İçindekiler

FUNDAMENTALS SHORT NOTES FOR LEARNİNG THE SECURİTY

1.KALİ LİNUX COMMANDS

Aşağıda genellikle kullanılan bazı kali Linux komutlarının ne işe yaradığını belirten bir tablo belirtilmiştir. Bu komutların detaylı kullanımı için "**man [command]** "şeklinde terminalden taratabilirsiniz.

KOMUTLAR	Açıklama
touch	Dosya oluşturma
mkdir	Klasör oluşturma
pwd	Hangi dizinde olduğunu gösterir.
mv	Move a file or directory
cp	Kopyalar ve taşır.
rm	Dosya kaldırmak. Klasör için -R swicth'ini kullan.
sudo	Süper user do.
cat	Okuma
ls	Bulunduğun yerdeki dosyaları liste şeklinde gösterme
if-config	IP bilgileri gösterir. Yerel IP adresini gösterir. Bu IP'yi başkalarıyla paylaşmakta sorun yok. Önemli ve gizli kalması gereken PUBLİC IP.
file	Dosya türünü söyler.

Şimdi de kali Linux içindeki dosyaların ne olduğuna bakacağız ve içinde ne bulunduklarına kısaca değineceğiz.

bin klasörü: Bizim çalıştırdığımız komutlarımız vs. bu klasörde toplanır diyebiliriz basit bir tabirle.

sbin: System binary (daha çok root kullanıcı)

dev: Donanım dosyaları bulunur.

etc: Ayar dosyaları (DNS vs.), Konfigürasyon dosyaları bulunur.

var: Variable (loglar vs.), web sunucuları bulunur.

tmp: Geçici dosyalar burada tutulur.

boot: Açılış dosyasıdır. Bilgisayarımızı her yeniden başlattığımızda bu dosya çalışır diyebiliriz.

srv: Service survers'lar bulunur.

usr: Çalıştırılabilir kullanıcılar, paket yükleyicileri.

lib: Kütüphaneler.

Ayrıca bir paket indireceğiniz zaman "**apt**" komutunu kullanırız. Bu komutun iki farklı yerde kullanımına değineceğiz.

apt update: İndireceği paketlerin içeriğini günceller.

apt upgrade: kendi bilgisayarınızdakileri günceller.

Sırada ise security konusu kapsamında işimize yarayabilecek bazı komutlara değineceğiz.

1.1.wget komutu

Kısaca bahsedecek olursak webden dosya indirmeni sağlar. HTTP, HTTPS ve FTP protokollerini kullanarak indirmeyi sağlar. (Bu konulara ileride Network adı altında inceleyeceğiz) Kullanımı ise şu şekildedir:

wget https://[dosyanın pathi]/filename

şeklinde bir formatı vardır. Örnek bir gösterim yaparsak;

wget https://wget.wget/wget1/wget2/wget3/wget.txt

şeklinde gösterilebilir. Bu komutla ilgili daha ayrıntılı bilgi öğrenmek için araştırma yapmanızı öneririm.

1.2.scp komutu

Başka bilgisayardan veya kendi bilgisayarından başka bilgisayara dosya kopyalama. Kullanımının iki farklı yolu vardır.

Başka bilgisayara dosya gönderme:

scp [göndereceğin dosya] [user]@[IP adress]://[dosya path'i] filenames

Başka bilgisayardan dosya alma:

scp [user]@[IP adress]://[dosya path'i] filenames

1.3.ps komutu

Linuxta o an çalışan işlemler (process) listelenir. Gözüken listede işlemlerin özelliklerini belirten bölümler vardır. Bunlar:

PID: Benzersiz bir işlem kimliği (İşlem numarası gibi düşünebilirsin)

TTY: Bu kullanıcının giriş yapmış olduğu terminaldir.

TIME: İşlemin çalıştığı süreyi dakika ve saniye cinsiden veren süre

CMD: Süreci başlatan komut

O an çalışan işlemlerin daha detaylı bilgileri için şunları kullanabilirsin:

ps -A: Çalışan bütün işlemleri gösterir.

ps -aux: Daha detaylı gösterir(İşlemin nerede olduğu vs.)

ps -L 2549: 2549 numaralı işlem hakkında bilgi verir.

