Burada n ofset başlangıcı m ise belirlenen aralığın uzunluğudur.

eth.src[1-2] == 00:83

Üstteki örnekte tek bir aralık belirtmek için n-m formatı kullanılmıştır.Buarada n ofset başlangıcı m ise ofset bitişidir.

eth.src[:4] ==00:00:83:00

Üstteki örnekte sekans başlangıcından m ofsetine kadar herşeyi alan :m formatı kullanılmıştır

eth.src[4:] == 20:20

Üstteki örnekte n ofsetinden sekans sonuna kadar herşeyi alan n: formatı kullanılmıştır.

eth.src[2] == 83

Üstteki örnekte tek aralık belirtmek için n formatı kullanılmıştır.n:1 ile eşdeğerdir.

eth.src[0:3,1-2,:4,4:,2] ==

00:00:83:00:83:00:00:83:00:20:20:83

		Wireshark üstte gösterildiği gibi virgülle ayrılmış tek aralıkları birlikte dizi haline getirmenize olanak sağlar.
--	--	--

Yaygın Hata:

Uyarı!

!= operatörünün eth.addr, ip.addr, tcp.port, udp.port gibi ifadelerle birlikte kullanılması beklenildiği gibi çalışmayacaktır.

Çoğunlukla 1.2.3.4 ip adresini içeren bütün paketleri gösteren `ip.addr == 1.2.3.4` şeklinde filtre dizgisi kullanılır.

Benzer olarak 1.2.3.4 ip adresini içermeyen paketleri görüntülemek için **`ip.addr != 1.2.3.4`** kullanılır. Maalesef bu beklenildiği gibi olmaz.

Onun yerine o ifade kaynak yada hedef ip adresi 1.2.3.4 olan paketler için doğru olacaktır. Bunun sebebi **`ip.addr != 1.2.3.4`** ifadesi paket içeriğinde ip.addr 1.2.3.4 den başka bir değer içeren paketleri oku şeklindedir. Bir ip datagramı hem kaynak hemde hedef adreslerini içerir. ifade hedef yada kaynak adreslerinden biri 1.2.3.4 değerinden farklı olsa bile doğru olarak yürütülür.

Eğer ip datagramlarından hedef ve kaynak olarak 1.2.3.4 adresi içeren paketleri filtre etmek istiyorsanız doğru filtre `!(ip.addr == 1.2.3.4)` şeklinde olacaktır.

Örnek ifadeler:

`ip src host 10.0.0.3 && ! arp && port ! 53`

ifadesi ile kaynağı 10.0.0.3 olan, arp paketi olmayan ve dns olmayan paketleri yakalamasını bildiriyoruz.

`tcp port 23 && host 10.0.0.3`

ifadesiyle 10.0.0.3 makinasından yapılacak telnet bağlantıları takip altına alınır.

`tcp port 23 && dst host 10.0.0.2`

İfadesiyle 10.0.0.2 ye yapılacak telnet bağlantılarını takip edebiliriz.

`tcp.port eq 25 or icmp`

İfadesiyle SMTP veya ICMP trafiğini gösterir.

`ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16`

Yerel ağ altındaki iletişimi gösterir.

http.request.method == "POST" && ip.src == 10.0.0.5

İfadesi ile kaynağı 10.0.0.5 olan makinanın trafiği içinde http istek metodu post olan paketleri görüntüler.

PAKETLERLE OYNAMAK

Wireshark ile ağınızda neler olup bittiğini analiz etmeniz için öncelikle wireshark yüklü sistemi uygun bir şekilde ağınıza yerleştirmeniz gerekmektedir.

Wireshark ile analiz yapabileceğiniz olası yapılar

- *Direk trafiği takip edilmek istenen makina kurulumu yapılabilir.
- *Hub kullanılabilir
- *Network TAP/ port spanning, mirroring kullanılabilir
- *ARP poisoning ile trafik bir makinanın üstünden geçirilebilir veya sistemler network TAP gibi kullanılabilir

```
#iptables -A PREROUTING -t mangle -j ROUTE --gw xxx.xxx.xxx.xxx --tee
```

```
#iptables -A POSTROUTING -t mangle -j ROUTE --gw xxx.xxx.xxx.xxx --tee
```

komutlarıyla ilgili makinanın trafiğinin kopyası istenilen makina yönlendirilebilir.

Wireshark deryasında size bütün ihtiyaçlara göre çözümleri anlatamayacağım sadece wiresharkın güzel ayrıntıları üzerinde çalışacağız. Buraya kadar anlattıklarımı anlamak için başlangıç seviyesinde ağ bilgisi yeteriyken buradan sonraki anlatacaklarımı anlamak için biraz daha teknik bilgiye ihtiyacınız olacak. Wiresharkı kurduk, çalıştırdık. Programı çalıştırdıktan sonra ilk işimiz paket yakalamada kullanacağımız ağ arabiriminin seçimi olacak. Bunu **capture options** bölümünden yapacağımız gibi **list the available capture interface** kısa yolundan da yapabiliriz. Capture menüsü altından start dersek program default tanımlanmış özellikler ve ağ arabirimi ile paket yakalamaya başlayacaktır. ihtiyacımız olmayan paketleri capture filters ile eledik ve ayarlamalarımızı yapıp paket yakalamaya başladık. Program çalışıp eşzamanlı olarak patır patır paketleri yakalarken yavrusunu kaybetmiş koyun gibi ekrana bakmamalı amacımız doğrultusunda icraatlara başlamalıyız.

Not:

Wireshark capture options bölümünden update list of packets in real time seçeneğini seçtiyseniz iletişimle eş zamanlı olarak paketleri takip edebilirsiniz.

Topladığımız paketler özet bölümünde dururken incelemek istediğimiz paketi bir kere tıkladığımızda alt pencerede ilgili pakete ilişkin ayrıntılı bilgiler belirir. Paketin

ait olduğu protokol bilgileri, iletişimdeki hedef ve kaynak adresleri, port numaraları gibi protokolü gereği paketin barındırması gereken bütün bilgileri paket ayrıştırma bölümünden görebiliriz. Tabi yukarıda bahsettiğimiz gibi çalışmak istediğimiz paketler doğrultusunda Wireshark'ın güçlü filtreleme özelliğini kullanmalı ve harcamayacağımız vakti %50 azaltmalıyız.

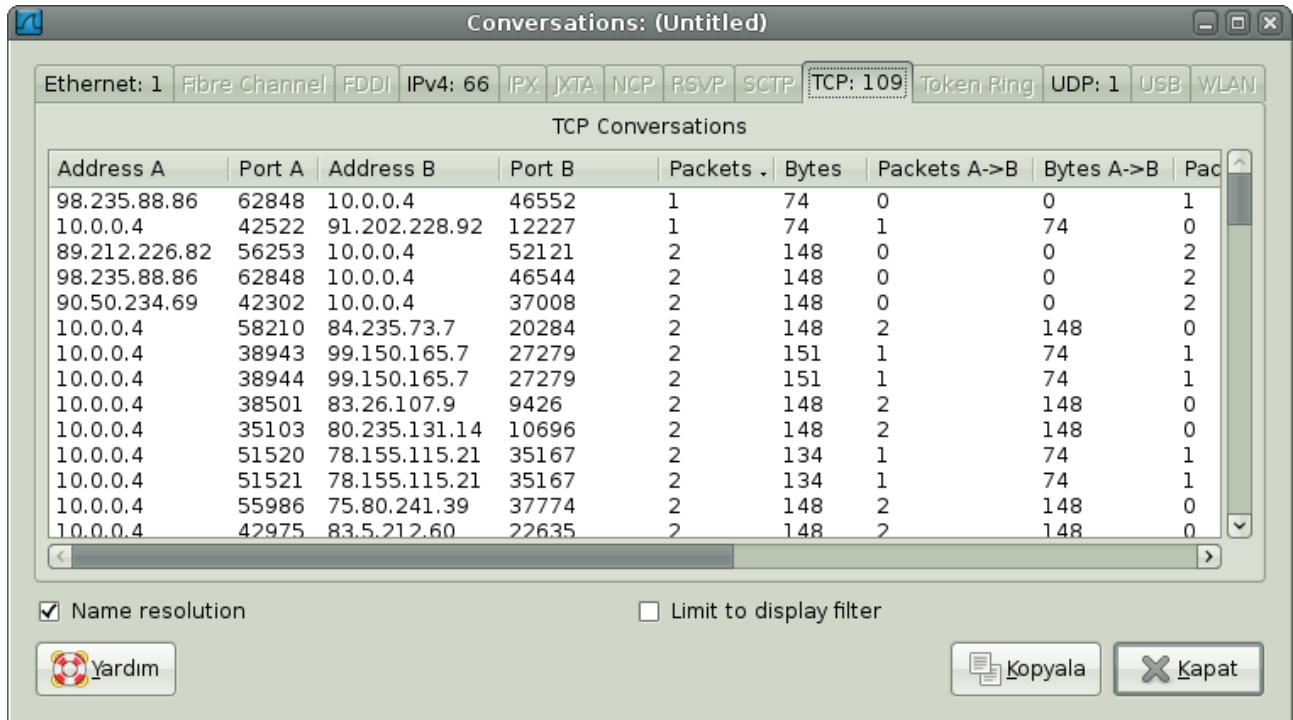
ÖRNEKLER

TORRENT SORUNU

Ağın aşırı yavaşlığından ötürü gelen şikayetler sonrası olaya el atmaya karar verdiniz. Sorunu çözmek için Wireshark'ı kullanacaksınız. Uygun şekilde trafiği kontrol etmeye başladınız ve istemcilerden birinin aşırı trafiği ile karşılaşıyorsunuz.

