

**T.C**

**FIRAT ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**ADLİ BİLİŞİM MÜHENDİSLİĞİ**

**AD: MEVLÜD**

**SOYAD: KUYUMCU**

**NU: 175509036**

# TARAMA VE KEŞİF ARAÇLARI:

# HOST KEŞFETME VE TARAMA:

Host kelime anlamı olarak Türkçe ’ye çevirdiğimizde ev sahibi, yer sağlayan, sağlayan anlamlarına gelmektedir.

Host keşfetmek ve Taramak için çeşitli yazılım ve donanımları kullanırız. Bunlardan bazılarından doküman içerisinde detaylı bir şekilde bahsedilmiştir.

# NMAP

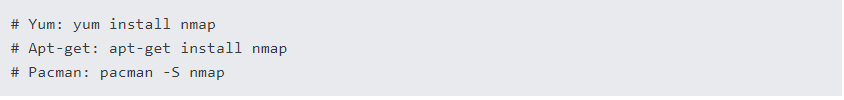
Nmap (Network Mapper), ağ araştırması veya güvenlik denetimi için sıkça kullanılan açık kaynaklı ücretsiz bir programdır. Nmap başlangıçta büyük ağları taramak için tasarlandı, ancak tek ana bilgisayarlar için eşit derecede de iyi çalışabiliyor.

Ağ envanteri, servis yükseltme programlarını yönetme ve ana makine veya servis çalışma süresini izleme gibi görevler için faydalı bir araç olan Nmap, Windows, macOS ve Linux platformlarını destekliyor.

Nmap ağdaki **cihazlara** ve **cihazların portlarına** çeşitli paketler gönderip ve cevaplara bakarak cihazın açık olup olmadığını, açık olan portlar üzerinde hangi servislerin çalıştığını, cihazların hangi işletim sistemini kullandığını öğrenmemize olanak sağlayan open-source bir araçtır. Bu saydığım özelliklerin yanı sıra Nmap’in çok kullanışlı farklı özellikleri de bulunuyor.

# KURULUMU:

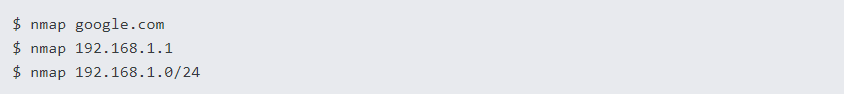
Retrat, Debian ve Arch based linux dağıtımlarının paket depolarında Nmap bulunuyor. Dağıtımın kendi package manager kısmından yükleyebilirsiniz.



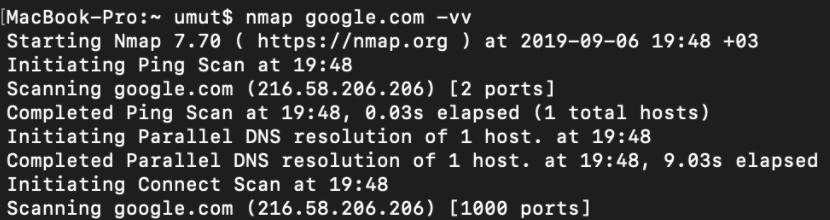
Eğer paket yöneticinizle ilgili bir sorun varsa ya da kendiniz kurmak isterseniz Nmap’in dowloand sayfasından rpm halini indirip kendiniz de kurabilirsiniz. Kullandığınız işletim sistemini seçip yükleme talimatlarını izleyebilirsiniz.

# Hedef Belirtme (Target Specification)

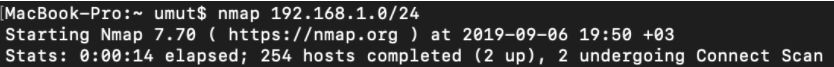
Nmap’i en basit olarak kullanmak için bir hedef belirtmememiz gerekiyor. Hedef olarak bir web adresi, IP adresi verebileceğinizin yanı sıra bir ağ da verebiliriz. Nmap’in kendi sayfasında yazana göre aslında büyük ağları taramak için tasarlanmış fakat tekli hedeflerde de son derece iyi bir sonuç veriyor. Çok uzatmadan bir hedefi nasıl belirteceğimize geçelim.



Hedef belirtmek için argüman olarak vermemiz yeterli. Eğer Nmap’ e sadece hedef verirseniz ve herhangi bir opsiyon belirtmezseniz **Nmap default olarak en çok kullanılan 1000 porta SYN taraması yapar.**

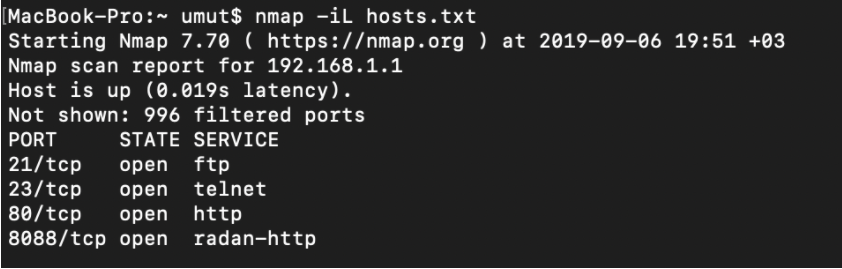


Gördüğünüz gibi Nmap*‘*e sadece hedef verdiğimizde en çok kullanılanilk 1000 porta SYN taraması yaptı *ve* bize açık olan portları ve portlarda çalışan servisleri getirdi. Böyle teker teker bütün networkü taramak zahmetli bir iş. Bunun içinde Nmap le 192.168.1.0/24 subnet ini tek seferde “**nmap 192.168.1.0/24**” şeklinde tarayabiliyoruz.



Bütün subnet i tarattığımızda nmap bize açık olan makineleri ve açık olan makinelerdeki açık portları bize listeliyor.

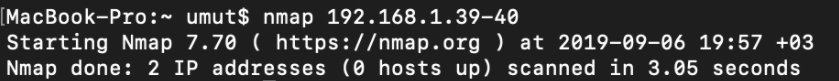
Farzedinki elinizde açık olan hostların IP adreslerinin yazılı olduğu bir txt var. Nmap bunu da düşünmüş ve **-iL** parametresi ile elinizde bulunan hostların yazılı olduğu dosyayı hedef olarak nmap’ e verebiliyorsunuz.



Peki ya bütün ağı tararken taramasını istemediğiniz bir host varsa? Nmap’te çareler tükenmez. “**–exclude**” ve “**–excludefile**” parametreleri ile nmap e taramasını istemediğiniz hedefleri belirtebilirsiniz.



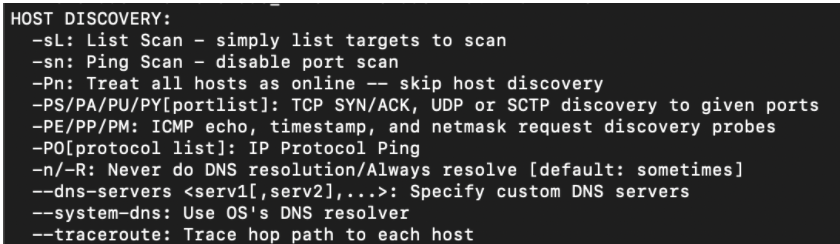
Belirli aralıklardaki IP adreslerinide tarayabilirsiniz.



Yani bu komut sadece **192.168.1.39** ve **192.168.1.40** IP adreslerini tarayacaktır.

# Hedef Keşfi (Host Discovery)

Nmap ile ağda bulunan hostları bulabiliyoruz ve Nmap bunu bize birden fazla seçenek ile yapabilme olanağı sunuyor. Birden fazla seçenek sunmasındaki neden bazı işletim sistemleri bazı tekniklere cevap vermemesidir. Örneğin **güncel Windows işletim sistemleri ping taramasına cevap vermemektedir**, bu yüzden hostun up olup olmadığını bulmak için farklı bir tarama yapmamız gerekiyor.



Nmap bize **ICMP echo request** ile host keşfi yapabilme seçeneği sunarken aynı zamanda **SYN/ACK/UDP** paketlerini belli *portlara* göndererekte host keşfi yapabilmemizi sağlıyor.

Hiç bir parametre belirtmezsek (yani sadece hedef belirtirsek) nmap host keşfi için **ICMP echo request, 443 portuna TCP SYN paketi, 80 portuna TCP ACK paketi ve ICMP timestamp request** yaparak host keşfi yapıyor.