1.4.top komutu

Anlık olarak sunucumuzun durumunu izleme olanağı sağlar. İşletim sistemi üzerinde çalışan işlemleri ve kullanıcıların işlemlerini izlemelerine izin veren, varsayılan olarak bulunan bir uygulama

Sadece proses(işlem) değil birçok detayda bulunur. Bu detayda toplam kaç tane görev olduğu, kaç tanesinin çalıştığı, kaç tanesinin uykuda olduğunu, kaç tanesinin zombi veya uyuyan olduğu detaylarını verir diyebiliriz.

1.5.kill komutu

Cevap vermeyen, durmuş ve duraksamış dosyalar için kullanılır. Windows'taki görev yöneticisi gibi düşünebilirsin. İşlemlere sinyal gönderir ve onları kapatır. Kill komutunun sinyal numaralı vardır. Eğer hiçbir şey belirtilmemişse default olarak SIGTERM (15) sinyalini kullanır (Bu sinyal değerlerini araştırabilirsin bunu terminal üzerinden "kill -l" ile de bulabilirsin.).

Bir örnek verecek olursak mesela bir prosesi kill ile kapatmak istiyoruz ama onun PID değerini bilmiyoruz. Bunun için "**ps-aux**" komutunu kullanıp prosesin PID değerini öğrendikten sonra mesela bu 1234 olsun, "**kill -9 1234**" diye prosesi kapatabiliriz. Burada 9 zorla öldürmek anlamına geliyor diyebiliriz. Birden çok proses kapatmak için ise 1234 yazan yerin yanına proseslerin PID değerlerini yazarak onları da kapatabilirsin.

1.6. systemctl komutu ve systemd

Systemd, Linux'ta bulunur ve amacı, bilgisayardaki sistem ve servislerin çalışmasını organize etmektir. Sistemdeki diğer işlemleri başlatan, yöneten ve sonlandıran; günlükler, dosya sistemi durumu vb. hakkında bilgi sağlamanın yanı sıra özetle "ana" işlemdir diyebiliriz.

"systemctl" komutu ise systemd servislerini yönetmek, durdurmak, başlatmak ve durumlarını kontrol etmek gibi işlemlerin yapılmasını sağlayan komuttur.

Örnek verecek olursak systemctl komutunu şu şekilde kullanabiliriz.

systemctl start apache2 → apache2'yi başlat
systemctl stop apache2 → durdur
systemctl disable apache2 → devre dışı bırak
systemctl enable apache2 → devreye al

şeklinde kullanabiliriz.(apache2 Kali Linux'ta var olan bir hizmettir.)

NOT

→ Linux'ta bir komut çalıştırdığında belki de yanlış bir şey yaptığınızda **^Z** (Control + Z) ile durduğumuzu zannediyoruz ama sadece pause etmiş oluyoruz. Bunu anlamak için ^Z ile pause ettikten sonra **"fg"** komutunu yazıp çalıştırırsak en son çalışan işlemi yani devam eden işlemi bize gösterecektir. Tamamen durdurmak için **^C** yapmalıyız.
→**"ps aux | less"** komutu çalışan tüm işlemleri tane tane döker.

1.7. crontab komutu ve cron

İşletim sistemi Unix'te cron, süreçleri veya komut dosyalarını düzenli aralıklarla (her dakika, her gün, ay vs.) çalıştıran normal bir arka plan işlem yöneticisidir. Yürütülmesi gereken işlemler ve yürütülmeleri gereken saat ise crontab dosyasında bulunur.

"crontab" komutu, ayarlanan zaman veya zaman diliminde belirtilen komut, script ya da uygulamanın çalışmasını sağlar diyebiliriz.

Temelde işlemin gerçekleştirileceği zaman ve tekrar belirtilip işlemin kendisi tanımlanarak crontab dosyasına eklenmesi ile gerçekleşir.