890	129.517863	10.0.0.4	202.62.69.71	TCP	45978 > 50478 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2841924 TSER=0 V
891	129.456354	89.160.151.52	10.0.0.4	TCP	17887 > 51868 [ACK] Seq=126824 Ack=1054 Win=17133 Len=524 TSV=200155 T
892	129.456382	10.0.0.4	89.160.151.52	TCP	51868 > 17887 [ACK] Seq=1054 Ack=127348 Win=64096 Len=0 TSV=2842158 T
893	129.529926	10.0.0.4	98.235.88.86	TCP	46552 > 62848 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2842177 TSER=0 V
894	129.530075	10.0.0.4	123.231.105.70	TCP	52507 > 45682 [FIN, ACK] Seq=933 Ack=832 Win=8000 Len=0 TSV=2842177 T
895	129.530109	10.0.0.4	91.202.228.92	TCP	42522 > 12227 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2842177 TSER=0 V
896	129.598155	89.160.151.52	10.0.0.4	TCP	17887 > 51868 [ACK] Seq=127348 Ack=1054 Win=17133 Len=524 TSV=200157 T
897	129.598182	10.0.0.4	89.160.151.52	TCP	51868 > 17887 [ACK] Seq=1054 Ack=127872 Win=64096 Len=0 TSV=2842194 T
898	129.646027	89.160.151.52	10.0.0.4	TCP	17887 > 51868 [ACK] Seq=127872 Ack=1054 Win=17133 Len=524 TSV=200157 T
899	129.646051	10.0.0.4	89.160.151.52	TCP	51868 > 17887 [ACK] Seq=1054 Ack=128396 Win=64096 Len=0 TSV=2842206 T
900	129.808340	89.160.151.52	10.0.0.4	TCP	[TCP Previous segment lost] 17887 > 51868 [ACK] Seq=128920 Ack=1054 W
901	129.808371	10.0.0.4	89.160.151.52	TCP	[TCP Dup ACK 899#1] 51868 > 17887 [ACK] Seq=1054 Ack=128396 Win=64096
902	129.878989	86.99.245.79	10.0.0.4	TCP	23664 > 32979 [ACK] Seq=457 Ack=892 Win=66304 Len=0 TSV=764602 TSER=2
903	130.108303	123.231.105.70	10.0.0.4	TCP	45682 > 52507 [ACK] Seq=832 Ack=934 Win=64603 Len=0 TSV=907843 TSER=2
904	130.109549	123.231.105.70	10.0.0.4	TCP	45682 > 52507 [FIN, ACK] Seq=832 Ack=934 Win=64603 Len=0 TSV=907843 T

Örnek kayıt dosyası incelendiğinde bu istemcinin birçok makine ile iletişimde olduğu görülüyor.



Conversation diyalog penceresine baktığımızda TCP :109 durumunun vahametini ortaya koyuyor.

102	37.789061	10.0.0.4	221.18.60.4	TCP	42855 > 42819 [ACK] Seq=1 Ack=1 Win=5856 Len=0 TSV=2819241 TSEQ=
103	37.789094	10.0.0.4	221.18.60.4	TCP	42855 > 42819 [PSH, ACK] Seq=1 Ack=1 Win=5856 [TCP CHECKSUM INCO
104	38.185081	221.18.60.4	10.0.0.4	TCP	42819 > 42855 [PSH, ACK] Seq=1 Ack=547 Win=66560 Len=250 TSV=2711
105	38.185118	10.0.0.4	221.18.60.4	TCP	42855 > 42819 [ACK] Seq=547 Ack=251 Win=6912 Len=0 TSV=2819340 T
106	38.189115	10.0.0.4	221.18.60.4	TCP	42855 > 42819 [PSH, ACK] Seq=547 Ack=251 Win=6912 [TCP CHECKSUM
107	38.405864	10.0.0.4	80.235.131.14	TCP	35110 > 10696 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2819396 TSI
108	38.421875	10.0.0.4	121.219.6.4	TCP	46343 > 26619 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2819400 TSI
109	38.554616	221.18.60.4	10.0.0.4	TCP	42819 > 42855 [FIN, ACK] Seq=251 Ack=671 Win=66560 Len=0 TSV=2711
110	38.554702	10.0.0.4	221.18.60.4	TCP	42855 > 42819 [FIN, ACK] Seq=671 Ack=252 Win=6912 Len=0 TSV=2819
111	38.554743	10.0.0.4	221.18.60.4	TCP	42856 > 42819 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=2819433 TSI
112	38.916330	221.18.60.4	10.0.0.4	TCP	42819 > 42855 [ACK] Seq=252 Ack=672 Win=66560 Len=0 TSV=27101221
113	38.921482	221.18.60.4	10.0.0.4	TCP	42819 > 42856 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1432 WS=
114	38.921512	10.0.0.4	221.18.60.4	TCP	42856 > 42819 [ACK] Seq=1 Ack=1 Win=5856 Len=0 TSV=2819524 TSE=
115	38.921562	10.0.0.4	221.18.60.4	BitTorrent	Handshake
116	39.288022	221.18.60.4	10.0.0.4	TCP	42819 > 42856 [FIN, ACK] Seq=1 Ack=1 Win=260 Len=0 TSV=27101259
117	39.288121	10.0.0.4	221.18.60.4	TCP	42856 > 42819 [FIN, ACK] Seq=1 Ack=2 Win=183 Len=0 TSV=2819616 T

▶ Frame 115 (134 bytes on wire, 134 bytes captured)
 ▶ Ethernet II, Src: AsustekC_8e:72:2d (00:13:d4:8e:72:2d), Dst: TaicomDa_53:67:18 (00:d0:da:53:67:18)
 ▶ Internet Protocol, Src: 10.0.0.4 (10.0.0.4), Dst: 221.18.60.4 (221.18.60.4)
 ▶ Transmission Control Protocol, Src Port: 42856 (42856), Dst Port: 42819 (42819), Seq: 4294967229, Ack: 1, Len: 68
 ▼ BitTorrent
 Protocol Name Length: 19
 Protocol Name: BitTorrent protocol
 Reserved Extension Bytes: 0000000000100000
 SHA1 Hash of info dictionary: 29CD237E4F38E8864A53195CAF151B8FE7BAB1D8
 Peer ID: 2D5452313232302D353278723633713462673734

Paketleri ayrı ayrı incelediğimizde 115. pakette görüldüğü üzere sorun torrentten kaynaklanıyor.Artık yönetici olarak yapılacaklar size kalmış.Umarsızca download yapan istemcimize tatlı dille durumu anlatabilir yada tam download kasarken arkadan koşup bi uçan tekme patlatabilirsiniz :))

WIRESHARK İLE VERİ MADENCİLİĞİ

Günümüzde Youtube ve türevi birçok site üzerinden video izlenebilmektedir.Bunların çoğunda ise izlenilen videoyu kaydetmek sitenin sunduğu olanaklar dahilinde mümkün değildir.Örneğimiz youtube üzerinden izlenilen videonun wireshark ile kaynağının bulunması ve kaydedilmesi üzerinedir.Wireshark programını başlattıktan sonra www.youtube.com adresine girip izlemek istediğimiz videoyu çağırıyoruz.Video oynamaya başladığında wiresharkda kenarda kendi işini yapıyor.Wiresharkı durdurup http paketleri içinden GET isteklerini **http.request.method == "GET"** ifadesiyle filtreliyoruz.

Wireshark interface showing a filtered list of HTTP GET requests. The filter is `http.request.method == "GET"`. The list shows various video requests with their source (ytimg.l.google.com) and destination (youtube.com). The selected packet is a GET request for a video with ID `u_t0RtmKijE&t=vjVQalPpcFPi-MnFFpPQ2bJlM3wPCGTHJtbETO6LLUA=&el=detailpage&ps=6&mt=34`.

Hypertext Transfer Protocol

GET /get_video?video_id=u_t0RtmKijE&t=vjVQalPpcFPi-MnFFpPQ2bJlM3wPCGTHJtbETO6LLUA=&el=detailpage&ps=6&mt=34 HTTP/1.1\r\n

Host: www.youtube.com\r\n

User-Agent: Mozilla/5.0 (X11; U; Linux i686; tr-TR; rv:1.9.0.7) Gecko/2009032803 Iceweasel/3.0.6 (Debian-3.0.6-1)\r\n

0000 00 d0 da 53 67 18 00 13 d4 8e 72 2d 08 00 45 00 ...Sg...r...E.

0010 05 c0 12 10 40 00 40 06 5c 5e 0a 00 00 0e d0 75 ...@.e. ^...u

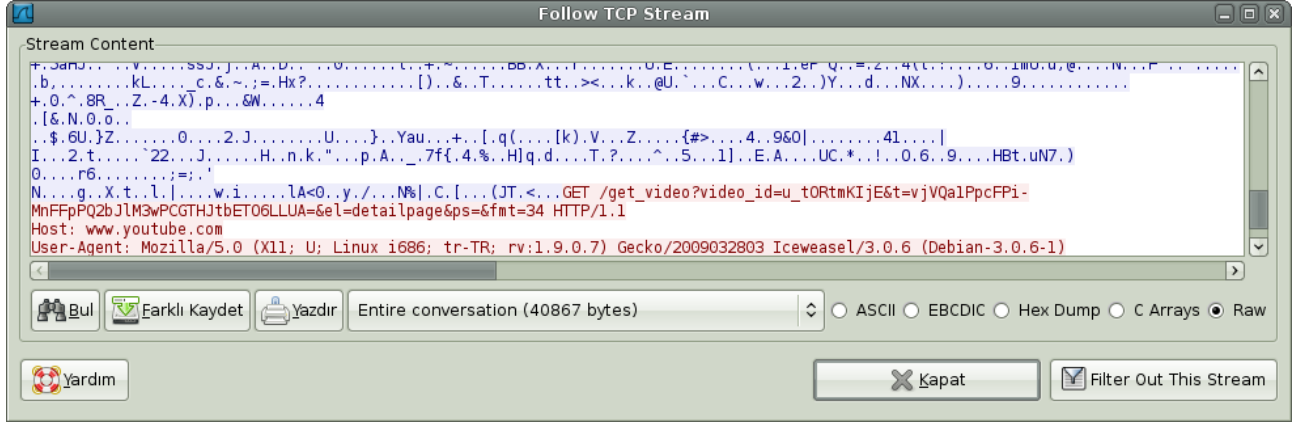
0020 ec 46 e8 aa 00 50 58 6c 73 f1 44 58 c5 17 50 10 ...F...PXL s.DX..P.