* **-sL**: Herhangi bir paket göndermeden bütün IP adreslerini listeler. Açık olan olmayan bütün adresleri. Daha önce hiç işime yaramadı, belki sizin yarar.
* **-sn**: Bu parametre nmap e host keşfi yaptıktan sonra port taraması yapmamasını söyler. Normalde nmap host u keşfettikten sonra açık olan hostlara port taraması yapar, bu parametre nmap e bunu yapmamasını söyler. Yetkisiz bir kullanıcı tarama yaptığında nmap sadece **80 ve 443 portlarına TCP SYN** paketlerini gönderir. Yetkili bir kullanıcı ile yaptığınızda **ARP istekleri** gönderir.
* **-Pn**: **Host keşfini yapmaz** (bütün hostları açık olarak kabul eder). Port taramasını hızlandırmak için bu seçeneği kullanabilirsiniz. Eğer açık olduğunu bildiğiniz bir host a nse scriptlerini kullanacaksanız **-Pn** ve **-sn** parametrelerini kullanarak işleminizi iyice hızlandırabilirsiniz.
* **-PS**: Belirlenen portlara **TCP SYN** paketleri göndererek host keşfi yapar. Örneğin 22, 80 ve 443 portlarına SYN paketleri ile host discovery yapmak istersek şöyle bir komut kullanmamız gerekirdi : “**nmap -PS 22,80,443 192.168.1.0/24**“
* **-PA**: Bir önceki TCP SYN gibi bu sefer **TCP ACK** paketlerini belli portlara göndererek host discovery yapar. SYN ve ACK paketleri göndermek firewall ları geçme ihtimalinizi arttırır.
* **-PU**: **UDP paket**leri göndererek host discovery yapar. Firewall ları geçme ihtimaliniz artar fakat UDP protokolü gereği paketinizin gittiği garanti olmadığı için hata ihtimali yüksek olur, çok gerekli olmadığı sürece önermem ben.
* **-PE; -PM; -PP**: **ICMP paket**leri göndererek host discovery yapar.
* **-PR**: Tüm ağa **ARP request** gönderir, eğer **ARP response** gelirse hostun up olduğu belli olur ve başka deneme yapması gerekmez.
* **-n:** Nmap bazen *DNS* çözümleme yapar. Biz Nmap ten DNS çözümleme yapmasını engellemek istersek bu parametreyi kullanabiliriz. İşlemimizi hızlandırır.
* **-R:** Bu seferde DNS çözümlemesi yapmasını istersek bu parametreyi kullanabiliriz.

Gördüğünüz gibi sadece host discovery ile ilgili bir sürü seçenek bulunuyor. Nmap’in avantajı gerektiği yerde gerektiği seçeneği kullanarak en doğru ve en kolay yoldan sonuca ulaşmaktır.

Bahsetmediğim bir-iki parametresi daha bulunuyor fakat onlar çokta önemli değil, eğer işiniz düşerse “**nmap –help**” yazarak bakabilirsiniz.

## Port Taraması (Port Scanning)

Nmap‘in bunca özelliğinin içinde \_port taraması\_da mevcuttur. Nmap ilk geliştirilmeye başlandığından itibaren bu özelliğinin bulunmasının yanı sıra artık nmap in ana özelliklerinden biri haline gelmiştir.

Nmap port tarama sonucunu sadece açık/kapalı yerine port tarama sonucunu **6 kategoriye** ayırıyor. Bunlar:

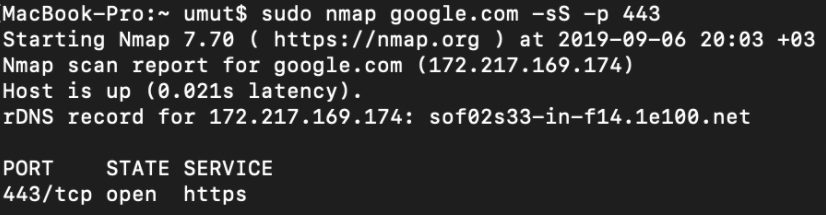
**open**, **closed***,* **filtered**, **unfiltered**, **open*|*filtered** ve **close|filtered**

* **open**: Tarama sonuçlarında open port görmek bize o portta bir servis çalışıyor ve bu port üzerinde bir filtreleme işlemi olmadığını gösteriyor. Sızma testi sırasında filtreli bir port görmek istemeyiz bu yüzden durumu open olan bir port görmek bizim işimize yarar.
* **closed**: Tarama sonucunda close port görmemiz o portun erişilebilir (Yanlış anlaşılmasın erişilebilir olmasından kastım paket gönderip cevap alabildiğimiz için) fakat o port üzerinde hiç bir servis/uygulama çalışmadığı için bir şey yapamayacağımız anlamına gelir. Tabiki Nmap bu özellikten şu şekilde faydalanıyor: ***eğer bir port closed durumundaysa bu o hostun açık olduğu anlamına gelir***, yani nmap bu özelliği *host discovery* kısmında da kullanıyor.
* **filtered**: Bunun anlamı Nmap’in o portun açık olup olmadığını anlayamadığıdır. Örneğin firewall o porta giden/gelen veriyi engelliyor da olabilir ya da host açık değildir ve paketler ulaşmıyordur. Nmap doğrulamak için bir kaç paket daha göndermeyi dener, buda tarama hızını yavaşlatır.
* **unfiltered**: Portun erişilebilir fakat Nmap’in portun açık ya da kapalı olduğunu anlayamadığı zaman verdiği sonuçtur. Bunu çözmek için o porta *SYN/FIN* taraması yapabilirsiniz.
* **open|filtered**: Nmap’in o portun açık ya da filtered olduğunu anlayamadığı zaman verdiği sonuçtur.
* **closed|filtered**: Nmap’in o portun kapalı ya da filtered olduğunu anlayamadığı zaman verdiği sonuçtur.

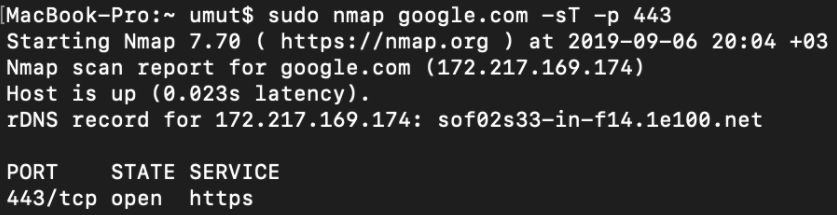
Nmap te bazı taramaları kullanabilmeniz için yetkili bir kullanıcı ile çalıştırmanız gerekiyor, bunun sebebi nmap in raw packet kullanmasıdır.

**Nmap, default olarak en çok kullanılan 1000 porta SYN taraması yapmaktadır.** Siz bütün portlara ya da belirli portlara istediğiniz taramayı yapmak isterseniz Nmap’in bize sunduğu birçok parametreyi kullanabilirsiniz. İlk önce port tarama türlerine bakalım.

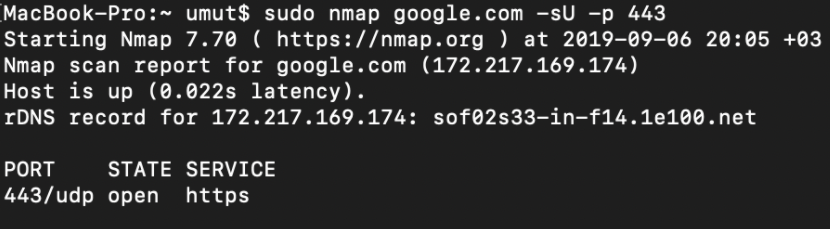
* ***SYN taraması (-sS)*** belirli portlara SYN *flag i set* edilip gönderilerek ve gelen cevaba bakarak yapılan bir taramadır. Sızma testlerinde bu tarama *sıklıkla* kullanılır. Bunun 2 tane sebebi var. İlk olarak SYN taramasında sadece bir paket gönderildiğinden tarama **hızlı** bir şekilde gerçekleşir, ikincisi ise **3 lü el sıkışma tamamlanmadığı için** firewall/IPS gibi yapılar tarafındanengellenmez ve loglardakayıt bırakmaz*.*



* **TCP connect (-sT)** taramasının *SYN* taramasından farkı ise *üçlü el sıkışma tamamlanıyor*. Bunun avantajı üçlü el sıkışma tamamlandığı için o port üzerinde çalışan servis bilgilerini daha detaylı/doğru bir şekilde alabiliyoruz. Dezavantajı ise log larda kayıt bırakıyoruz ve firewall/IPS tarafından engellenme ihtimalimiz artıyor.



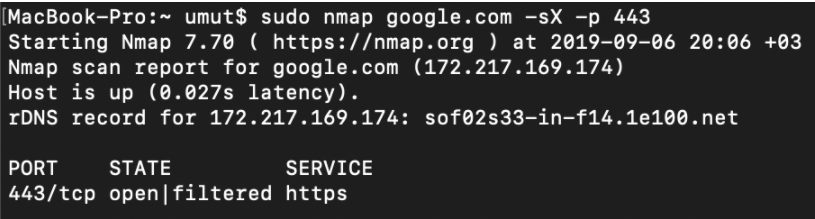
* ***UDP taramasında (-sU)*** nmap TCP yerine UDP protokolünü kullanıyor. Protokol gereğipaketlerin gittiğinin garantisiolmadığı için bu tarama tipi yavaştır*.* Ama UDP portları çok umursanmadığı için bu portlar üzerinde çok güvenlik önlemi alınmaz. Bu da sızma testi uzmanına bir avantaj sağlar. Bu taramada mantık olarak şöyle oluyor: Nmap hedefteki portlara UDP paketleri gönderiyor. Eğer**ICMP port unreachableerror** dönerse nmap **portun kapalı** olduğunu anlıyor. Diğer ICMP unreachable error lardan biri gelirse portu filtered olarak işaretliyor. Geri dönmeyen paket olursa nmap bir daha gönderiyor ve eğer yine geri dönüş olmaz ise open|filtered olarak gösteriliyor. UDP taramasındaki bir diğer önemli olay ise işletim sistemleri ICMP port unreachable mesajını belirli birsayıda gönderiyor buda doğru bir sonuç almamızı engelliyor.