Crontab aslında 6 yıldızdan oluşur diyebiliriz:

- MIN : Hangi dakikada yürütülür. *
- HOUR : Hangi saatte yürütülür. *
- DOM : Ayın hangi gününde *
- MON : Yılın hangi ayında *
- DOW : Haftanın hangi gününde *
- CMD : Yürütülecek Gerçek Komut *

-00 12,15 * * [komut veya script] = her saat [00 başlangıç saati]-öğlen 12''de akşam 15'te-*-Her gün-*-Her ay-*-Haftanın her günü

* * * * * → yerine genel zaman ifadeleri kullanılarak da pratik şekilde zaman tanımlamaları yapabiliriz.

@reboot: bir defa ve başlangıçta
@yearly ya da **@annually:** Yılda bir defa (0 0 1 1 *)
@monthly: ayda bir defa (0 0 1 * *)
@weekly: haftada bir defa (0 0 * * 0)
@daily ya da **@mignight:** Günde bir defa (0 0 * * *)
@hourly: saatte bir defa(0 * * * *)

1.8. apt komutu

-Ubuntu'da paket yüklemeni sağlayan komut.

-APT deposu, APT araçları tarafından okunabilen deb paketleri ve meta veri dosyalarını içeren bir ağ sunucusu veya yerel bir dizindir.

-Apt dosyaları "/etc/apt/source.list" ya da "/etc/apt/sources.list.d/" içerisinde bulunur.

-Ayrıca **"dpkg"** de paket yükleyici olarak kullanılabilir. Ama **"apt"** komutunun faydası sisteminizi her güncellediğiniz zaman, eklediğimiz yazılım parçalarını içeren deponun da güncellemeler için kontrol edilmesi anlamına gelir.

-Yazılım eklerken, indirdiğiniz şeyin bütünlüğü GPG (Gnu Privacy Guard) anahtarlarının kullanılmasıyla garanti edilir.

NOT

→ **"less"** komutu bize komut satırında bir dosyanın içeriğini görmemizi saplar.

NOT

→ Sistemde meydana gelen hata, sorun, işlemler, değişiklikler ve neredeyse her şey kayıt altına alınıp saklanır. Kayıt altına alınan bilgilere de **"log"** denir. Log tutmak zorundayız çünkü sistemde oluşan herhangi bir hata, hasar veya aksaklığın belirlenmesi için log dosyalarına bakılır. Log dosyaları /var/log 'da bulunur.

2.INTRODUCTİNG NETWORKİNG

2.1.The OSİ (Open Systems Interconnection)

Bilgisayar ağının arka planındaki teoriyi anlatan standartlaştırılmış bir model olan OSİ 7 katmandan oluşur.

Katman 7	**Application**
Katman 6	**Presentation**
Katman 5	**Session**
Katman 4	**Transport**
Katman 3	**Network**
Katman 2	**Data Link**
Katman 1	**Physical**

Her bir katmanın görevi bir üst katmana servis sağlamaktır ama application katmanı hariç. Yukardan aşağı doğru gider veri local bilgisayarımızda ama hedef bilgisayarda veri alt katmandan üst katmana doğru gider.

Veri iletim şekli:

1. Veri halinde alınan bilgi, transport katmanında segment parçalara (birimlere)ayrılır. Bu sayede veriyi alan makinede tekrar bir araya getirilirken doğru sıralanmış olur.
2. Network katmanına segment olarak gelen veriye burada adres bilgileri eklenir ve adres bilgilerinin eklenmesiyle segmentler paketlere dönüşür.
3. Data Link katmanında paketlere MAC adresleri eklenerek frame yapı oluşur.
4. Physical katmanına gelen frameler burada bir bit dizisine dönüşüp iletime hazır hale gelmiş olur.

APPLİCATİON LAYER (KATMAN)

Uygulamaların ağ üzerinde çalışmasını sağlar. Ağ servisini sağlayan program. HTTP, DNS protokolleri ve tarayıcılar bu katmanda. E-posta ve veri tabanı gibi uygulamalarda bu katmanda. Bu katman kullanıcıların gereksinimini sağlar. Sadece bu katman servis sağlamaz.