0030 16 20 cc 7c 00 00 47 45 54 20 2f 67 65 74 5f 76 ...|.GE T /get_v

0040 69 64 65 6f 3f 76 69 64 65 6f 5f 69 64 3d 75 5f ...ideo?vid_id=u

Frame (frame), 1486 bytes Packets: 535 Displayed: 13 Marked: 0 Dropped: 0 Profile: Default

Önümüze gelen filtrelenmiş get isteklerinden birisi ilgili videoya ait kaynak adresidir.İşimize yarayacak adresi o paketler içinden bulabileceğimiz gibi hedef adrese göre sağ tıklayıp **follow tcp stream** özelliğini seçerek oradan da adresi çıkartabiliriz.Follow tcp stream penceresinde kırmızı yazıyla yazılanlar bizim tarafımızdan yollanan paketler mavi yazıyla yazılanlar ise karşı tarafın yolladığı verilerdir(bu renkler değiştirilebilir).



Şekildede görüldüğü üzere videoyu çağırdığımız konum **/get_video?video_id=u_tORtmKIjE&t=vjVQa1PpcFPi-MnFFpPQ2bJlM3wPCGTHJtbETO6LLUA=&el=detailpage&ps=&fmt=34** olarak karşımıza çıkıyor.

www.youtube.com/get_video?video_id=u_tORtmKIjE&t=vjVQa1PpcFPbQlJO7dWuHgsHvRqeOf1JBCn2N76wl5w=&el=detailpage&ps=&fmt=34 adresini browserımıza yapıştırıp videoyu çekebiliyoruz.

Ağımızın güvenliği konusunda wireshark paket toplayıp bize haber vermekten öte özelleştirilmiş birkaç filtre ve fonksiyon ile bir uyarı mekanizması gibi çalışabilir.Daha önceden bahsettiğimiz üzere ağımızda arp protokolüne yönelik herhangi bir saldırı durumunda arp trafiğindeki anormallikleri, worm saldırılarına yönelik bilgileri ve syn flooding saldırılarına yönelik tespitleri hazırlayacağımız uyarı filtreleri ve renklendirme özellikleri ile kolayca farkedilebilir hale getirebiliriz.

ARP POISONING TESPİTİ

ARP zehirlenmesi saldırısına örnek olarak şekilde de görüldüğü üzere 10.0.0.14 adresli makinadan ağa arp request paketleri yağmaktadır. Filtre ifadesi olarak arp yazarsak yakalanan paketler arasında arp protokolüne yönelik olanları filtreler ve olası bir saldırı hakkında bilgi sahibi oluruz.

No.	Time	Source	Destination	Protocol	Info
6	0.066416	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.188? Tell 10.0.0.14
7	0.077894	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.157? Tell 10.0.0.14
8	0.089360	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.253? Tell 10.0.0.14
9	0.099528	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.169? Tell 10.0.0.14
10	0.109676	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.131? Tell 10.0.0.14
11	0.119815	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.236? Tell 10.0.0.14
12	0.129967	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.121? Tell 10.0.0.14
13	0.140140	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.226? Tell 10.0.0.14
14	0.150303	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.181? Tell 10.0.0.14
15	0.160458	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.241? Tell 10.0.0.14
16	0.170626	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.221? Tell 10.0.0.14
17	0.180771	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.193? Tell 10.0.0.14
18	0.190930	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.151? Tell 10.0.0.14
19	0.201095	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.191? Tell 10.0.0.14
20	0.211240	AsustekC_8e:72:2d	Broadcast	ARP	Who has 10.0.0.132? Tell 10.0.0.14

BLASTER WORMU TESPİTİ

Bir diğer örneğimiz blaster wormuna ilişkin. İlgili kullanıcı makinasının istemsiz 60 saniye içinde kapanmasından şikayet ediyor.Port mirroring kullanarak ilgili makinanın trafiği kontrol edildiğinde ağ altında başka bir makinarya paketler yollamakta olduğu görülüyor.Standart hedef portu 4444 gelen paketlerden ağda bir worm tehlikesi olduğu anlaşılıp gereken işlemler yapılır.

▶ Frame 18 (60 bytes on wire, 60 bytes captured)
 ▶ Ethernet II, Src: Cisco_3c:3f:a8 (00:01:96:3c:3f:a8), Dst: AmbitMic_aa:af:80 (00:d0:59:aa:af:80)
 ▶ Internet Protocol, Src: 10.234.0.239 (10.234.0.239), Dst: 10.234.2.116 (10.234.2.116)
 ▼ Transmission Control Protocol, Src Port: rsc-robot (1793), Dst Port: krb524 (4444), Seq: 31, Len: 0
 Source port: rsc-robot (1793)
 Destination port: krb524 (4444)
 Sequence number: 31 (relative sequence number)
 Acknowledgment number: Broken TCP. The acknowledge field is nonzero while the ACK flag is not set
 Header length: 20 bytes
 ▶ Flags: 0x04 (RST)
 Window size: 0
 ▶ Checksum: 0x9ced [incorrect, should be 0xa0ed (maybe caused by "TCP checksum offload"?)]
 ▶ [SEQ/ACK analysis]

```

0000 00 d0 59 aa af 80 00 01 96 3c 3f a8 08 00 45 00  ..Y....<?...E.
0010 00 28 08 f2 40 00 7f 06 d9 a7 0a ea 00 ef 0a ea  .(..@... ..
0020 02 74 07 01 11 5c 76 be 16 6e cd 5a 82 d8 50 04  .t...v...n.Z..P.
0030 00 00 9c ed 00 00 00 00 00 00 00 00 00 00 00 00  .....
  
```

Transmission Control Protocol (tcp), ... Packets: 18 Displayed: 18 Marked: 0 Profile: Default

Şekildede görüldüğü gibi hedef port 4444 e sorunlu makinadan paketler gönderilmekte.

```

0000 00 80 ad d1 84 d7 00 d0 59 aa af 80 08 00 45 00  .....Y....E.
0010 00 3c 00 3a 40 00 80 06 e1 4b 0a ea 02 74 0a ea  .<.:@... .K...t..
0020 00 ef 11 5c 07 01 cd 5a 82 b2 76 be 16 50 50 18  ...\.Z..V..PP.
0030 fb db 73 31 00 00 0d 0a 43 3a 5c 57 49 4e 4e 54  ..s1... C:\WINNT
0040 5c 73 79 73 74 65 6d 33 32 3e                   \system3 2>
  
```

Data (data), 20 bytes

Paketleri incelediğimizde bir şeyin sistem dosyalarımıza erişmeye çalıştığı görülüyor.

```

0000 00 d0 59 aa af 80 00 01 96 3c 3f a8 08 00 45 00  ..Y....<?...E.
0010 00 3a 08 ef 40 00 7f 06 d9 98 0a ea 00 ef 0a ea  .:..@... ..
0020 02 74 07 01 11 5c 76 be 16 50 cd 5a 82 c6 50 18  .t...v...P.Z..P.
0030 43 9e a0 4d 00 00 73 74 61 72 74 20 6d 73 62 6c  C..M..st art msbl
0040 61 73 74 2e 65 78 65 0a                          ast.exe.
  
```

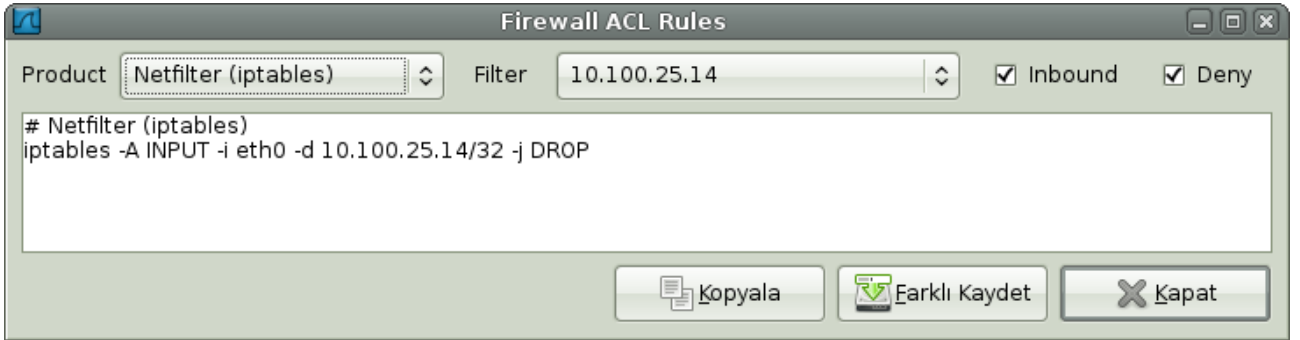
Data (data), 18 bytes

Bir diğer pakette ise sorunun tam olarak ne olduğu ortada.Verit bölümünde start msblast.exe ifadesi ile sorunun adı konmuş durumda.