* **TCP NULL/FIN/Xmas (-sN/-sF/-sX)** taramaları TCP protokolündeki flag larideğiştirerek portların open ya da closed olduğunu anlamak için kullanılır. Bu taramaların kullanılmasındaki neden ise flag leri set ettiğinden göze batma olasılığını düşürüyor ve  beklenmedik bir yerden yaptığı için engellenme olasılığıda düşük. Sistemler taranırken SYN/RST/ACK bitlerini içermeyen herhangi bir paket bağlantı noktası kapatıldığında RST ile sonuçlanır, bağlantı noktası açıksa hiç bir yanıt alınmaz.**Null scan**de hiçbir bit setedilmez. (**-sN**)

**FINscan** de sadece FIN bit i set edilir. (**-sF**)

**Xmas scan** de FIN, PSH ve URG flag leri set edilir. (**-sX**)



* **TCP ACK (-sA)** taramasında ise bu sefer ACK flag ini set edip gönderiyoruz. Bu taramayı portun açık ya da kapalı olduğunu anlamak için değilde firewall gibi yapıların olup olmadığını anlamak için kullanıyoruz.
* **–scanflags** parametresiyle istediğimiz flag i set edip gönderebiliyoruz. Nmap bize gerçekten geniş bir tarama seçeneği sunuyor. İhtiyacımız ve doğru olan yöntemi seçmek ise bize kalıyor. Bu parametrenin detaylarına help ten ulaşabilirsiniz.
* **Idle scan (-sI )** sayesinde zombie hostlarınızdan tarama yapabilirsiniz. Gerçek IP adresinizi gizlemek için kullanılabilir.

Nmap varsayılan olarak en çok kullanılan 1000 portu taramaktadır. Kurnaz bir sys admin servislerini kullanılmayan başka portlara taşıdıysa nmap taramasında bunu görmemiz zor olacak. Bununda önüne geçmek için biz elle nmap’in hangi portları taraması gerektiğini belirtebiliyoruz. Yine birkaç parametre ile işimiz kolaylaşıyor.

* **-p** parametresi ile hangi portları tarayacağını elimizle belirtebiliyoruz.

Örnek olarak 1 den 100 e kadar olan portları taramak istersek “**nmap 192.168.1.40 -p 1-100**”

Sadece 22,80 ve 443 portlarını taramak istersek “**nmap 192.168.1.40 -p 22,80,443**“

* **–exclude-ports** parametresi ile taramasını istemediğimiz portları taramadan çıkartabiliyoruz.

1 den 100 e kadar fakat 80. portu taramasını istemezsek “**nmap 192.168.1.40 -p 1-100 –exclude-ports 80**“

* **-F** parametresi ile nmap in default olarak en çok kullanılan 1000 portu taraması yerine en çok kullanılan 100 portu taramasını sağlayabiliriz. Tarama sonucunun hızlı olmasını istiyorsanız kullanılabilir fakat bütün portları taramayacağı için diğer sonuçları göremezsiniz.

“**nmap 192.168.1.40 -F**“

* **–top-ports** parametresi sayesinde en çok kullanılan portlar arasından kaç tanesinin kullanılacağını seçebiliyorsunuz.

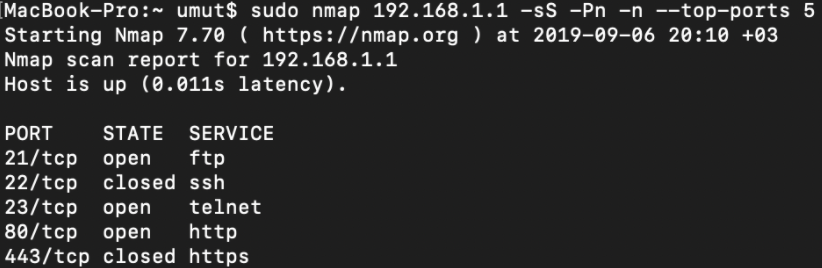
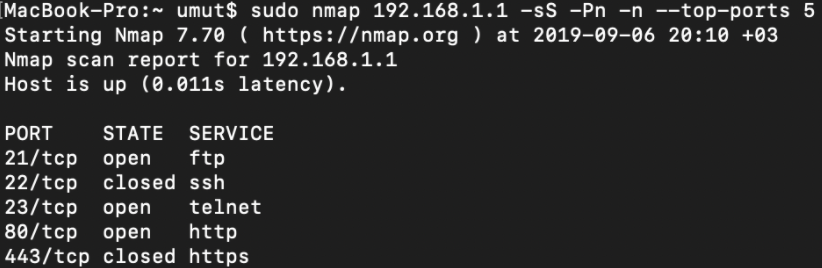
Örnek olarak en çok kullanılan 10 portu taramak isterseniz “**nmap 192.168.1.40 –top-ports 10**“

* **-p-** parametresi ile bütün portları tarayabilirsiniz. Tarama süresi uzayabilir fakat açık olan portu atlamamış olursunuz.

Port belirleme parametreleride bu kadardı. İsterseniz tarama seçenekleriyle port belirleme seçeneklerini karıştırarak örnek yapalım.

Farzedinki ben 192.168.1.1 IP adresindeki hostun up olduğunu biliyorum ve DNS çözümleme yapmasını istemiyorum. Tarama olarakta SYN scan olsun ve en çok kullanılan 5 portu tarayıp bana versin istiyorum. Bunun için **-sS**, **-Pn, -n ve –top**

**-ports** parametrelerini kullanmam gerekiyor.



Sonuçtanda görüldüğü gibi demekki **en çok kullanılan ilk 5 port 21, 22, 23, 80 ve 443** müş. Nmap bizim için bu portlara SYN taraması yaptı ve açık ve kapalı olma durumlarına göre listeledi.

Sizinde bu parametreleri karıştırıp denemenizi öneririm.

## Servis & Versiyon ve İşletim Sistemi Keşfi (Service & Version and OS Detection)

Nmap ile kolay bir şekilde servis ve o servisin versiyonunu keşfedebiliyoruz. Nmap o port üzerinde çalışan servisi **/usr/share/nmap/nmap-services** dosyasından buluyor. Fakat doğru bir sonuç olması için versiyon bilgilerini ve çalışan servisin doğrulamasını o port üzerinden gelen cevaba göre **/usr/share/nmap/nmap-service-probes** database inden karşılaştırarak buluyor. Dosyanın ne kadar büyük olduğunu **wc** komutu ile görebiliyoruz.



-1 parametresi kaç satır olduğunu gösterir.

Gelelim versiyon keşfinin nasıl yapıldığına.

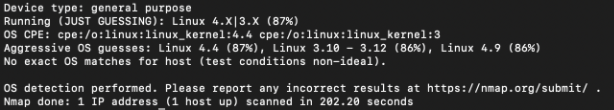
**-sV** parametresi

ile nmap daha önceden bahsettiğim **/usr/share/nmap/nmap-service-probes** database inden gelen cevaba göre karşılaştırıp versiyon bilgisini tahmin etmeye çalışıyor. Nmap dökümantasyonunda **-A** parametresini alternatif olarak kullanabileceğimiz yazılmış fakat -A parametresi, versiyon keşfinin yanı sıra işletim sistemi keşfi+script taraması da yaptığından eğer sadece versiyon taraması yapmak istersek işlemimizi bir hayli uzatacaktır. Burada değinmem gereken en önemli nokta nmap version taraması yaparken 3 lü el sıkışmayı tamamlamak durumunda kalıyor. Bu yüzden firewall lar tarafından engellenme ihtimalimiz artıyor ve bağlantı kurduğumuz için log larda gözüküyor.



Nmap’in en bilindik özelliklerinden biri ise **işletim sistemi keşfidir.** Nmap bunu açık olan portların numaralarına + window size uzunluğuna  ve gelen paketlerdeki fingerprintlere bakarak bulmaya çalışıyor. Bu fingerprintler **/usr/share/nmap/nmap-os-db** database inde bulunuyor ve Nmap dökümantasyonunda yazana göre **2,600 den fazla bilindik işletim sistemi** **fingerprinti** bulunuyor.

*OS discovery’i* **-O** parametresi ile yapabiliyoruz ve eğer nmap işletim sistemini bulamazsa ya da hatalı bir şekilde bulursa verdiği linkten bunu bildirip geliştirilmesinde katkı sağlayabiliyoruz.



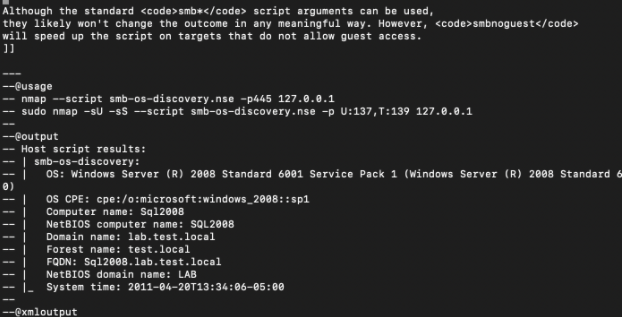
## Nmap Scripting Engine (NSE)

Evet, gelelim Nmap’in bize sağladığı en iyi özelliğe.