PRESENTATİON LAYER (KATMAN)

Verinin karşı bilgisayarda anlaşılabilecek şekilde çevrilmesini sağlar. Bunun anlamı farklı programlar birbirlerinin verisini kullanabiliyor diyebiliriz. Verinin formatı belirlenir. Verinin sıkıştırılması, şifrelenmesi, açılması bu katmanda yapılır.

SESSİON LAYER (KATMAN)

Bu katman veri üst katmandan geldiğinde karşı bilgisayarla ağ üzerinden bağlantı kurulup kurulamayacağını kontrol eder. Eğer bağlantı kullanılamaz ise üst katmana bir error hatası bildirimi gönderir ve veri aktarma süreci sona erer. Bağlantı sağlanabiliyorsa, local bilgisayar birden çok bilgisayarla iletişim halinde olduğunda gerektiği zaman doğru bilgisayarla iletişim kurmayı sağlar. Bu veriler karışmadan aynı anda farklı uç noktalara birden fazla istekte bulunmanıza izin veren şey. (web tarayıcınızda aynı anda iki tab açtığınızı düşünün.)

TRANSPORT LAYER

Üstündeki katmanlara taşıma servisi sağlarken ayrıca ağın da servis kalitesini artırır. Bu katmanın ilk amacı verilerin iletileceği protokolü seçmek. En yaygın 2 protokol TCP (Transmission Contal Protocol) ve UDP (User Datagram Protocol) 'dir.

TCP → Bağlantı tabanlı bir protokol olan TCP verilerin uygun bir hızda gönderilmesini ve kaybolan verilerin gönderilmesini sağlamak için iki bilgisayarın sürekli iletişimde kalmasını sağlar. Bağlantı tabanlı olduğu için güvenli bir protokoldür.

UDP → Eğer doğruluğu hıza edersen TCP, ama hızı tercih edersen UDP kullanabilirsin. Veri paketlerini karşı bilgisayara atar ve orada çalışması umut edilir. Genelde video paylaşımı için ideal bir protokoldür.

Transport katmanında bir protokol seçilip ona göre veriler küçük birimlere ayrılır.

NOT

→ UDP' de küçük birimleri **"datagram"** denirken TCP' de **"segment"** denir.

NETWORK LAYER

Bir isteğinizin hedefini bulmaktan sorumludur diyebiliriz. Yani bir web sayfasından bilgi talep etmek istediğinizde, o sayfanın IP adresini alan ve izlenecek en iyi yolu belirleyen katmandır. Bu katmanda verinin router vasıtasıyla yönlendirilmesi sağlanır. Ayrıca burada mesajlar adreslenirken mantıksal adresler (yani IP adresleri) fiziksel adreslere dönüştürülür. IP protokolü de bu katmanda çalışır.

Data Link Layer

Ağ üzerindeki diğer bilgisayarları tanımlama, kablonun o anda kimin tarafından kullanıldığının tespiti ve fiziksel katmandan gelen verinin hatalara karşı kontrolü gerçekleşir bu katmanda. Data Link katmanının iki alt bölümü vardır.

MAC→ Media Access Control. Gelen veriyi hata kodu ve MAC adresi ile paketleyip pyhsical katmana gönderir.

LLC → Logical Link Control. Burası network ile pyhsical katmanı arasında köprü görevi görüyor diyebiliriz. Veri karşı makinede LLC' ye gider ve burada mantıksal portlar oluşturulur. Böylece kaynak ve hedef makinelerde aynı protokoller iletişime geçebilir. (TCP/IP <->TCP/IP). Ayrıca veri paketlerinden bozuk gidenlerin tekrar gönderilmesinden sorumludur. "Flow Control" denilen yani alıcının işleyebileceğinden fazla veri paketi göndererek boğulmasını engellemek LLC' nin görevidir.

Pyhsical Layer

Verinin karşı bilgisayara nasıl gideceğini tanımlar. Karşı bilgisayara veriler bit olarak iletilirken bu katman bir ve sıfırların nasıl elektrik, ışık veya radyo sinyallerine çevrilebileceğini ve aktarılacağını tanımlar.

NOT

→Local bilgisayarda gerçekleşen tüm bu süreçlere **"encapsulation"** denirken karşı bilgisayarda gerçekleşen tersi işlemlere de **"de-encapsulation"** denir.