PORT TARAMA TESPİTİ

1	0.000000	10.100.25.14	10.100.18.12	TCP	syscomlan > netbios-ssn [SYN] Seq=0 Win=8 Len=0
2	0.100476	10.100.25.14	10.100.18.12	TCP	19491 > epmap [SYN] Seq=0 Win=8 Len=0
3	0.201152	10.100.25.14	10.100.18.12	TCP	7358 > microsoft-ds [SYN] Seq=0 Win=8 Len=0
4	0.301714	10.100.25.14	10.100.18.12	TCP	27524 > http [SYN] Seq=0 Win=8 Len=0
5	0.403133	10.100.25.14	10.100.18.12	TCP	20193 > ssh [SYN] Seq=0 Win=8 Len=0
6	0.503604	10.100.25.14	10.100.18.12	TCP	1023 > printer [SYN] Seq=0 Win=8 Len=0
7	0.607512	10.100.25.14	10.100.18.12	TCP	16748 > telnet [SYN] Seq=0 Win=8 Len=0
8	0.707986	10.100.25.14	10.100.18.12	TCP	12502 > ftp [SYN] Seq=0 Win=8 Len=0
9	0.808340	10.100.25.14	10.100.18.12	TCP	30382 > x11 [SYN] Seq=0 Win=8 Len=0
10	0.904949	10.100.25.14	10.100.18.12	TCP	27986 > blackjack [SYN] Seq=0 Win=8 Len=0
11	1.004235	10.100.25.14	10.100.18.12	TCP	25488 > smtp [SYN] Seq=0 Win=8 Len=0
12	1.110883	10.100.25.14	10.100.18.12	TCP	6729 > sunrpc [SYN] Seq=0 Win=8 Len=0
13	1.212836	10.100.25.14	10.100.18.12	TCP	29169 > 1028 [SYN] Seq=0 Win=8 Len=0
14	1.307771	10.100.25.14	10.100.18.12	TCP	24305 > hp-pdl-datastr [SYN] Seq=0 Win=8 Len=0
15	1.407052	10.100.25.14	10.100.18.12	TCP	17851 > solid-mux [SYN] Seq=0 Win=8 Len=0
16	1.512738	10.100.25.14	10.100.18.12	TCP	10985 > finger [SYN] Seq=0 Win=8 Len=0
17	1.614648	10.100.25.14	10.100.18.12	TCP	ifor-protocol > dantz [SYN] Seq=0 Win=8 Len=0

Sistemin farklı servislerine kısa zamanda gelen bağlantı istekleri port taraması olarak bilinir. Wireshark paket özetleri bölümümüzde görüldüğü üzere 10.100.25.14 makinasından 10.100.18.12 adresli makinanın farklı portlarına gönderilen syn bayraklı TCP paketleri ilgili servisin çalışıp çalışmadığını yoklayan birinin varlığına işaret.



Firewall acl rules özelliği ile kaynak adresten gelen paketleri drop edecek kural ifadesini oluşturabiliriz.

SYN FLOODING TESPİTİ

TCP protokolünün tasarım özelliklerinden kaynaklanan sorunlardan teki syn flooding saldırılarına zemin hazırlamıştır. Bir tcp bağlantısı kurulması 3 adımda gerçekleşen ve 3 way handshake adı verilen prosedürle oluşturulmaktadır. Burada bağlantı kurmak isteyen atar taraf syn bayrağı set edilmiş tcp paketini ilgili makinanın ilgili servisine yollar. Bağlantı kurulmasında herhangi bir sorun yoksa sunucu taraf istemciye syn,ack bayrakları set edilmiş paket yollar ve bunu alan istemci ack bayraklı tcp paketi ile bağlantıyı gerçekleştirir. Burada bir servis kendisine gelen syn bayraklı paketlere karşılık syn,ack atmak üzere programlandığından kötü niyetli birinin fazlaca syn bayraklı paketlerle bağlantı isteğinde bulunması sunucunun bir yerden sonra syn,ack paketi gönderemeyecek duruma getirir ve servisin durmasına sebep olur. Örnek pcap görüntüsündeki Syn flooding saldırısında görüldüğü üzere değiştirilmiş ip adreslerinden hedefin 445 inci portuna deli gibi syn bayraklı paketler yağdırılmakta.

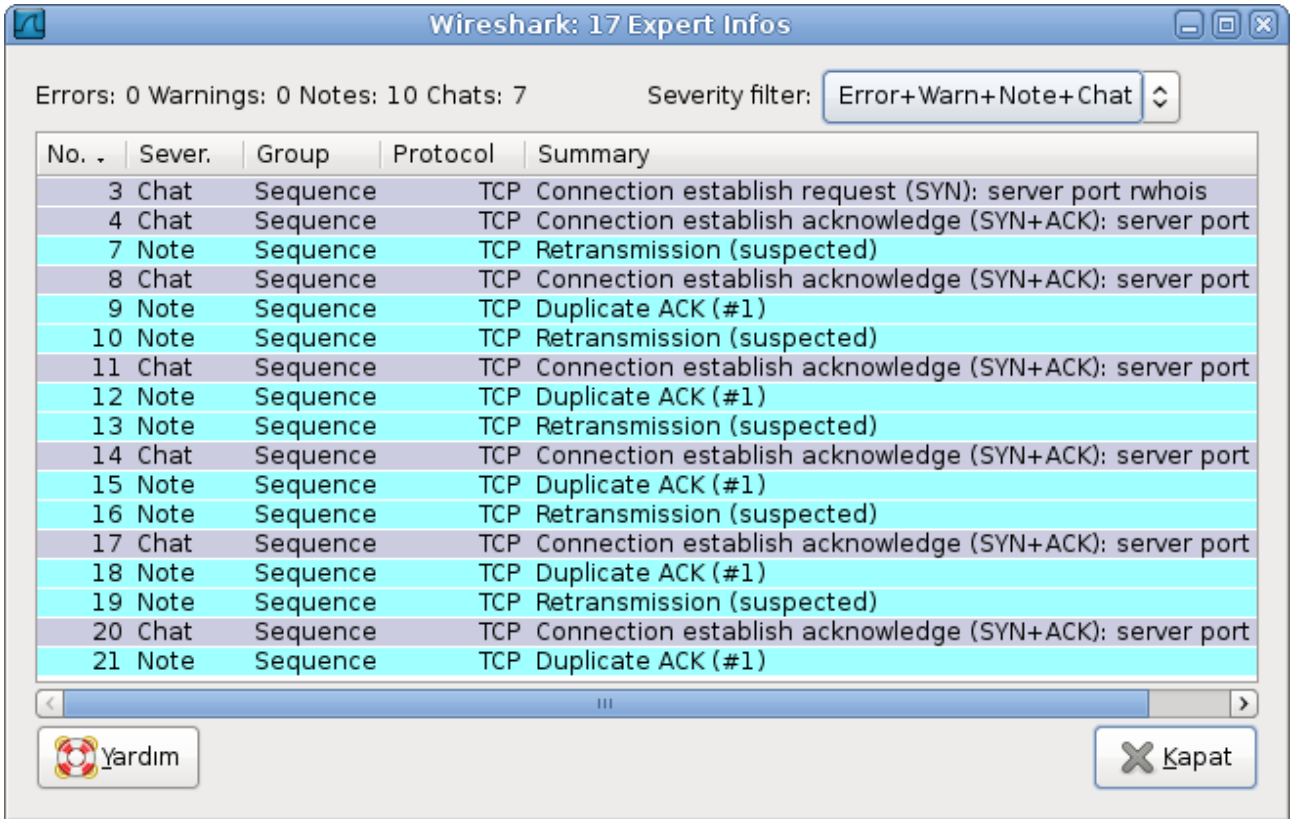
No.	Time	Source	Destination	Protocol	Info
314786	15.296600	229.3.109.76	10.0.0.2	TCP	55266 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314787	15.296624	209.237.143.221	10.0.0.2	TCP	55267 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314788	15.296649	243.205.237.149	10.0.0.2	TCP	55268 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314789	15.296673	66.161.225.30	10.0.0.2	TCP	55269 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314790	15.296697	30.30.66.223	10.0.0.2	TCP	55270 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314791	15.296721	194.130.83.191	10.0.0.2	TCP	55271 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314792	15.297172	192.198.231.109	10.0.0.2	TCP	55272 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314793	15.297199	194.241.135.217	10.0.0.2	TCP	55273 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314794	15.297224	148.248.234.209	10.0.0.2	TCP	55274 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314795	15.297268	136.208.78.131	10.0.0.2	TCP	55275 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314796	15.297292	3.56.233.68	10.0.0.2	TCP	55276 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314797	15.297317	136.253.242.31	10.0.0.2	TCP	55277 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314798	15.297341	81.193.184.216	10.0.0.2	TCP	55278 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
314799	15.297365	12.71.65.68	10.0.0.2	TCP	55279 > microsoft-ds [SYN] Seq=0 Win=512 Len=0
Frame 314793 (54 bytes on wire, 54 bytes captured)					
Ethernet II, Src: AsustekC_8e:72:2d (00:13:d4:8e:72:2d), Dst: TaicomDa_53:67:18 (00:d0:da:53:67:18)					
Internet Protocol, Src: 194.241.135.217 (194.241.135.217), Dst: 10.0.0.2 (10.0.0.2)					
Transmission Control Protocol, Src Port: 55273 (55273), Dst Port: microsoft-ds (445), Seq: 0, Len: 0					
0000 00 d0 da 53 67 18 00 13 d4 8e 72 2d 08 00 45 00 ...Sg... ..E.					
0010 00 28 5e 6f 00 00 40 06 c7 94 c2 f1 87 d9 0a 00 .(^o..@.					
0020 00 02 d7 e9 01 bd 6c 1a 62 10 2d a7 d3 5c 50 02l. b...P.					
0030 02 00 b0 40 00 00@..					