Yazının başında Nmap’in zafiyet taraması ve exploit edebilmesinden bahsetmiştim hatırladınız mı? İşte Nmap bunu ***NSE scriptleri*** ile yapıyor. Nmap’in içerisinde şuanlık ***567 adet NSE scripti*** bulunmakta. Bunlar yardımı ile zafiyet taraması, discovery ve exploit işlemlerini yapabilirsiniz. Nmap’in güzel olaylarından birisi ***open-source*** olması. Bunun sayesinde kendimizde NSE scripti yazabiliyoruz. Bu scriptlere ulaşmak isterseniz ***/usr/share/nmap/scripts*** dizininden ulaşabilirsiniz.

Bu scriptlerin ne işe yaradıklarını istediğiniz bir editörle (vim, nano ya da arayüzlü bir editör, tamamen size kalmış) açıp, açıklamaları okuyup ne işe yaradıklarını ve nasıl kullanıldıklarını öğrenebilirsiniz.

Örneğin **os discovery** için kullanılan **smb-os-discovery.nse** scriptine bakalım.



Açıklamaları okuduğumuzda bu scriptin **smb protokolünü** kullanarak **işletim sistemi keşfi** yaptığını öğrendik ve yukarıdaki fotoğraftan görüldüğü gibi nasıl kullanabileceğimizide gösteriyor. İsterseniz elimizi kirletip bir deneyelim.

* Hangi script i kullanacağımızı nmapte **–script** parametresi ile belirtiyoruz.
* Script argümanı olarak vereceklerimizi **–script-args** parametresi ile belirtiyoruz.



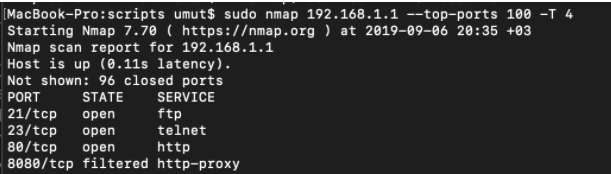
Eğer istediğiniz scripti bulamazsanız kendiniz de yazabilirsiniz.

## İleri Düzey Kullanım (Advanced)

Bunca saydığım özelliklerin yanı sıra sızma testi sırasında işinize yarayacak, bazen hayati bir önem taşıyabilecek bir kaç özellikten daha bahsetmek istiyorum son olarak.

Bazen firewall gibi yapılar ardı ardına gelen paketleri kötü amaçlı gördüğünden engelleyebiliyor. Bunun önüne geçmek için nmap te tarama hızını (-T parametresi ile) ayarlayabiliyoruz. T parametresine 0′ dan 6′ ya kadar numara verebiliyoruz.

* 0 (paranoid): Bu hıza ayarladığımızda paketler arasındaki süre 5 dakikaya kadar çıkar.
* 1 (sneaky): Bu hızla paketler arası süre 15 saniyeye kadar çıkar.
* 2 (polite): Paketler arası süre 0.4 saniye olur.
* 3 (normal): Default olarak kullanılan mod.
* 4 (aggressive): Agresif mod. Hızlı ve güvenilir bir ağda olduğunuzu varsayıp ona göre tahmin yapar.
* 5 (insane): Bu mod doğru sonuç uğruna hızlı bir sonuç almanıza olanak sağlar. Yani mümkün olabilecek en hızlı şekilde size sonuç verir fakat sonuçlar tam olarak doğru olmayabilir.



* **-D** parametresi ile yem verebilirsiniz. Nmap tarama yaparken aynı anda farklı IP adresleri ilede paket atıp, logları karıştırır.
* **-A** parametresi os detection (-O) + version detection (-sV) + script scanning (-sC) ve traceroute (–traceroute) özelliklerinin birleşimi
* **–spoof-mac** parametresi ile mac spoof yapabiliriz.

## Nmap ve Az Bilinen Yönleri:

Nmap, bilgisayar ağları uzmanı Gordon Lyon (Fyodor) tarafından C/C++ ve Python programlama dilleri kullanılarak geliştirilmiş bir güvenlik tarayıcısıdır.

Nmap kullanarak ağa bağlı herhangi bir bilgisayarın işletim sistemi, çalışan fiziksel aygıt tipleri, çalışma süresi, yazılımların hangi servisleri kullandığı, yazılımların sürüm numaraları, bilgisayarın ateşduvarına sahip olup olmadığı, ağ kartının üreticisinin adı gibi bilgiler öğrenilebilmektedir.

10 yılı aşkın süredir açık kaynak kodlu geliştirilen Nmap’e özellikle son birkaç yıldır güvenlik uzmanlarının işlerini kolaylaştıran özellikler ekleniyor.

İşte Nmap’in az bilinen fakat çok işe yarayan özelliklerinden birkaçı.

## En sık kullanılan portlar üzerinde tarama:

Port taramalarında en büyük sorunlardan biri hangi portların taramaya dahil edileceğidir. Bilindiği üzere TCP ve UDP protokollerinin her biri 65535 port olasılığı var. Taramalarda tüm bu portları taramaya dahil edecek olursak tarama zamanı oldukça uzayacaktır. Dahil edilmezse de arada açık olup fakat bizim taramadığımız portlar olabilir. Bu sıkıntıyı aşmak için Nmap yazarı Fyodor geçen senelerde internet üzerinde yaptığı uzun araştırmalar sonucu internete açık portların belli oranını çıkartmış. Bu araştırma ile top 10, top 100, top 1000 gibi portları taratmak mümkün hale gelmiştir.

Taramanın sonuçlarına göre internette en fazla bulunan açık portlar şu şekildedir.

TCP

1. 80  
2. 23  
3. 22  
4. 443  
5. 3389  
6. 445  
7. 139  
8. 21  
9. 135  
10. 25

UDP

1. 137  
2. 161  
3. 1434  
4. 123  
5. 138  
6. 445  
7. 135  
8. 67  
9. 139  
10. 53

**Taramalarda bu özelliği kullanmak için –top-ports 10 ya da –top-ports 1000 parametreleri kullanılabilir.**

C:Documents and SettingMk>nmap 192.168.2.1 –top-ports 10

Starting Nmap 4.76 ( ) at 2021-05-13 15:23 GTB Standard Time  
Interesting ports on RT (192.168.2.1):  
PORT STATE SERVICE  
21/tcp open ftp  
22/tcp open ssh  
23/tcp open telnet  
25/tcp closed smtp  
80/tcp open http  
110/tcp closed pop3  
139/tcp closed netbios-ssn  
443/tcp closed https  
445/tcp closed microsoft-ds  
3389/tcp closed ms-term-serv  
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds

## Tarama sonuçlarının sebepleri :

Nmap taraması yaparken bir porta ait open|closed|Filtered gibi sonuçlar alırız. Bu sonuçların neden olduğunu konusunda detay bilgi için –reason parametresi kullanılabilir. Böylece açık olan portun neden açık olduğu, kapalı olan portun neden kapalı olduğu konusunda bilgimiz olur.

## UDP taramalar için –reason kullanımı :

# nmap -sU -p 52,53 192.168.2.1 –reason

Starting Nmap 4.60 ( ) at 2009-01-17 13:35 GMT  
Interesting ports on RT (192.168.2.1):  
PORT STATE SERVICE REASON  
52/udp closed xns-time port-unreach  
53/udp open domain udp-response  
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

## TCP taramalar için –reason kullanımı:

# nmap -n -p 80,3389 -sS 192.168.2.1 –reason

Starting Nmap 4.60 ( ) at 2009-01-17 13:36 GMT  
Interesting ports on 192.168.2.1:  
PORT STATE SERVICE REASON  
80/tcp open http syn-ack  
3389/tcp closed ms-term-serv reset  
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.170 seconds

**Bu çıktılar tatmin etmediyse daha detaylı çıktı almak için paket\_trace özelliği ve -v parametresi kullanılabilir.**

## Taramalarda detay çıktı alma(nmap -v)

# nmap -n -p 80,3389 -sS 192.168.2.1 -vv

Starting Nmap 4.60 ( ) at 2009-01-17 13:37 GMT  
Initiating ARP Ping Scan at 13:37  
Scanning 192.168.2.1 [1 port]  
Completed ARP Ping Scan at 13:37, 0.01s elapsed (1 total hosts)  
Initiating SYN Stealth Scan at 13:37  
Scanning 192.168.2.1 [2 ports]  
Discovered open port 80/tcp on 192.168.2.1  
Completed SYN Stealth Scan at 13:37, 0.01s elapsed (2 total ports)  
Host 192.168.2.1 appears to be up … good.  
Interesting ports on 192.168.2.1:  
PORT STATE SERVICE  
80/tcp open http  
3389/tcp closed ms-term-serv  
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

Read data files from: /usr/local/share/nmap  
Nmap done: 1 IP address (1 host up) scanned in 0.176 seconds  
Raw packets sent: 3 (130B) | Rcvd: 3 (134B)

## -v kullanmadan yapılan tarama :

home-labs scripts # nmap -n -p 80,3389 -sS 192.168.2.1

Starting Nmap 4.60 ( ) at 2009-01-17 13:37 GMT  
Interesting ports on 192.168.2.1:  
PORT STATE SERVICE  
80/tcp open http  
3389/tcp closed ms-term-serv  
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.169 seconds