2.2.TCP/IP

Bir haberleşme protokolü olan TCP/IP, üst katmanı olan TCP' de verinin iletimden önce paketlere ayrılması ve karşı tarafta bu paketlerin yeniden düzgün olarak birleştirilmesi gerçekleşir. Alt katmanı olan IP' de ise iletilen paketlerin istenilen ağ adresine yönlendirilmesi kontrol edilir. TCP/IP protokolü 5 katmandan oluşur. Bunlar:

Application → farklı sunucular üzerindeki süreç ve uygulamalar arasındaki iletişimi sağlayan katmandır.

Transport → Bir noktadan bir noktaya veri akışı sağlayan katmandır.

Internet → Routerların birbirine bağlanmış ağlar boyunca verinin kaynaktan hedefe gitmesini sağlayan katmandır.

Data Link → Uç sistem ile alt ağ arasındaki lojik arabirime ilişkin katmandır.

Physical → İletişim ortamının karakteristik özelliklerini, sinyalleşme hızını ve kodlama şemasını belirleyen katmandır.

NOT

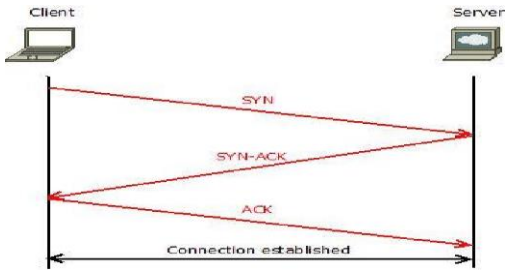
→ Bazı kaynaklarda Data Link ve Physical katmanları birleştirilip "**Network Interface**" adı altında birleştirilmiş olarak verilir.

TCP/IP iletişim protokolleri bütünüdür. TCP/IP verinin biçimlendirilmiş şekilde nasıl olması gerektiğini belirterek uçtan uca bir bağlantı sağlar. TCP/IP modeli ve ilgili protokoller "Internet Engineering Task Force (IETF)" tarafından korunur.

TCP için iki bilgisayar arasında bağlantı olması gerekirken bu işlem "**three way hanshake**" olarak adlandırılır. Süreç ise şöyle gerçekleşir:

1. Bilgisayarımız bağlantı kurmak istediğinde, remote servera bağlantı kurmak istediğini belirten bir istek gönderir. Bu gönderdiği ise Bir "**SYN**" bitidir.
2. Sunucu ise bu isteğe yanıt olarak **SYN** biti ile bir onay biti olan "**ACK** (Acknowledgement)" ile gönderecektir.
3. Sonra bilgisayarımız aldığı yanıttan sonra bağlantının başarıyla kurulduğunu onaylayarak bir **ACK** biti içeren paket gönderecektir.
4. Sonunda da bağlantı böylece kurulmuş olacaktır.

İşte tüm bu aşamaların hepsine **three way** hadshake denir.



OSİ	TCP/I
Application	**Application**
Presentation	
Session	
Transport	**Transport**
Network	**Internet**
Data Link	**Network Interface**
Physical	

Şimdi ise hedef sunucu hakkında bilgi toplamamızı sağlayabilecek bazı komutları göreceğiz.

2.3.ping komutu

Uzak kaynağa bağlantının mümkün olup olmadığını gösterir. ICMP protokolünü kullanır (TCP/IP modellerinden biri). OSİ Modeli' nin Network katmanında çalışır. En büyük avantajı olarak ise çoğu işletim sisteminin desteklediği bir komuttur. Bu komutun daha detaylı bilgileri için "**man ping**" diye aratabiliriz. Ping komutu sayesinde hedefin IP adresini de öğrenebilme imkânı sağlarız. Örnek kullanımı ise şöyledir:

ping google.com

2.4.traceroute komutu

İnternette her ağ birbirine bağlı olduğundan siz bir ağa istek attığınızda o sunucuya varana kadar bir sürü başka sunucudan geçersiniz. Bu isteğin yol haritasına ulaşmak isterseniz "**traceroute**" komutu işinize yarayacaktır. Kullanımı ise şöyledir:

traceroute google.com

2.5.whois komutu

Alan adının kimin olduğunu gösteren komuttur. Bu komut çoğu Kali Linux'ta yüklü gelirken eğer yoksa söyle yükleyebilirsiniz:

sudo apt update && apt-get install whois

Kullanımı ise:

whois google.com şeklindedir.