BAĞLANTI SORUNU

2	0.090367	68.87.76.178	67.161.32.69	DNS	Standard query response A 65.201.175.19
3	0.100131	67.161.32.69	65.201.175.19	TCP	combox-web-acc > rwhois [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2
4	0.184831	65.201.175.19	67.161.32.69	TCP	rwhois > combox-web-acc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
5	0.184887	67.161.32.69	65.201.175.19	TCP	combox-web-acc > rwhois [ACK] Seq=1 Ack=1 Win=256960 Len=0
6	0.185056	67.161.32.69	65.201.175.19	TCP	combox-web-acc > rwhois [PSH, ACK] Seq=1 Ack=1 Win=256960 Len=14
7	3.167930	67.161.32.69	65.201.175.19	TCP	[TCP Retransmission] combox-web-acc > rwhois [PSH, ACK] Seq=1 Ack=1 W
8	4.068284	65.201.175.19	67.161.32.69	TCP	rwhois > combox-web-acc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
9	4.068363	67.161.32.69	65.201.175.19	TCP	[TCP Dup ACK 7#1] combox-web-acc > rwhois [ACK] Seq=15 Ack=1 Win=2569
10	9.086813	67.161.32.69	65.201.175.19	TCP	[TCP Retransmission] combox-web-acc > rwhois [PSH, ACK] Seq=1 Ack=1 W
11	10.077707	65.201.175.19	67.161.32.69	TCP	rwhois > combox-web-acc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
12	10.077753	67.161.32.69	65.201.175.19	TCP	[TCP Dup ACK 10#1] combox-web-acc > rwhois [ACK] Seq=15 Ack=1 Win=256
13	21.125214	67.161.32.69	65.201.175.19	TCP	[TCP Retransmission] combox-web-acc > rwhois [PSH, ACK] Seq=1 Ack=1 W
14	22.273257	65.201.175.19	67.161.32.69	TCP	rwhois > combox-web-acc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
15	22.273301	67.161.32.69	65.201.175.19	TCP	[TCP Dup ACK 13#1] combox-web-acc > rwhois [ACK] Seq=15 Ack=1 Win=256
16	45.101697	67.161.32.69	65.201.175.19	TCP	[TCP Retransmission] combox-web-acc > rwhois [PSH, ACK] Seq=1 Ack=1 W
17	46.287080	65.201.175.19	67.161.32.69	TCP	rwhois > combox-web-acc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
18	46.287124	67.161.32.69	65.201.175.19	TCP	[TCP Dup ACK 16#1] combox-web-acc > rwhois [ACK] Seq=15 Ack=1 Win=256
19	75.097386	67.161.32.69	65.201.175.19	TCP	[TCP Retransmission] combox-web-acc > rwhois [PSH, ACK] Seq=1 Ack=1 W
20	94.555677	65.201.175.19	67.161.32.69	TCP	rwhois > combox-web-acc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=146
21	94.555726	67.161.32.69	65.201.175.19	TCP	[TCP Dup ACK 19#1] combox-web-acc > rwhois [ACK] Seq=15 Ack=1 Win=256

Örnekte görüldüğü üzere 3 yollu el sıkışmanın ardından gönderilen psh,ack bayraklı 6. paket ack cevabının gelmemesi üzerine 7. pakette tekrar yollanıyor.Bu sırada b makinası a nın ack bayraklı paketini almadığından ötürü syn,ack paketini tekrar yolluyor. sonrası ise kısır döngü.A makinasının ack onayını iletme çabası ile yolladığı dub ack paketleri ve psh ack bayraklı paketini iletme için yolladığı tcp retransmission paketleri ne karşılık b makinasının devamlı oalrak ack onayını almak için yolladığı syn ack bayraklı paketi.

Expert info penceresine baktığımızda sorun daha açık şekilde görülmekte



CASUS YAZILIM TESPİTİ

Aşağıdaki örnekte kullanıcı devamlı olarak ana sayfasının değişmesinden şikayetçi.İlgili makinaya wireshark kurarak trafiğini kontrol ediyoruz.Nete çıkan hiçbir uygulama açmadığımız durumda 5 numaralı pakette görüldüğü üzere http GET isteği ile internetteki bir adresten birşey çağırılmakta

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.184	24.46.230.187	TCP	1038 > 1706 [SYN] Seq=0 Len=0 MSS=1460
2	0.000019	192.168.0.184	69.206.234.66	TCP	1039 > 3531 [SYN] Seq=0 Len=0 MSS=1460
3	0.001354	192.168.0.184	24.46.230.187	TCP	1038 > 1706 [SYN] Seq=0 Len=0 MSS=1460
4	0.002375	192.168.0.184	69.206.234.66	TCP	1039 > 3531 [SYN] Seq=0 Len=0 MSS=1460
5	0.338822	192.168.0.184	64.124.109.200	HTTP	GET /command/commandv6.07.asp?key=&t=26962 HTTP/1.1
6	0.340546	192.168.0.184	64.124.109.200	HTTP	[TCP Out-of-Order] GET /command/commandv6.07.asp?key=&t=26962 HTTP/1.1
7	0.638241	192.168.0.184	64.124.109.200	TCP	1040 > http [ACK] Seq=286 Ack=243 Win=65041 Len=0
8	0.638386	192.168.0.184	64.124.109.200	TCP	[TCP Dup ACK 7#1] 1040 > http [ACK] Seq=286 Ack=243 Win=65041 Len=0
9	0.800253	192.168.0.184	64.124.109.200	TCP	1040 > http [ACK] Seq=286 Ack=613 Win=64671 Len=0
10	0.800403	192.168.0.184	64.124.109.200	TCP	[TCP Dup ACK 9#1] 1040 > http [ACK] Seq=286 Ack=613 Win=64671 Len=0

Ayrıntı penceresinde gördüklerimiz işgillendiriyor bizi :D

Hypertext Transfer Protocol	
GET /command/commandv6.07.asp?key=&t=26962 HTTP/1.1\r\n	
Request Method: GET	
Request URI: /command/commandv6.07.asp?key=&t=26962	
Request Version: HTTP/1.1	
User-Agent: Mozilla/3.0 (compatible; MSIE 4.0; win32)\r\n	
Host: command.weatherbug.com\r\n	
Connection: keep-alive\r\n	
Cookie: wxbug_cookie=has_cookies=1; RMID=4aecf9dc45a025d0; RMFD=011h3k3t0104ym 01058k; RMFS=011h3khLUI052U; LMB1per12h=1\r\n	

Paket ayrıntılarına baktığımızda bir uygulama command.weatherbug.com adresinin /command/commandv6.07.asp?key=&t=26962 konumundan bir dosya download etmeye çalışıyor.

No. -	Time	Source	Destination	Protocol	Info
11	3.725242	192.168.0.184	205.152.37.23	DNS	Standard query A deskwx.weatherbug.com
12	3.734060	192.168.0.184	205.152.37.23	DNS	Standard query A deskwx.weatherbug.com

11 ve 12. paketlere baktığımızda ise deskwx.weatherbug.com adresi için DNS sorguları görülmekte.İlgili makinayı kontrol ettiğimizde ise her açılışta arka planda çalışmaya başlayan uygulamanın paketlerde görülen adrese bağlandığını ve indirme işlemini gerçekleştirdiğini bununla beraber anasayfanın değiştiğini görüyoruz.Uygulamanın kaldırılması ile sorun çözülüyor.

OS FINGERPRINTING TESPİTİ (ICMP TABANLI)

Bilindiği üzere uzaktan işletim sistemi tespit etme yöntemlerinden biri sistemlerin döndüğü karakteristik icmp hata mesajlarıdır.Bu sebepten ötürü saldırganlar OS belirlemede hedef sisteme farklı tipte icmp paketleri yollayıp geri dönen cevapların dönüş süreleri gibi sistemden sisteme değişen ayırdedici özelliklerden faydalanırlar.

6	0.408876	10.0.0.2	10.0.0.29	ICMP	Echo (ping) reply
7	0.620683	10.0.0.29	10.0.0.2	ICMP	Echo (ping) request
8	0.620859	10.0.0.2	10.0.0.29	ICMP	Echo (ping) reply
9	0.847146	10.0.0.29	10.0.0.2	ICMP	Echo (ping) request
10	0.847295	10.0.0.2	10.0.0.29	ICMP	Echo (ping) reply
11	1.863030	10.0.0.29	10.0.0.2	ICMP	Timestamp request
12	1.863238	10.0.0.2	10.0.0.29	ICMP	Timestamp reply
13	1.869470	10.0.0.29	10.0.0.2	ICMP	Timestamp request
14	1.869609	10.0.0.2	10.0.0.29	ICMP	Timestamp reply
15	2.739445	10.0.0.29	10.0.0.2	ICMP	Address mask request
16	2.742531	10.0.0.29	10.0.0.2	ICMP	Address mask request
17	7.062589	10.0.0.29	10.0.0.2	ICMP	Information request
18	7.064628	10.0.0.29	10.0.0.2	ICMP	Information request
19	11.354823	10.0.0.29	10.0.0.2	ICMP	Echo (ping) request
20	11.355045	10.0.0.2	10.0.0.29	ICMP	Echo (ping) reply

Şekilde de görüldüğü üzere saldırgan hedef sistem üzerinde ICMP tabanlı işletim sistemi saptama taraması yapıyor.Farklı türlerde gelen icmp paketleri bizi durumdan haberdar ediyor.

NOT:

Normal trafik sırasında 13, 15, 17 gibi ICMP türleri kullanılmadığı için herhangi bir şüpheli durumda `icmp.type == 13 || icmp.type == 17 || icmp.type == 15` gibisinden bir filtre ile icmp tabanlı sistem belirleme saldırısına maruz kalıp kalmadığımızı anlayabiliriz.