**-v parametresinin yeterli olmadığı durumlarda -d[seviye] parametresi ile oldukça detaylı çıktılar alınabilir. Özellikle sonuçlarından şüphelendiğiniz ve sebebini bulamadığınız taramalarda oldukça yardımcı olacaktır.**

home-labs scripts # nmap localhost -p 22 -d9

Starting Nmap 4.60 ( ) at 2009-01-17 13:48 GMT  
Fetchfile found /usr/local/share/nmap/nmap-services  
The max # of sockets we are using is: 0  
————— Timing report —————  
hostgroups: min 1, max 100000  
rtt-timeouts: init 1000, min 100, max 10000  
max-scan-delay: TCP 1000, UDP 1000  
parallelism: min 0, max 0  
max-retries: 10, host-timeout: 0  
———————————————  
mass\_rdns: Using DNS server 192.168.2.1  
Initiating SYN Stealth Scan at 13:48  
Scanning localhost (127.0.0.1) [1 port]  
Pcap filter: dst host 127.0.0.1 and (icmp or (tcp and (src host 127.0.0.1)))  
Packet capture filter (device lo): dst host 127.0.0.1 and (icmp or (tcp and (src host 127.0.0.1)))  
SENT (0.0370s) TCP 127.0.0.1:55455 > 127.0.0.1:22 S ttl=40 iplen=44 seq=2252809853 win=1024  
\*\*TIMING STATS\*\* (0.0370s): IP, probes active/freshportsleft/retry\_stack/outstanding/retranwait/onbench, cwnd/ccthresh/delay, timeout/srtt/rttvar/  
Groupstats (1/1 incomplete): 1/\*/\*/\*/\*/\* 10.00/75/\* 1000000/-1/-1  
127.0.0.1: 1/0/0/1/0/0 10.00/75/0 1000000/-1/-1  
RCVD (0.0380s) TCP 127.0.0.1:55455 > 127.0.0.1:22 S ttl=40 iplen=44 seq=2252809853 win=1024  
Found 127.0.0.1 in incomplete hosts list.  
RCVD (0.0380s) TCP 127.0.0.1:22 > 127.0.0.1:55455 SA ttl=64 iplen=44 seq=644236104 win=32792 ack=2252809854  
Found 127.0.0.1 in incomplete hosts list.  
Discovered open port 22/tcp on 127.0.0.1  
Changing ping technique for 127.0.0.1 to TCP  
Timeout vals: srtt: -1 rttvar: -1 to: 1000000 delta 1352 ==> srtt: 1352 rttvar: 5000 to: 100000  
Timeout vals: srtt: -1 rttvar: -1 to: 1000000 delta 1352 ==> srtt: 1352 rttvar: 5000 to: 100000  
Moving 127.0.0.1 to completed hosts list with 0 outstanding probes.  
Completed SYN Stealth Scan at 13:48, 0.02s elapsed (1 total ports)  
pcap stats: 6 packets received by filter, 0 dropped by kernel.  
Host localhost (127.0.0.1) appears to be up … good.  
Interesting ports on localhost (127.0.0.1):  
PORT STATE SERVICE REASON  
22/tcp open ssh syn-ack  
Final times for host: srtt: 1352 rttvar: 5000 to: 100000

Read from /usr/local/share/nmap: nmap-services.  
Nmap done: 1 IP address (1 host up) scanned in 0.049 seconds  
Raw packets sent: 1 (44B) | Rcvd: 2 (88B)

## -flist

**(Bu parametre ile Nmap sistemde gördüğü arabirimleri ve yönlendirme tanımlarını gösterir.)**

home-labs scripts # nmap -iflist

Starting Nmap 4.60 ( ) at 2009-01-17 13:50 GMT  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INTERFACES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
DEV (SHORT) IP/MASK TYPE UP MAC  
lo (lo) 127.0.0.1/8 loopback up  
eth0 (eth0) 192.168.2.22/24 ethernet up 00:0C:29:29:96:05  
eth1 (eth1) 100.100.100.2/8 ethernet up 00:0C:29:29:96:0F

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ROUTES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
DST/MASK DEV GATEWAY  
192.168.2.0/0 eth0  
100.0.0.0/0 eth1  
127.0.0.0/0 lo  
0.0.0.0/0 eth0 192.168.2.1  
0.0.0.0/0 eth1 100.100.100.1

## –packet\_trace ile tarama için tüm adımların takibi :

**Nmap tarama yaparken gönderdiği ve aldığı tüm paketleri görmek isterseniz –packet\_trace parametresini kullanabilirsiniz. Arada başka bir cihaz yüzünden taramalarınız sağlıklı sonuçlar vermiyorsa bu çıktılarda görülecektir.**

home-labs scripts # nmap -p 80,3389 -sS 192.168.2.1 –packet\_trace

Starting Nmap 4.60 ( ) at 2009-01-17 13:38 GMT  
SENT (0.0340s) ARP who-has 192.168.2.1 tell 192.168.2.22  
RCVD (0.0350s) ARP reply 192.168.2.1 is-at 00:1A:2A:A7:22:5C  
NSOCK (0.0460s) msevent\_new (IOD #1) (EID #8)  
NSOCK (0.0460s) UDP connection requested to 192.168.2.1:53 (IOD #1) EID 8  
NSOCK (0.0460s) msevent\_new (IOD #1) (EID #18)  
NSOCK (0.0460s) Read request from IOD #1 [192.168.2.1:53] (timeout: -1ms) EID 18  
NSOCK (0.0460s) msevent\_new (IOD #1) (EID #27)  
NSOCK (0.0460s) Write request for 42 bytes to IOD #1 EID 27 [192.168.2.1:53]: ………….1.2.168.192.in-addr.arpa…..  
NSOCK (0.0470s) nsock\_loop() started (timeout=500ms). 3 events pending  
NSOCK (0.0470s) wait\_for\_events  
NSOCK (0.0470s) Callback: CONNECT SUCCESS for EID 8 [192.168.2.1:53]  
NSOCK (0.0470s) msevent\_delete (IOD #1) (EID #8)  
NSOCK (0.0470s) Callback: WRITE SUCCESS for EID 27 [192.168.2.1:53]  
NSOCK (0.0470s) msevent\_delete (IOD #1) (EID #27)  
NSOCK (0.0480s) wait\_for\_events  
NSOCK (0.0520s) Callback: READ SUCCESS for EID 18 [192.168.2.1:53] (58 bytes): ………….1.2.168.192.in-addr.arpa………….’….RT.  
NSOCK (0.0520s) msevent\_new (IOD #1) (EID #34)  
NSOCK (0.0520s) Read request from IOD #1 [192.168.2.1:53] (timeout: -1ms) EID 34  
NSOCK (0.0520s) msevent\_delete (IOD #1) (EID #34)  
NSOCK (0.0520s) msevent\_delete (IOD #1) (EID #18)  
SENT (0.0650s) TCP 192.168.2.22:37890 > 192.168.2.1:80 S ttl=40 id=12041 iplen=44 seq=1831862001 win=1024  
SENT (0.0660s) TCP 192.168.2.22:37890 > 192.168.2.1:3389 S ttl=39 iplen=44 seq=1831862001 win=4096  
RCVD (0.0660s) TCP 192.168.2.1:80 > 192.168.2.22:37890 SA ttl=64 iplen=44 seq=303354857 win=5840 ack=1831862002  
RCVD (0.0670s) TCP 192.168.2.1:3389 > 192.168.2.22:37890 RA ttl=255 iplen=40 seq=0 win=0 ack=1831862002  
Interesting ports on RT (192.168.2.1):  
PORT STATE SERVICE  
80/tcp open http  
3389/tcp closed ms-term-serv  
MAC Address: 00:1A:2A:A7:22:5C (Arcadyan Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.182 seconds

**Çıktıdan görüleceği üzere Nmap öncelikle hedef ip adresinin MAC adresini almak için arp istek paketi gönderiyor ve sonrasında -n parametresi kullanılmadığı için hedef  
ip adresinin dns sorgulamasını yapmaya çalışıyor. Sonradan ilgili TCP portlarina SYN bayraklı paketler göndererek bunların cevabını alıyor ve taramayı bitiriyor.**

## Nmap ile Traceroute

**Nmap bir port üzerinde TCP ya da UDP protokolünü kullanarak traceroute yapabilir.**

# nmap -n -P0 –traceroute www.gezginler.net

Starting Nmap 4.60 ( ) at 2009-01-17 13:43 GMT  
Interesting ports on 208.43.98.30:  
Not shown: 1700 closed ports  
PORT STATE SERVICE  
1/tcp open tcpmux  
21/tcp open ftp  
22/tcp open ssh  
25/tcp open smtp  
26/tcp open unknown  
53/tcp open domain  
80/tcp open http  
110/tcp open pop3  
111/tcp open rpcbind  
143/tcp open imap  
443/tcp open https  
465/tcp open smtps  
993/tcp open imaps  
995/tcp open pop3s  
1720/tcp filtered H.323/Q.931

TRACEROUTE (using port 1/tcp)  
HOP RTT ADDRESS  
1 0.94 192.168.2.1  
2 10.51 85.96.186.1  
3 …  
4 815.81 212.156.118.253  
5 10.71 81.212.26.125  
6 17.01 212.156.117.38  
7 28.49 212.156.119.246  
8 77.55 212.73.206.9  
9 78.76 4.68.109.158  
10 87.75 4.69.133.82  
11 80.41 4.69.132.142  
12 106.71 4.69.140.21  
13 169.46 4.69.141.110  
14 167.60 4.69.141.110  
15 174.82 4.69.134.146  
16 166.92 4.68.17.70  
17 166.83 4.79.170.174  
18 167.89 208.43.98.30

Nmap done: 1 IP address (1 host up) scanned in 46.502 seconds

## Metasploit :

Metasploit, en güçlü ve en popüler hack araçlarından biridir. Rapid7 tarafından geliştirilen programın kaynaklarının çoğu www.metasploit.com adresinde bulunabilir. Ticari ve ücretsiz olmak üzere iki farklı sürüme sahip olan Metasploit, temsili saldırılar düzenleyerek, bilgisayar ve ağ sistemlerindeki güvenlik açıklarını bulmanıza yardımcı olur.