2.6.dig komutu

Bir web tarayıcınızdan bir sitesine ziyaret ettiğinizde oluşan işlemler çok hızlı ve otomatik yapılır. İşte "dig" komutu bunu manuel yapar. Syntaxi ise şöyle:

dig <domain> @<dns-server-ip>

Özetle diyebiliriz ki bu bilgi bize bir sorgu gönderdiğimizi ve başarıyla bir tam yanıt alıp almadığımızı söyler. Beklendiği gibi sorguladığımız domainin IP adresini verir. Ayrıca bu komut bize TTL (Time The Live) yani yaşama süresini de verir.

NOT

→ Bir domain aradığınızda bilgisayarınızın sorgulayacağı DNS türü "**Recursive**" dir.

→ Bilgisayarınızın bir domain adresini bulmak için bakacağı ilk yer "**Local Cache**" (Yerel Önbellek) dir.

→ **TLD**: URL' lerde noktadan sonra gelen kısma verilen isimdir. Top-Level-Domain

3.NMAP

Bir bilgisayar uzak sunucularla olan tüm iletişimleri için rastgele portlar açar. Her bilgisayarda toplam 65535 port vardır ama örneğin; HTTTP neredeyse sunucunun çoğunlukla 80 numaralı portunda bulunur. HTTPS ise 443, Windows NETBİOS 139 veya SNP 445 numaralı portlarda bulunurlar genellikle.

Bir sunucunun portlardan hangisini açtığını bilmiyorsak hedefe başarılı bir şekilde saldırma şansımız yoktur. Bu nedenle herhangi bir saldırıdan önce port taraması yapmamız gerekir. Bunu yapmanın çeşitli yolları vardır. Mesela bunlardan biri "NMAP" tool idir.

Nmap birçok farklı türde port taraması için kullanılabilir. Nmap, sırayla hedefimizin portlarına bağlanıp portun nasıl yanıt verdiğine göre portun açık, kapalı veya filtrelenmiş şekilde belirler.

Portların durumu sorgulandıktan sonra hangi portların açık olduğu belirlenince manuel ya da nmap kullanarak her portta hangi servislerin çalıştığını numaralandırmaya bakabiliriz.

Bunlar şöyledir:

1. Tcp Connect Scan için: **-sT**
2. Syn Scan taraması için: **-sS**
3. Hedefin hangi işletim sisteminde çalıştığını tespit etmek istiyorsan: **-O**

4. Nmap hedef üzerinde çalışan servislerin sürümünü öğrenmek için switch sağlar. Bu da: **-sV**
5. Nmap için çıkan otomatik çıktının detayını artırmak istiyorsak: **-v**
6. Nmap de seviyeler vardır. Seviye 1 ayrıntısı iyidir ama daha da ayrıntı öğrenmek istiyorsak: **-vv**
7. Nmap sonucu çıkan verileri kaydetmemiz bize daha sonrası için kolaylık sağlayabilir. Bunun için ise 3 farklı format var:
    i. 3 ana formatta kaydetmek için: **-oA**
    ii. Normal formatta kaydetmek için: **-oN**
    iii. grepable formatta kaydetmek için: **-oG**
8. Bazen nmap ile çıkan sonuçlar yeterli olmayabiliyor.
9. Çok gürültüyü önemsemiyorsak agresif moda geçebiliriz. Bu bize service detection, operating system detection, a traceroute ve common script scanning (ortak komut taraması) etkinleştiren kısayoldur: **-A**
10. Eğer nmap hızını artırmak istersen **-T5** anahtarını kullanabilirsin. Ama unutma, çok fazla hız çok fazla gürültü ve hata olabilir demektir.
11. Ayrıca nmap aracında hangi portların taranacağını da seçebiliriz (Mesela port 80):
    i. **-p 80:** sadece 80 numaralı portu tara
    ii. **-p 1000-1050:** 1000-1050 arasını tara
    iii. **-p-** : tüm portları tara
12. Nmap komut dosyası kitaplığında bir komut dosyasını aktif etmek için: **--script**
13. Bir kategorideki scriptleri denemek için mesela "Vuln": **--script=vuln**