NEDEN SSH

Wireshark ile sadece sistem yöneticileri ağı takip edebilir diye bi kaide yok.Hackerlarda bu güçlü snifferi kullanıyorlar.Yukarıda wireshark ile analiz yapabileceğiniz olası yapıları yazmıştık.Hepimizin bildiği üzere telnet, rlogin gibi zamanı geçmiş uzaktan erişime imkan tanıyan protokoller veriyi şifrelemeden düz metin olarak yolladıklarından ötürü güvensizdir.Herhangibir sniffing saldırısında iletişim gün gibi ortadadır.Buna karşılık ssh iletişim sırasında sağlam şifreleme sistemi kullandığından araya herhangi bir saldırganın girmesi durumunda yakaladığı veri anlamsız olacaktır. Şimdi bu durumları snifflenmiş oturumlarla gösterirsek..

83	75.853847	192.168.1.2	192.168.1.3	TELNET	Telnet Data ...
84	75.854916	192.168.1.2	192.168.1.3	TELNET	[TCP Out-Of-Order] Telnet Data ...
87	77.334968	192.168.1.3	192.168.1.2	TELNET	Telnet Data ...
88	77.351319	192.168.1.3	192.168.1.2	TELNET	[TCP Retransmission] Telnet Data ...
89	77.368651	192.168.1.2	192.168.1.3	TELNET	Telnet Data ...
90	77.369965	192.168.1.2	192.168.1.3	TELNET	[TCP Out-Of-Order] Telnet Data ...
93	77.493408	192.168.1.2	192.168.1.3	TELNET	Telnet Data ...
94	77.496278	192.168.1.2	192.168.1.3	TELNET	[TCP Out-Of-Order] Telnet Data ...
97	79.024691	192.168.1.3	192.168.1.2	TELNET	Telnet Data ...
98	79.041217	192.168.1.3	192.168.1.2	TELNET	[TCP Retransmission] Telnet Data ...
99	79.092111	192.168.1.2	192.168.1.3	TELNET	Telnet Data ...
100	79.093207	192.168.1.2	192.168.1.3	TELNET	[TCP Out-Of-Order] Telnet Data ...
103	85.547606	192.168.1.3	192.168.1.2	TELNET	Telnet Data ...

Telnet

Data: BAT\r\n
Data: 01/19/2004 09:45 PM 0 CONFIG.SYS\r\n
Data: 06/26/2004 12:12 PM <DIR> Documents and Settings\r\n
Data: 02/03/2005 11:40 PM <DIR> EasyBoot\r\n
Data: 02/29/2004 02:51 PM 11,531 installer-debug.txt\r\n
Data: 12/19/2004 12:50 AM <DIR> mga\r\n
Data: 12/19/2004 12:51 AM <DIR> mgafold\r\n
Data: 11/24/2004 07:47 PM <DIR> mnt\r\n

Saldırgan halihazırda kurulmuş olan telnet oturumunu görmek için ilgili makinalara ait trafiği gösterecek şekilde filtre dizisini yazar.(ip.addr eq 192.168.1.2 and ip.addr eq 192.168.1.3).Yakalanan paketler telnet oturumunda neler döndüğünü kabak gibi ortaya koyuyor.Wiresharkın follow tcp stream özelliği ilede saldırı paketleri ayrı ayrı kurcalamaktan kurtulur.

28	2.433172	10.0.0.5	SSHv2	Encrypted response packet len=32
29	2.433206	10.0.0.5	TCP	56214 > ssh [ACK] Seq=1265 Ack=1757 Win=10112 Len=0 TSV=3757330 TSER=
30	2.433382	10.0.0.5	SSHv2	Encrypted request packet len=64
31	2.474654	10.0.0.5	SSHv2	Encrypted response packet len=48
32	2.474832	10.0.0.5	SSHv2	Encrypted request packet len=448
33	2.541704	10.0.0.5	SSHv2	Encrypted response packet len=112
34	2.563725	10.0.0.5	SSHv2	Encrypted response packet len=480
35	2.563777	10.0.0.5	TCP	56214 > ssh [ACK] Seq=1777 Ack=2397 Win=11552 Len=0 TSV=3757362 TSER=
36	2.755745	10.0.0.5	SSHv2	Encrypted response packet len=64
37	2.761208	10.0.0.5	SSHv2	Encrypted response packet len=48
38	2.761260	10.0.0.5	TCP	56214 > ssh [ACK] Seq=1777 Ack=2509 Win=11552 Len=0 TSV=3757412 TSER=
39	3.792005	10.0.0.5	SSHv2	Encrypted request packet len=48
40	3.827833	10.0.0.5	SSHv2	Encrypted response packet len=48

Frame 33 (178 bytes on wire (178 bytes captured))

Ethernet II, Src: TaicomDa_53:67:18 (00:d0:da:53:67:18), Dst: AsustekC_8e:72:2d (00:13:d4:8e:72:2d)
Internet Protocol, Src: 10.0.0.5 (10.0.0.5), Dst: 10.0.0.5 (10.0.0.5)
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 56214 (56214), Seq: 1805, Ack: 1777, Len: 112
SSH Protocol
SSH Version 2
Encrypted Packet: 8ECE1C9B107D1985F3501BD2DBAAA772D581E85723F31B88...

SSH ta ise durum farklı.Görüldüğü üzere Sniffleyen için anlamsız şifreli paketler iletiliyor.

MESSENGER ÜZERİNE

Messenger ve türevi yazılımlar yazılanları düz metin olarak ilettiğinden ağ dinleyen biri msn görüşmelerini olduğu gibi yakalayabilir.

No.	Time	Source	Destination	Protocol	Info
127	17.870000	65.54.171.48	10.0.0.14	MSNMS	MSG
128	17.876683	65.54.171.48	10.0.0.14	MSNMS	[TCP Out-Of-Order] MSG
135	19.021485	207.46.26.110	10.0.0.14	MSNMS	MSG
136	19.021587	207.46.26.110	10.0.0.14	MSNMS	[TCP Out-Of-Order] MSG
141	19.469592	207.46.26.110	10.0.0.14	MSNMS	MSG
142	19.469684	207.46.26.110	10.0.0.14	MSNMS	[TCP Out-Of-Order] MSG
145	22.937571	65.54.171.48	10.0.0.14	MSNMS	MSG
146	22.937660	65.54.171.48	10.0.0.14	MSNMS	[TCP Out-Of-Order] MSG
149	23.770240	65.54.171.48	10.0.0.14	MSNMS	MSG
150	23.770331	65.54.171.48	10.0.0.14	MSNMS	[TCP Out-Of-Order] MSG

Frame 150 (464 bytes on wire (464 bytes captured) on interface 0)

Ethernet II, Src: AsustekC_8e:72:d (00:13:d4:8e:72:d), Dst: Asiarock_9e:80:dc (00:0b:6a:9e:80:dc)

Internet Protocol, Src: 65.54.171.48 (65.54.171.48), Dst: 10.0.0.14 (10.0.0.14)

Transmission Control Protocol, Src Port: msnp (1863), Dst Port: alias (1187), Seq: 5798, Ack: 639, Len: 410

MSN Messenger Service

[truncated] MSG

MIME-Version: 1.0\r\n

Content-Type: text/plain; charset=UTF-8\r\n

X-MMS-IM-Format: FN=Comic%20Sans%20MS; EF=; CO=0; CS=0; PF=42\r\n

\r\n

BeLki sen yapabilirsin diye sordm :S

Amacı sadece msn görüşmelerini takip etmek olan bir saldırgan wireshark ta messenger ın kullandığı msnms protokolüne göre paketleri filtrelediğinde kurban makinanın yaptığı msn görüşmelerini kolayca ele geçirir.

TCP OTURUMUNA MÜDAHALE ÜZERİNE

Bilindiği gibi bir tcp bağlantısı 3 adımda gerçekleşir.3 yollu el sıkışma adı verilen bu prosedürde bağlantı kurmak isteyen istemci Makine sunucunun ilgili servisine benim sana gönlüm var anlamında syn bayrağı set edilmiş tcp paketi yollar.Sunucu makinada ilgili servis çalışıyorsa ve diğer koşullar sağlanıyorsa bende sana boş değilim manasında syn/ack bayrakları set edilmiş tcp paketi ile cevap verir.Syn/ack bayraklı tcp paketini alan istemci Makine tamam bu iş oldu manasında ack bayraklı tcp paketini sunucu makinaya gönderir ve tcp bağlantısı sağlanmış olur.

2	3.769524	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [SYN] Seq=0 Win=16384 Len=0 MSS=1460
3	3.769696	192.168.1.101	192.168.1.103	TCP	telnet > bridgecontrol [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=146
4	3.769725	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=1 Ack=1 Win=17520 [TCP CHECKSUM INCO

Bağlantılar 3 yollu el sıkışmada kullanılan sekans numaraları üzerinden güvenilirlik kontrolü yaparlar.Bu durum hali hazırda kurulmuş olan oturuma sekans numarasını tahmin edebildiği durumda paket enjeksiyonunu mümkün kılmaktadır.Ağdaki iletişimi takip eden biri trafik içerisinde var olan oturumlar ve sekans numaraları hakkında bilgi sahibi olur.Normalde bir oturumda bir sonraki paketin sekans numarasının tahmini 4 milyonda 1 ihtimaldir.Ama snifflenen bir ağda durum farklıdır.Saldırgan iletişim sırasında bir sonraki sekans numarasını zaten o sırada okuduğu paketler içerisinde çıkartıp hazırladığı sahte paketi sunucuya bağlantıda olduğu makinadan geliyormuş gibi yedirebilir.

Paket boyutunda örnek bir saldırıyı inceleyecek olursak...

Öncelikle wireshark display filter çubuğunda saldırganın ethernet adresine göre filtre ettiğimiz paketler ile sadece saldırgan tarafından yapılan işlemlere bakalım.