Komut istemiyle veya web kullanıcı arabirimiyle kullanılabilen program, Windows, Linux ve macOS de dahil olmak üzere tüm platformları destekliyor.

## Burpsuite :

BurpSuite, web uygulamalarının güvenlik testlerinin yapılması için kullanılan popüler bir platformdur. Bir uygulamanın saldırı yüzeyinin ilk haritalandırılmasından analize, güvenlik açıklarının detaylı şekilde bulunmasından daha pek çok test sürecini destekleyen çeşitli araçlara sahip olan BurpSuite, manuel teknikleri verimli testler için otomasyonla birleştirme konusunda tam kontrol sağlar.

## Angry IP Scanner:

Angry IP Scanner, platformlar arası IP adresi ve port tarayıcısı görevi görür.  Herhangi bir aralıktaki IP adreslerini tarayabilen program, tarama hızını artırmak için çok iş parçacıklı bir yaklaşım kullanır. Burada taranan her IP adresi için ayrı bir tarama iş parçacığı oluşturulur.

Angry IP Scanner, IP adresinin sağlıklı olup olmadığını tespit etmek için ping işlemi yapar ve ana bilgisayar adını belirler. Portları tarayan ve IP-Port liste dosyalarını çıkaran program, IP'ler hakkında her türlü bilgiyi toplar. Ayrıca program ile tüm bilgileri CSV, TXT, HTML, XML formatlarında dışa aktarabilirsiz.

## Jonh The Ripper :

John The Ripper, Unix, macOS, Windows, DOS, BeOS ve OpenVMS için yayın olarak kullanılan hızlı bir şifre kırıcıdır. Ücretsiz ve açık kaynaklı bir program olan John The Ripper, şifre kuvveti kontrolü, belgelerin çevrimiçi taranması ve farklı şifre türleri için destek gibi birçok olanağa sahip.

Başlangıçta Unix işletim sistemi için geliştirilen John Ripper, şimdi 11’i Unix olmak üzere şimdi on beş farklı platformda (DOS, Win32, BeOS ve OpenVMS) çalışıyor. Eğer şifrelerinizin gücünü sınamak istiyorsanız, John The Ripper’a bir göz atmanızda fayda var.

## Wireshark

Wireshark, dünyanın önde gelen ve yaygın olarak kullanılan ücretsiz bir ağ protokolü analizörüdür. Ağınızda neler olup bittiğini mikroskobik düzeyde görmenizi sağlayan program, birçok ticari ve kar amacı gütmeyen kuruluş, devlet kurumu ve eğitim kurumu için fiili bir standarttır.

Zengin bir özellik kümesine sahip olan Wireshark, yüzlerce protokolü derinlemesine incelemenize, ağ çerçevelerinin içeriğini yakalamanıza ve etkileşimli olarak taramanıza, VoIP analizi yapmanıza ve bilgileri XML, PostScript, CSV veya düz metin olarak dışa aktarmanıza imkan sağlıyor.

Linux, Windows ve macOS platformlarında çalışan Wireshark’ın “TShark” adlı GTK + tabanlı bir sürümü de bulunuyor.

## Netsparker :

Kullanıcı dostu bir web uygulama güvenlik tarayıcısı olan Netsparker, yerleşik penetrasyon testleri ve raporlama gibi araçlara sahip. Netsparker, gelişmiş tarama teknolojisi sayesinde binlerce web uygulamasındaki ve servislerindeki zayıflıkları birkaç saat içinde kolayca tespit edebilir.

Kullanımı kolay, gelişmiş bir web güvenlik çözümü olan program, en karmaşık güvenlik açıklarıyla ilgili dahi detaylı rapor sunabiliyor.

## Acunetix :

Acunetix, kötü niyetli davetsiz misafirlerden bir adım önde olmanızı sağlayacak bir web güvenlik açığı tarayıcısıdır. HTML5, JavaScript ve Tek-sayfa uygulamalarını hızlı ve efektif bir şekilde tarayabilen Acunetix, karmaşık, kimliği doğrulanmamış web uygulamalarını denetleyebilir ve çok çeşitli ağ açıkları konusunda uyumluluk ve yönetim raporları sunabilir.

WordPress açıklarının tespit edilmesinde başarılı bir sicile sahip olan program, hızlı ve ölçeklenebilir yüzlerce sayfayı kesintisiz olarak tarar. Kötü amaçlı Siteler Arası Komut Dosyası (XSS) ve SQL Enjeksiyon gibi kritik güvenlik açıklarını yüzde yüz doğruluk oranı ile tespit eden Acunetix, Windows XP / Windows 7 / 8 / 10 platformlarında kullanılabilir.

## Hashcat :

Hashcat, dünyanın en hızlı şifre kurtarma aracıdır. 2015'e kadar özel bir kod tabanına sahipti; ancak şimdi açık kaynaklı yazılım olarak piyasaya sürüldü. Linux, OS ve Windows platformlarında kullanılabilen program, GPU tabanlı bir motora sahip.

Hashcat, Etkileşimli duraklatma / sürdürme,  geri yükleme, otomatik performans ayarlama, Hex-tuz ve hex-charset, aynı sistemdeki karma cihaz türlerinin kullanılması ve yerleşik kıyaslama sistemi ile pek çok özelliği destekliyor.

## Vega :

Vega, web uygulamalarının güvenliğini test etmek için kullanılan ücretsiz ve açık kaynaklı bir web güvenlik tarayıcısı ve web güvenliği test platformudur. SQL Enjeksiyon, Siteler Arası Komut Dosyası (XSS) ve diğer güvenlik açıklarını bulmanıza ve doğrulamanıza yardımcı olan program, Java, GUI tabanlı yazılmıştır ve Linux, OS ve Windows üzerinde çalışır. Vega ayrıca TLS / SSL sunucularınızın güvenliğini artırma konusunda da size çeşitli fırsatlar sunar.

## Nessus :

Nessus, dünyada birçok kullanıcısı bulunan güvenlik zafiyeti tarama programıdır.  Nessus Professional, Nessus Manager, Nessus Home ve Nessus Cloud sürümleri mevcuttur. Fiziksel, sanal ve bulut ortamlarında güvenlik zafiyetlerinin ve zararlı yazılımların tespitini sağlar.

Bilişim altyapılarının güvenlik açısından denetlenmesi ve güvenlik seviyelerinin tespit edilerek açıklıklarının kapatılması gibi çalışmaları kapsayan penetrasyon/sızma testi hizmeti kurumların gerçekleşecek siber saldırılara karşı hazırlıklı ve dayanıklı olmalarını sağlar. Böylelikle penetrasyon testi yapan güvenlikçiler, hacker gibi düşünüp sisteme sızma ve ele geçirme senaryolarını uygulayarak ve saldırganların deneyebileceği tüm yöntemleri deneyerek gerçek bir saldırı ile karşılaşıldığında sistemin açıklık barındıran noktalarının onarılmış ve güvenliği sıkılaştırılmış olmasını sağlamaktadırlar.

Penetrasyon testlerinde lisanslı veya açık-kaynak kodlu araçlar kullanılmakta, otomatize tarama araçlarının yanı sıra kuruma özel manuel testler de uygulanarak mümkün olduğunca tüm zafiyetler tespit edilip düzeltilmeye çalışılmaktadır.

İç ağ penetrasyon testlerinde kullanılan en önemli araçlardan biri olan Nessus, siber suçlular tarafından kullanılan çeşitli saldırı tekniklerini simüle eden kod parçalarından oluşan pluginleri veritabanında barındırır ve tarama yapılacak cihazlarda uygulayarak sahip oldukları açıklıkları ortaya çıkarır.

Nessus, cihazlarda hangi işletim sisteminin çalıştığını, hangi servislerin hangi portlar üzerinde çalıştığını, işletim sistemi ve servislere ait açıklıklar, yazılım bileşenleri ve ağ protokollerinin sahip olduğu zafiyetler ve uyumluluk gereksinimlerinin ne durumda olduğu gibi kriterlere göre hedef cihazların sahip oldukları açıklıklarla ilgili geniş kapsamlı raporlama sunar. Nessus, sistemde tespit ettiği zafiyetleri Birçok farkı kritere göre sıralama ve filtreleme özelliği sunarakzafiyetleri daha iyi anlamlandırmaya imkân tanır.

## Nesus ‘a Ait Bazı Temel Özellikler :

Nessus sayesinde kurumun saldırı düzeyleri küçültülür.

Nessus, yüksek-hızlı varlık tespiti, yapılandırma denetimi, hedef ayrımlama, zararlı yazılım tespiti ve hassas veri tespiti özelliklerinin yanı sıra pek çok başka özellikle birlikte sunulmaktadır.

Geniş ve kapsamlı bir tarama sağlar.

Saldırıları sınıflandırma özelliği ile ölçeklendirilebilir bir yapı sunar.