Switchlerini kullanabiliriz. Nmap toolunun kullanımına yönelik basit bir örnek vermek gerekirse:

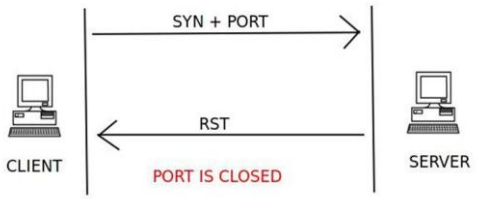
nmap -sT -vv -T4 -p 1000-1050 [target-IP]

diyebiliriz.

3.1. TCP Connect Scan

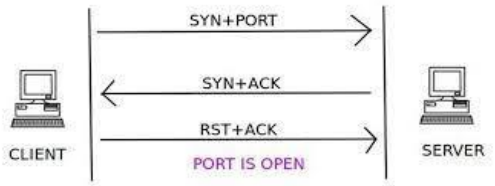
"TCP Connect Scan", sırayla her bir hedef port için three way handshake gerçekleştirerek çalışır. Nmap, belirtilen her bir TCP portuna bağlanmaya çalışır ve aldığı yanıtla hizmetin açık olup olmadığını belirler.

a. Eğer port kapalıysa hedef sunucu geri bildirim yolluyor. Yani şöyle:



b. Eğer port açık ise nmap bir "SYN" paketi yollar. Hedef sunucu ise geri "TCP" paketi ile birlikte "SYNACK" gönderir. Nmap ise en son TCP paket ile ACK geri gönderir.

i.



c. Eğer port açık ama güvenlik duvarının arkasında gizlenmişse;
   i. Güvenlik duvarları gelen paketleri basitçe düşürmek için yapılandırabilir. Yani şöyle;
      a. Nmap bir TCP/SYN istek istek gönderir ama geri cevap alamaz. Bunun anlamı port güvenlik duvarının arkasında demek. Port güvenlik duvarı tarafından korunuyor da diyebiliriz. Böylece port filtrelenmiş olarak kabul edilir.
d. Biz saldırganı şaşırtmak istiyorsak, port hakkında doğru bilgi almasını istemiyorsak, port güvenlik duvarı tarafından korunsa bile ona geri bildirim gönderebiliriz. Böylece saldırganın port hakkında doğru bilgi almasını zorlaştırabiliriz. Geri cevap olarak "RST TCP" gönderilir.

iptables -I INPUT -p tcp <port> -j REJECT –reject-with tcp-reset

3.2.SYN Scans

"TCP Conect Scan" ile benzer özelikleri olan "SYN Scans" taraması TCP' den farklı olarak hedef sunucudan bir SYN/ACK aldığında sunucuya "**RST TCP**" paketi gönderir.

Bununla birlikte bize sağladığı avantajları ve dezavantajları da vardır. Avantajları arasında;

1. Three way handshake arayan eski saldırı tespit algılama sistemlerini atlatmak için kullanılabilir. Bu artık modern IDS çözümlerinde geçerli olmadığından SYN taramalarına "**stealth**" denir.
2. Standart uygulamalarda bir bağlantı tamamen kurulduktan sonra günlüğe kaydedilir ama SYN taramaları genellikle açık portları dinleyen uygulamalar tarafından günlüğe kaydedilmez. İşte bu yüzden bu tarama türüne "stealth" denir.
3. TCP Connect Scan taramalarına göre daha hızlı bir tarama türüdür.