Filter: `eth.addr == 00:01:03:87:a8:eb` Expression... Temizle Uygula

No.	Time	Source	Destination	Protocol	Info
507	474.321627	172.16.6.1	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xe50c0387
510	475.833984	172.16.6.1	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0xe60c0387
514	477.349141	172.16.6.1	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xe60c0387
519	497.332534	3com_87:a8:eb	Broadcast	ARP	Who has 192.168.1.101? Tell 192.168.1.100
520	497.332540	WesternD_29:36:e8	3com_87:a8:eb	ARP	192.168.1.101 is at 00:00:c0:29:36:e8
521	497.332686	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
716	497.359406	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...

Frame 716 (91 bytes on wire, 91 bytes captured)

- Ethernet II, Src: 3com_87:a8:eb (00:01:03:87:a8:eb), Dst: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 - Destination: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 - Source: 3com_87:a8:eb (00:01:03:87:a8:eb)
 - Type: IP (0x0800)
- Internet Protocol, Src: 192.168.1.103 (192.168.1.103), Dst: 192.168.1.101 (192.168.1.101)
- Transmission Control Protocol, Src Port: bridgecontrol (1073), Dst Port: telnet (23), Seq: 243, Ack: 1105, Len: 37
- Telnet
 - Data: echo "echo HACKED" >>\$HOME/.profile\n
 - Data: \000

Şekilde görüldüğü üzere 507, 510 ve 514 . paketler ile ağa dahil olan saldırgan dhcp sunucusunun ip havuzundan kendine ip alıyor.

519	497.332534	3com_87:a8:eb	Broadcast	ARP	Who has 192.168.1.101? Tell 192.168.1.100
520	497.332540	WesternD_29:36:e8	3com_87:a8:eb	ARP	192.168.1.101 is at 00:00:c0:29:36:e8
521	497.332686	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
716	497.359406	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
98737	681.331764	3com_87:a8:eb	Broadcast	ARP	Who has 192.168.1.103? Tell 192.168.1.100
98738	681.331780	DellComp_d5:1e:e7	3com_87:a8:eb	ARP	192.168.1.103 is at 00:06:5b:d5:1e:e7
98739	681.331961	192.168.1.101	192.168.1.103	TCP	telnet > warmspotMgmt [FIN, ACK] Seq=846 Ack=270 Win=31744 Len=0
99095	803.726739	192.168.1.101	192.168.1.103	TCP	telnet > rdmshc [RST] Seq=1085 Win=31744 Len=0
99098	808.721846	3com_87:a8:eb	DellComp_d5:1e:e7	ARP	Who has 192.168.1.103? Tell 192.168.1.100
99099	808.721875	DellComp_d5:1e:e7	3com_87:a8:eb	ARP	192.168.1.103 is at 00:06:5b:d5:1e:e7

Frame 519 (60 bytes on wire, 60 bytes captured)

- Ethernet II, Src: 3com_87:a8:eb (00:01:03:87:a8:eb), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Source: 3com_87:a8:eb (00:01:03:87:a8:eb)
 - Type: ARP (0x0806)
 - Trailer: 00000000000000000000000000000000
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (0x0001)
 - Sender MAC address: 3com_87:a8:eb (00:01:03:87:a8:eb)
 - Sender IP address: 192.168.1.100 (192.168.1.100)

519. pakette görüldüğü üzere 192.168.1.100 ip adresine ve 00:01:03:87:a8:eb mac adresine sahip saldırganımız Broadcast ARP request paketi ile ağa 192.168.1.101 adresli makinenin mac adresini soruyor.

716	497.359406	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
98737	681.331764	3com_87:a8:eb	Broadcast	ARP	Who has 192.168.1.103? Tell 192.168.1.100
98738	681.331780	DellComp_d5:1e:e7	3com_87:a8:eb	ARP	192.168.1.103 is at 00:06:5b:d5:1e:e7
98739	681.331961	192.168.1.101	192.168.1.103	TCP	telnet > warmspotMgmt [FIN, ACK] Seq=846 Ack=270 Win=31744 Len=0
99095	803.726739	192.168.1.101	192.168.1.103	TCP	telnet > rdmshc [RST] Seq=1085 Win=31744 Len=0
99098	808.721846	3com_87:a8:eb	DellComp_d5:1e:e7	ARP	Who has 192.168.1.103? Tell 192.168.1.100
99099	808.721875	DellComp_d5:1e:e7	3com_87:a8:eb	ARP	192.168.1.103 is at 00:06:5b:d5:1e:e7

▶ Frame 716 (91 bytes on wire, 91 bytes captured)
 ▼ Ethernet II, Src: 3com_87:a8:eb (00:01:03:87:a8:eb), Dst: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 ▶ Destination: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 ▶ Source: 3com_87:a8:eb (00:01:03:87:a8:eb)
 Type: IP (0x0800)
 ▶ Internet Protocol, Src: 192.168.1.103 (192.168.1.103), Dst: 192.168.1.101 (192.168.1.101)
 ▶ Transmission Control Protocol, Src Port: bridgecontrol (1073), Dst Port: telnet (23), Seq: 243, Ack: 1105, Len: 37
 ▼ Telnet
 Data: echo "echo HACKED" >>\$HOME/.profile\n
 Data: \000

716. pakete baktığımızda 00:01:03:87:a8:eb mac adresine sahip saldırganımız 192.168.1.103 ip adresi ile 192.168.1.101 ip adresli makinaya ölümcül vuruşu yapıyor Data: echo "echo HACKED" >>\$HOME/.profile\n

Şimdi saldırgan ve birinci kurban açısından paket boyutunda olayı gördük peki unuttuğumuz oturumuna konulan ikinci kurbanı ne oldu ona bakalım.

517	497.330102	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
518	497.330631	192.168.1.101	192.168.1.103	TELNET	Telnet Data ...
519	497.332534	3com_87:a8:eb	Broadcast	ARP	Who has 192.168.1.101? Tell 192.168.1.100
520	497.332540	WesternD_29:36:e8	3com_87:a8:eb	ARP	192.168.1.101 is at 00:00:c0:29:36:e8
521	497.332686	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
522	497.344503	192.168.1.101	192.168.1.103	TCP	telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win=32120 Len=0
523	497.344515	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
524	497.344662	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#1] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
525	497.344672	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
526	497.344818	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#2] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
527	497.344827	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
528	497.344972	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#3] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
529	497.344980	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
530	497.345126	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#4] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
531	497.345135	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
532	497.345280	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#5] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
533	497.345286	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM

▶ Frame 517 (55 bytes on wire, 55 bytes captured)
 ▶ Ethernet II, Src: DellComp_d5:1e:e7 (00:06:5b:d5:1e:e7), Dst: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 ▶ Internet Protocol, Src: 192.168.1.103 (192.168.1.103), Dst: 192.168.1.101 (192.168.1.101)
 ▼ Transmission Control Protocol, Src Port: bridgecontrol (1073), Dst Port: telnet (23), Seq: 232, Ack: 1104, Len: 1
 Source port: bridgecontrol (1073)
 Destination port: telnet (23)
 Sequence number: 232 (relative sequence number)
 [Next sequence number: 233 (relative sequence number)]
 Acknowledgement number: 1104 (relative ack number)
 Header length: 20 bytes
 ▶ Flags: 0x18 (PSH, ACK)

Saldırı sırasında olağan trafikteki diğer paketlere bakarsak 517. pakette 192.168.1.103 adresinin asıl sahibi 00:06:5b:d5:1e:e7 adresli kurbanımız telnet sunucusuna son paketini yolluyor.

Destination port: telnet (23)
 Sequence number: 232 (relative sequence number)
 [Next sequence number: 233 (relative sequence number)]
 Acknowledgement number: 1104 (relative ack number)

Paket içerisinde pakete ait sekans numarası 232 ve bir sonraki paketin sekans numarası 233 olarak belirtilmiş. Bir sonraki paketin sekans numarasını alan saldırganımız ne yapıyor..

521	497.332686	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
522	497.344503	192.168.1.101	192.168.1.103	TCP	telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win=32120 Len=0
523	497.344515	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
524	497.344662	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#1] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
525	497.344672	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
526	497.344818	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#2] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
527	497.344827	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
528	497.344972	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#3] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
529	497.344980	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
530	497.345126	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#4] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
531	497.345135	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
532	497.345280	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#5] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
533	497.345280	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM

▶ Frame 521 (64 bytes on wire, 64 bytes captured)
 ▶ Ethernet II, Src: 3com_87:a8:eb (00:01:03:87:a8:eb), Dst: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 ▶ Internet Protocol, Src: 192.168.1.103 (192.168.1.103), Dst: 192.168.1.101 (192.168.1.101)
 ▼ Transmission Control Protocol, Src Port: bridgecontrol (1073), Dst Port: telnet (23), Seq: 233, Ack: 1104, Len: 10
 Source port: bridgecontrol (1073)
 Destination port: telnet (23)
 Sequence number: 233 (relative sequence number)
 [Next sequence number: 243 (relative sequence number)]
 Acknowledgement number: 1104 (relative ack number)

Hikayeden bir paket atarak bir sonraki sekans numarasını değiştiriyor. Mac adresine dikkat bu paket saldırgandan geliyor. Artık sunucunun bir sonraki pakette beklediği sekans numarası 243. Peki bizim 2. kurbanımızda durum ne? Gariban hala son attığı pakete binaen bir sonraki sekans numarasını 233 sanıyor. 96 paket boyunca 233 sekans numaralı paketi yolluyor ve buna karşı sunucu tarafı 96 tane duplicate ack mesajı ile seq uyuşmazlığını bildirmeye çalışıyor. Taki 716. pakette beklediği sekans numarası ile naneyi yiyinceye kadar :))

708	497.358789	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#93] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
709	497.358797	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
710	497.358942	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#94] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
711	497.358950	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
712	497.359096	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#95] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
713	497.359104	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
714	497.359249	192.168.1.101	192.168.1.103	TCP	[TCP Dup ACK 522#96] telnet > bridgecontrol [ACK] Seq=1105 Ack=243 Win
715	497.359257	192.168.1.103	192.168.1.101	TCP	bridgecontrol > telnet [ACK] Seq=233 Ack=1105 Win=16416 [TCP CHECKSUM
716	497.359406	192.168.1.103	192.168.1.101	TELNET	Telnet Data ...
717	497.359408	192.168.1.101	192.168.1.103	TELNET	Telnet Data ...