Düşük maliyet sunar.

Her geçen gün gelişen saldırı teknik ve taktiklere karşı güncel çözümler ve tarama teknikleri sunmaktadır.

Nessus Home aracı ile ücretsiz hizmet sağlamaktadır.

Tek kullanıcılı lisan veren temel ihtiyaçları sağlayan Nessus Proffesional, üst düzey ihtiyaç için Nessus Manager veya Nessus Cloud’a sürümlerini sunmaktadır.

## Ağ Tarama / Analiz Saldırıları:

Aktif hostları, ip adreslerini, aktif hostların açık portlarını, işletim sistemlerini ve sistem mimarisini, hostlarda çalışan servisleri ve aktif hostların zafiyetlerini keşfetmek için yapılır.

## TCP İletim Bayrakları:

URG (Urgent/Acil): Pakette bulunan verileri hemen işler.

PSH (Push/İtme): Arabelleğe alınmış verileri hemen yollar.

FIN (Finish/Bitirme): Daha fazla iletim olmaz.

ACK (Acknowledgement/Alındı): Bir paketin alındığını onaylar.

RST (Reset/Resetleme): Bağlantıyı resetler.

SYN (Synchronize/Senkronize): Hostlar arasında iletim başlatır.

## TCP/IP İletimi

Three-way Handshake (Üç Yönlü El Sıkışma): TCP Oturum Kuruluşu;

* Client: Server 21. portundan seninle konuşmak istiyorum açık mı? SYN, SEQ# 10
* Server: Tamam konuşalım Client, 21. portum açık. SYN + ACK, ACK# 11, SEQ# 142
* PC A: Tamam, teşekkürler Server. ACK, ACK# 143, SEQ# 11
* TCP Oturum Sonlandırma;
* Client: Veri iletimi işim bitti. FIN, SEQ# 50
* Server: Tamam, sonlandırma talebini aldım. ACK, ACK# 51, SEQ# 170
* Server: Tüm gönderilen verileri aldım. FIN, SEQ# 171
* Client: Tamam, teşekkürler Server. ACK, ACK# 172, SEQ# 51

## TCP Bayraklarını Kullanarak İsteğe Uyarlanmış Paket Yaratma:

Colasoft Packet Builder çeşitli saldırılar için ağları denetlemeye olanak sağlar. Saldırganlar da ağdaki güvenlik duvarlarını ve IDS sistemlerini atlatmaya yarayan parçalı paketler yaratmak için kullanabilirler.

## Tarama Yöntemleri

## Aktif Sistem Kontrol Etme:

ICMP Taraması, hosta ICMP ECHO talepleri göndermektir. Host aktifse ICMP ECHO yanıtı döner. Bu tarama aktif cihazları belirlemede ve ICMP’nin güvenlik duvarını geçip geçmediğini anlamada işe yarar. Nmap kullanılır. nmap –sn ip adresi komutu ile yapılır. Ping Sweep (Ping İstilası), birçok hosta ICMP ECHO talepleri göndererek ip adres aralığındaki aktif hostları belirlemede kullanılır. nmap –sn –PE –PA portlar ip adres aralığı komutu ile yapılır.

## Açık Portları Kontrol Etme:

Basit Servis Keşif Protokolü (Simple Service Discovery Protocol/ SSDP) Taraması, ağdaki mevcut tak çalıştır cihazlarını belirlemede UPnP (Universal Plug&Play/Evrensel Tak Çalıştır) ile beraber çalışan ağ protokolüdür. UPnP’deki zafiyetler saldırganın arabellek taşması veya DoS saldırısı başlatmasına izin verebilir. Nmap ile tarama yapılabilir. Kodların sonuna koyulan –reason komutu portun neden açık ya da kapalı olduğunu söyler. Hping2/Hping3 ile yapılabilir.

hping3 -1 ip adresi: ICMP Ping   hping3 –A ip adresi –p port numarası:ACK Port tarama

hping3 -2 ip adresi –p port numarası: UDP Port tarama hping3 –s –p port numarası ip adresiSYN

## Tarama Teknikleri

TCP Connect/Full Open Scan (TCP Bağlantı/Açık Tarama), three-way handshake ile port açık olduğunda tespit eder. Port kapalıysa hedeften RST paketi döner. nmap –sT komutu ile yapılır.

  Stealth Scan/Half-open Scan (Gizli Tarama/Yarı Açık Tarama), yarı açık bağlantı yapan three-way handshake tamamlanmadan önce aniden sunucu ve istemci arasındaki TCP bağlantısına reset atar. Saldırganlar güvenlik duvarı kurallarını, loglama mekanizmasını atlatıp kendilerini gizlemek için bunu kullanırlar. Inverse TCP Flag Scanning (Ters TCP Bayrağı Tarama), saldırganlar bir TCP bayrağıyla (FIN, URG, PSH) veya bayraksız araştırma paketleri (probe packets)  gönderirler, cevap gelmezse port açık demektir, RST dönerse ise kapalı. Xmas Scan (Xmas Taraması) bunun bir çeşididir. Saldırganlar uzaktaki cihaza FIN, URG ve PUSH bayraklarıyla TCP çerçevesi yollarlar.

Portun açık veya kapalı olma durumu Inverse TCP’de anlatıldığı gibi anlaşılır. ACK Flag Probe Scanning (ACK Bayrak Araştırma Taraması), saldırganlar ACK bayrağıyla TCP probe paketlerini uzaktaki cihaza gönderip portun açık mı kapalı mı olduğunu anlamaya çalışan RST paketleriyle alınan başlık (header) bilgisini (TTL ve WINDOW alanı) analiz ederler. Belirli bir porttaki RST paketinin TTL değeri sınır değeri (boundary value) 64’ten azsa veya WINDOW değeri sıfırdan farklıysa port açıktır. ACK Flag Probe’ta saldırganlar rastgele sequence (bölüm) numarası yollarlar cevap yoksa port filtrelidir (filtered) yani güvenlik duvarı vardır, RST yanıtı dönerse port filtresizdir yani güvenlik duvarı yoktur. IDLE/IPID Header Scan (Boş Başlık Taraması), çoğu ağ sunucusu 80 portundaki web sunucular ve 25 portundaki e-posta sunucular gibi TCP portlarını dinler. Uygulama portu dinliyorsa port açık kabul edilir. Portun açık olup olmadığını anlama yollarından biri SYN paketi göndermektir. SYN|ACK paketi geri dönüyorsa port açıktır ve RST dönüyorsa kapalıdır. İstenmeyen SYN|ACK paketi alan bir makine RST ile cevap verir. İstenmeyen RST önemsenmeyecektir. İnternetteki her IP paketi bölümlenme kimliğine (fragment identification/IPID) sahiptir. İşletim sistemi gönderilen her bir paket için IPID’i artırır bu yüzden bir IPID araştırması saldırgana gönderilen paketlerin sayısını verir.

**Adım 1:** Saldırgan zombi olarak seçtiği makinenin IPID numarasını araştırmak için SYN + ACK paketi yollar. SYN + ACK paketi beklemeyen zombi içinde IPID bulunduran RST paketi ile dönüş yapar.

**Adım 2:** Saldırgan hedef makinenin 80 portuna zombinin IP adresiyle SYN paketi yollar. Port açıksa hedef SYN + ACK paketi ile zombiye geri dönüş yapar ve zombi de cevap olarak hedefe RST yollar. Port kapalıysa hedef zombiye RST ile geri dönüş yapar ama zombi hedefe herhangi bir şey yollamaz.

**Adım 3:** Adım 1 tekrarlanır. Sonucunda IPID numarası kontrol edilir eğer 2 artış olmuşsa 80 portunun açık olduğu anlaşılır. UDP Tarama, porta açık olup olmadığı sorulur, cevap yoksa açıktır, ICMP porta erişilemiyor (ICMP Port unreachable) mesajı alınıyorsa port kapalıdır.

## Port Taramaya Karşı Önlemler:

1. Güvenlik duvarı ve IDS kurallarını probeları belirleyip engellemek için yapılandırmak
2. Güvenlik duvarı düzgün şekilde port tarama eylemlerini tespit ediyor mu anlamak için ağdaki hostlara karşı port tarama araçları çalıştırmak
3. Routerlarda ve güvenlik duvarlarında yönlendirme ve filtreleme yapan mekanizmanın belli kaynak portlar veya kaynak yönlendirmesi (routing) yöntemleri kullanılarak atlatılamamasını (bypass) sağlamak
4. Router, IDS, güvenlik duvarı yazılımlarının (firmware) en güncel şekillerinde olmasını sağlamak
5. Ağ kontrol edip güvenlik duvarında istenmeyen portları engelleyen özel kural kullanmak
6. Güvenlik duvarları ve Routerlardaki ICMP mesajlarının hepsini filtrelemek (gelen ICMP mesaj türleri ve giden ICMP tip 3 erişilemez mesajları gibi)
7. Ağ yapılandırmasını ve mevcut portları kontrol etmek için organizasyonun IP adres uzayına karşı ICMP probelarıyla birlikte TCP ve UDP taraması yapmak
8. Taramaya ve kandırmaya (spoofing) karşı olan kuralların yapılandırılmasını sağlamak

## IDS’in Ötesini Tarama:

IDS Atlatma Teknikleri; Bölümlenmiş IP paketleri kullanmak, saldırı başlayıp sunucudan gelen cevaplar dinlenirken IP adres kandırması yapmak (spoofing), mümkünse kaynak yönlendirmesi (source routing) kullanmak, Saldırı başlatmak için Vekil (Proxy) sunuculara veya trojan bulaştırılmış makinelere bağlanmak. IP Bölümleme (IP Fragment) Kullanarak SYN/FIN Tarama; TCP header birkaç pakete bölünür böylece paket filtreleri paketlerin ne yapmak istediğini anlayamaz.

nmap –sS T4 –A –f –v ip adresi komutu ile yapılır.

Başlık Yakalama (Banner Grabbing): İşletim sistemi parmak izi (fingerprinting) olarak da bilinir. Uzak hedef sistemde çalışan işletim sistemini belirlemek için kullanılan bir yöntemdir. Aktif Banner Grabbing, özel bir şekilde hazırlanan paketler uzaktaki işletim sistemine gönderilip yanıtlar kaydedilir. Daha sonra yanıtlar işletim sistemini belirlemek için bir veri tabanıyla karşılaştırılır. Farklı işletim sistemlerindeki yanıtlar TCP/IP yığını (stack) uygulamasından dolayı farklılık gösterir. Pasif Banner Grabbing, Hata mesajları (error messages) uzaktaki hedef sistem tarafından kullanılan SSL aracı, işletim sistemi türü, sunucu türü gibi bilgiler sağlar. Ağ trafiği dinlenerek hedefteki paketler yakalanıp analiz edilir böylece işletim sistemi belirlenebilir. URL uzantısını arama uygulama versiyonunu belirlemede yardımcı olabilir. Örneğin .aspx à IIS sunucu ve Windows platformu.

Netcat’te nc –vv domain adı 80 ve GET / HTTP/1.0 komutlarıyla veya

Telnet ile telnet domain adı 80 ve GET / HTTP/1.0 komutlarıyla yapılabilir.

## Banner Grabbing’e Karşı Önlemler:

1. Banner’ı devre dışı bırakmak ya da değiştirmek (yanlış bannerler göstermek, bilginin açığa çıkmasını sınırlandırmak için ağdaki gereksiz servisleri kapatmak, banner bilgisini devre dışı bırakmak ya da değiştirmek için ServerMask –www.port80software.com- araçlarını kullanmak, “Yeni Sunucu İsmi” girdisi girilen sunucu header’ının banner bilgisini değiştirmek için httpd.conf dosyasındaki direktifi kullanan mod\_headers modülü ile Apache 2.x, alternatif olarak httpd.conf dosyasındaki ServerSignature satırını Server Signature Off olarak değiştirmek)
2. Web sayfalarındaki dosya uzantılarını gizlemek (web teknolojisini maskelemek için dosya uzantılarını gizlemek, sunucuların kimliğini gizlemek için .asp gibi uygulama haritalamalarını .htm veya .foo ile değiştirmek, Apache kullanıcıları mod\_negotiation direktiflerini kullanabilirler, IIS kullanıcıları da dosya uzantılarını yönetmek için PageXchanger gibi araçları kullanabilirler)

**Zafiyet Tarama:** En iyi araç Nessus. OpenVas de bir zafiyet tarama aracıdır.

**Ağ Şemaları Çizme:** Network Topology Manager bunun için kullanılan bir araçtır.

Vekil Sunucular (Proxy) Hazırlama:

Vekil (Proxy) sunucu başka bilgisayarla iletişim kurmada bir aracı olarak hizmet edebilen bir uygulamadır. Kaynak ip adresini saklar, saldırgan Vekil (Proxy)’nin sahte kaynak adresini taklit ederek saldırısının asıl kaynağını maskeler, normalde girmesi yasak olan web sitesi kaynaklarına ve iç ağlara uzaktan erişim sağlar, kullanıcıdan gönderilen taleplerin arasına girip onları üçüncü hedefe iletir bu yüzden kurbanlar sadece Vekil (Proxy) sunucu adresini belirleyebilir, saldırganlar tespit edilmenin önüne geçmek için birden fazla Vekil (Proxy) sunucuyu birbirine bağlar. Vekil (Proxy) Switcher ziyaret edilen web sitelerinde ip adresini saklar. TOR, ağ izlemelerine ve trafik analizlerine karşı gizliliğini korur ve kendini savunmanı sağlar. CyberGhost, çevrimiçi gizlilik, gizlice dolaşma ve sansürlenmiş veya engellenmiş içerik erişimi sağlar, IP’ni gizler ve tercihine göre değiştirir.

Google’da birçok “free proxy server” vardır. Anonymizers, internette gezinirken kullanıcı bilgisayarındaki tüm tanımlı bilgileri temizler. İnternette izlenmenin önüne geçer. Sansürlü interneti atlatmayı sağlar. Tails, aktif bir işletim sistemidir, kullanıcı bilgisayar üzerindeki DVD’den, USB bellekten hafıza kartından başlatabilir. Gizlice internette gezinmeyi ve sansürden kurtulmayı sağlar. Bilgisayarda herhangi bir iz bırakmaz. Dosyaları, mailleri ve anlık mesajlaşmayı şifrelemek (encrypt) için son model şifreleme kullanır. G-Zapper, kullanıcının web eylemlerini (arama anahtar sözcükleri, arama sonuçları, gezilen web siteleri gibi) ve izlemeyi sağlayan bir tanımlayıcıyla (unique identifier) kullanıcı sistemindeki çerezleri (cookie) düzenler. Google çerezlerindeki bilgi mahkemede delil olarak kullanılabilir. Anonymizers için kral araç Ultrasurf’tür (rakibi websense).

IP Adres Sızdırma kaynak IP adresini değiştirmektir böylece saldırı başka birinden geliyormuş gibi görünür. *hping3* –s –p port numarası saldırılacak ip adresi –a spoof edilecek ip adresi komutuyla yapılır. Bu komutun sonuna –rand-source –flood eklenirse bir sürü ip ile ethernet kartı hızı kadar saldırı yapar. Hem spoof hem DDoS olmuş olur.

IP Sızdırma Anlama Teknikleri; Direk TTL Araştırması (Direct TTL Probes), yanıtlamaya yol açıp şüpheli paketteki TTL’i karşılaştıran spoof edilmiş şüpheli paketin hostuna paket gönderilir. Cevaptaki TTL kontrol edilen pakettekiyle aynı değilse spoof edilmiş pakettir. Bu teknik saldırgan kurbandan farklı subnetteyse etkilidir. IP Kimlik Numarası (IP Identification Number), yanıtlamaya yol açıp şüpheli trafikteki IP ID’sini karşılaştıran spoof edilmiş şüpheli trafiğin hostuna probe gönderilir. IP IDleri kontrol edilen pakettekine yakın bir değerde değilse şüpheli trafik spoof edilmiştir. Bu teknik saldırgan aynı subnete olsa bile etkilidir. TCP Akışı Kontrol Yöntemi (TCP Flow Control Method), spoof edilmiş TCP paketleri gönderen saldırganlar hedefin SYN-ACK paketlerini almazlar. Bu yüzden saldırganlar pencere büyüklüğü tıkanmasındaki (congestion window size) değişimlere cevap veremezler. Alınan trafik window size tükendikten sonra devam ediyorsa yüksek ihtimalle paketler istismar (spoof) edilmiştir.

## IP Sızdırma’ya Karşı Önlemler:

1. Tüm ağ trafiğini IPsec, TLS, SSH ve HTTPs gibi şifrelenmiş ağ protokolleriyle şifrelemek
2. Çok katmanlı koruma derinliği sağlayan birçok güvenlik duvarı kullanmak
3. IP tabanlı doğrulamaya bel bağlamamak
4. Sequence numarası spoof etmeye dayanan IP spoofing saldırılarını önlemek için ilk sequence numarasını rastgele kullanmak
5. Giren Filtreleme: İç IP adresinden geliyormuş gibi görünen paketleri filtrelemek için ağın perimetresinde (DMZ) güvenlik duvarları ve routerlar kullanmak
6. Çıkan Filtreleme: Kaynak adres olarak geçersiz yerel IP adresle çıkan tüm paketleri filtrelemek

Sızma Testi Taraması: Sızma testi, yasal olarak ağ istismar girişiminde bulunarak aktif sistemleri belirleyip, açık portları, ortak servisleri, banner grabbing sistemini keşfederek ağın güvenlik durumuna karar verir. Sistem yöneticilerine; kapatılacak kullanılmayan portlar, devre dışı bırakılacak gereksiz servisler, gizlenecek veya özelleştirilecek bannerlar, sorunları giderilecek servis yapılandırma hataları ve ayarlanacak güvenlik duvarı kuralları hakkında bir rapor sunar.

## Ağ analizi sızma testi

Penetrasyon Testi Taraması;

Host keşfi yapılır (Nmap, Angry IP scanner vb.)

Port taraması yapılır (Nmap, Netscan Tools Pro vb.)

Banner grabbing/İşletim sistemi fingerprinting yapılır (Telnet, Netcraft, ID Serve vb.)

Zafiyet taranır (Netsus, SAINT, GFI LANGuard vb.)

Ağ şemaları çizilir (Network Topology Mapper, OpManager vb.)

Vekil (Proxy)ler hazırlanır (Proxy Workbench, Proxifier, Proxy Switcher vb.)

Tüm bulgular dokümante edilir.