▶ Frame 716 (91 bytes on wire, 91 bytes captured)
 ▶ Ethernet II, Src: 3com_87:a8:eb (00:01:03:87:a8:eb), Dst: WesternD_29:36:e8 (00:00:c0:29:36:e8)
 ▶ Internet Protocol, Src: 192.168.1.103 (192.168.1.103), Dst: 192.168.1.101 (192.168.1.101)
 ▼ Transmission Control Protocol, Src Port: bridgecontrol (1073), Dst Port: telnet (23), Seq: 243, Ack: 1105, Len: 37
 Source port: bridgecontrol (1073)
 Destination port: telnet (23)
 Sequence number: 243 (relative sequence number)
 [Next sequence number: 280 (relative sequence number)]
 Acknowledgement number: 1105 (relative ack number)
 Header length: 20 bytes
 ▶ Flags: 0x18 (PSH, ACK)
 Window size: 31744
 ▶ Checksum: 0x45e1 [correct]
 ▼ Telnet
 Data: echo "echo HACKED" >>\$HOME/.profile\n
 Data: \000

Wireshark kullanımına alışmanız açısından tcp oturumuna müdahale etmenin paket boyutunda incelemesini gösterdim. Umarım faydalı olur. Wireshark ile ağınızda bu tarz bir saldırıyı farketmeniz açıkçası biraz zor. Çok dikkatli incelemeniz lazım her paketi.

FTP SALDIRI TESPİTİ

Ftp RFC 959 da belirtildiği üzere 7 katmanda çalışan dosya transfer protokolüdür. Örneğimizde ftp sunucusuna yapılan login denemelerini inceleyeceğiz. Her tcp bağlantısında olduğu gibi ftp oturumu sağlanmadan önce tarafların el sıkışma prosedürünü gerçekleştirmesi gerekir. Kayıt

dosyamıza baktığımızda ilk 3 paket HS olayını sağlıyor.

1	0.000000	192.168.0.114	192.168.0.193	TCP	trim > ftp [SYN] Seq=0 Win=16384 Len=0 MSS=1460
2	0.002319	192.168.0.193	192.168.0.114	TCP	ftp > trim [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1452
3	0.002338	192.168.0.114	192.168.0.193	TCP	trim > ftp [ACK] Seq=1 Ack=1 Win=17424 Len=0

El sı sıkma işlemi tamamlandıktan sonra sunucu Hoş geldin mesajı atarak login işlemine hazır olduğunu belirtir. Ardından kullanıcı USER ve PASS ftp request komutlarıyla sunucuya bağlanır. Normal bağlanma işlemi bu şekilde gerçekleşir. Aşağıdaki kayıt dosyasında ise 4. pakette görüldüğü üzere başarısız bir giriş gerçekleştirilmiş.

1	0.000000	10.121.70.151	10.234.125.254	TCP	ftp > gotodevice [ACK] Seq=1 Ack=1 Win=49152 [TCP CHECKSUM INCORRECT]
2	0.007624	10.121.70.151	10.234.125.254	TCP	ftp > di-msg [SYN, ACK] Seq=0 Ack=0 Win=49152 [TCP CHECKSUM INCORRECT]
3	0.007741	10.234.125.254	10.121.70.151	TCP	di-msg > ftp [ACK] Seq=0 Ack=1 Win=17520 [TCP CHECKSUM INCORRECT] Len=
4	0.012755	10.121.70.151	10.234.125.254	FTP	Response: 530 Login incorrect.
5	0.013188	10.121.70.151	10.234.125.254	TCP	ftp > ehome-ms [SYN, ACK] Seq=0 Ack=0 Win=49152 [TCP CHECKSUM INCORRE
6	0.013244	10.234.125.254	10.121.70.151	TCP	ehome-ms > ftp [ACK] Seq=0 Ack=1 Win=17520 [TCP CHECKSUM INCORRECT] L
7	0.019545	10.121.70.151	10.234.125.254	TCP	ftp > EtherNet/IP-1 [ACK] Seq=1 Ack=1 Win=49152 [TCP CHECKSUM INCORRE
8	0.024625	10.234.125.254	10.121.70.151	TCP	gotodevice > ftp [FIN, ACK] Seq=1 Ack=23 Win=17447 [TCP CHECKSUM INCO
9	0.027745	10.121.70.151	10.234.125.254	TCP	ftp > netiq [ACK] Seq=1 Ack=1 Win=49152 [TCP CHECKSUM INCORRECT] Len=
10	0.031952	10.121.70.151	10.234.125.254	FTP	Response: 331 Password required for admin.
11	0.032425	10.234.125.254	10.121.70.151	FTP	Request: PASS merlin.
12	0.033229	10.121.70.151	10.234.125.254	TCP	ftp > rockwell-cspl [ACK] Seq=1 Ack=1 Win=49152 [TCP CHECKSUM INCORRE
13	0.040913	10.121.70.151	10.234.125.254	FTP	Response: 530 Login incorrect.

```
Frame 4 (76 bytes on wire, 76 bytes captured)
Ethernet II, Src: Cisco_3c:3f:a8 (00:01:96:3c:3f:a8), Dst: AmbitMic_aa:af:80 (00:d0:59:aa:af:80)
Internet Protocol, Src: 10.121.70.151 (10.121.70.151), Dst: 10.234.125.254 (10.234.125.254)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: gotodevice (2217), Seq: 1, Ack: 1, Len: 22
File Transfer Protocol (FTP)
  530 Login incorrect.\r\n
    Response code: Not logged in (530)
    Response arg: Login incorrect.
```

Aynı şekilde diğer paketlere baktığımızda çok kısa süre içerisinde birçok başarısız giriş denemesinin yapıldığını görüyoruz.

28	0.104155	10.234.125.254	10.121.70.151	TCP	netiq > ftp [ACK] Seq=2 Ack=24 Win=17447 [TCP CHECKSUM INCORRECT] Len=
29	0.108560	10.121.70.151	10.234.125.254	FTP	Response: 530 Login incorrect.
30	0.108773	10.121.70.151	10.234.125.254	TCP	ftp > rockwell-cspl [ACK] Seq=35 Ack=15 Win=49152 [TCP CHECKSUM INCOR
31	0.112332	10.234.125.254	10.121.70.151	TCP	EtherNet/IP-1 > ftp [FIN, ACK] Seq=14 Ack=57 Win=17447 [TCP CHECKSUM
32	0.120024	10.121.70.151	10.234.125.254	FTP	Response: 530 Login incorrect.
33	0.121851	10.234.125.254	10.121.70.151	TCP	rockwell-cspl > ftp [FIN, ACK] Seq=15 Ack=57 Win=17447 [TCP CHECKSUM
34	0.122830	10.121.70.151	10.234.125.254	TCP	ftp > rockwell-csp2 [ACK] Seq=35 Ack=12 Win=49152 [TCP CHECKSUM INCOR
35	0.141432	10.121.70.151	10.234.125.254	TCP	ftp > EtherNet/IP-1 [ACK] Seq=57 Ack=15 Win=49152 [TCP CHECKSUM INCOR
36	0.141886	10.121.70.151	10.234.125.254	TCP	ftp > EtherNet/IP-1 [FIN, ACK] Seq=57 Ack=15 Win=49152 [TCP CHECKSUM
37	0.141939	10.234.125.254	10.121.70.151	TCP	EtherNet/IP-1 > ftp [ACK] Seq=15 Ack=58 Win=17447 [TCP CHECKSUM INCOR
38	0.145312	10.121.70.151	10.234.125.254	TCP	ftp > rockwell-cspl [ACK] Seq=57 Ack=16 Win=49152 [TCP CHECKSUM INCOR
39	0.145896	10.121.70.151	10.234.125.254	FTP	Response: 530 Login incorrect.
40	0.147244	10.121.70.151	10.234.125.254	TCP	ftp > rockwell-cspl [FIN, ACK] Seq=57 Ack=16 Win=49152 [TCP CHECKSUM

ftp.request.command == "USER" || ftp.request.command == "PASS"

Filtre ifadesi ile ftp oturumunda USER ya da PASS komutlarının geçtiği paketleri kısacası login denemelerini listeleyebiliriz.

Son Sözlər...

Sizinde bildiđiniz gibi sorunlar bitmez. Wireshark 'ı kullanmaya alışmanız açısından basit ama faydalı olduğunu düşündüğüm birkaç örnek verdim.Şimdilik bu kadarı yeterli.Okuduklarınızın kalıcı olması için sizinde bildiđiniz gibi uygulama şart.<http://wiki.wireshark.org/SampleCaptures> adresinde örnek pcap dosyaları var indirip incelemeniz faydanıza olur.Kendiniz kullandıkça pratik çözümleri geliştirebilirsiniz.Makalede bulduğunuz hataları bildirirseniz en kısa zamanda düzeltirim.Yazımı burada bitirirken okuyan herkese teşekkür eder faydalı olmasını dilerim.

Teşekkürler..

Bıkmadan usanmadan sorularıyla uğraşan hocam Ahmet CİHAN 'a (hurby) saygı ve sevgilerimle..

KAYNAKLAR

<http://wiki.wireshark.org/>

Practical Packet Analysis by Chris SANDERS

Wireshark User's Guide: 20996