Bunların yanında dezavantajlarını sayacak olursak:

1. Bu tarama türünün düzgün çalışabilmesi için sudo izinlerine ihtiyaç var. Yani root olmanız ya da root'a erişip oradan çalıştırmalısınız. Bunun nedeni ise sadece root kullanıcıların sahip olduğu ayrıcalık olan ham paketler oluşturma becerisi gerektirmesidir.
2. Unstable servisler bazen SYN taramaları tarafından düşürülür. Bu bir işlemci test için bir üretim ortamı sağladıysa sorunlu olabilir.

NOT

→ Eğer sudo izini **varsa** nmap default olarak **SYN** taraması yapar.
→Eğer sudo izini **yoksa** nmap default olarak **TCP** taraması yapar.

3. SYN taramasında hedef port kapalı ise geri bir "RST TCP" paketi gönderilir. Eğer portta güvenlik duvarı varsa ya hiçbir şey göndermez TCP SYN bırakılır ya da bir TCP sıfırlamasıyla aldatılır.

NOT

→ "TCP Connect Scan" ve "SYN Scans" arasındaki fark açık portları nasıl işledikleri.

3.3.UDP Scans (-sU)

TCP taramasından farklı olarak bağlantı durumu belirsiz olan "UDP Scans" taramaları three way handshake yerine paketleri hedef porta göndermeye ve çalışmasını umduğu bir taramadır.

TCP taramalarına göre daha hızlı, ama TCP taramaları daha kalitelidir.

UDP video paylaşımı için mükemmel bir tercihtir.

UDP' nin onay eksikliği vardır ve bu da UDP taramasını zor hale getirir.

Eğer açık bir UDP portuna paket yollarsan büyük ihtimalle cevap alamayacaksın. Böyle bir durum olduğunda yani açık UDP portunda cevap gelmediğinde nmap, portu "**open|filtered**" olarak algılar, yani portun açık olduğundan şüphelenir ama güvenlik duvarı olabilir diyor. Fakat cevap alırsan port "açık" demektir ama böyle bir durum çok olağandışıdır.

Nmap' in "open|filtered" olarak algıladığı porta ikinci kez bir istek gönderilir (**double check**). Eğer yine cevap alınamazsa nmap o zaman o portu "open|fitered" olarak işaretler ve yoluna devam eder.

Kapalı bir UDP portuna paket gönderildiğinde ise hedeften geri cevap olarak portun ulaşılamaz olduğunu belirten bir mesaj içeren "**ICMP (ping)"** paketi iletilir. Nmap, böyle bir yanıt aldığında portu kapalı diye işaretler ve yoluna devam eder.

UDP portunun açık olup olmadığını belirlemek oldukça zor olduğundan bu işlem yavaştır: **"—top-port <number>"**

nmap -sU –top-port 20 <targer>

3.4.NULL, FIN ve Xmas

Üçünün de birbirine bağlı olduğu bu tarama türleri SYN taramasından daha gizli olma eğilimde olduklarından öncelikle kullanılırlar.

a. **NULL**: "-**sN**" anahtarı ile kullanılır. TCP isteğinin hiçbir flag ayarlanmadan gönderildiği zamandır. Yani boş paket gönderir. RFC' ye göre, port kapalıysa hedef ana bilgisayara bir "RST" ile yanıt vermelidir.
b. **FIN**: "-**sF**" anahtarı ile kullanılır. NULL ile hemen hemen aynı şekilde çalışır ama boş paket göndermek yerine "FIN" flag ile bir istek gönderir. Eğer port kapalıysa nmap "RST" bekler.
c. **Xmas**: "-**sX**" anahtarı ile kullanılır. Hatalı biçimlendirilmiş bir TCP paketi gönderir. Port kapalıysa "RST" bekler. Gönderdiği flaglar ise: PSH, URG ve FIN idir.

Bu taramalar sonucu portlar "**open|filtered**", "**closed**" ya da "**filtered**" olarak işaretlenir. Eğer bir port "filtered" olarak işaretlenmişse, bunun nedeni hedefin erişilemez olduğunu belirten ICMP paketi ile yanıt vermesidir.

NOT

→ RFC 793, ağ ana bilgisayarların hatalı biçimlendirilmiş paketlere kapalı portlar için bir RST TCP paketi ile yanıt vermesini beklerken açık portlar için yanıt vermemesini ister.
→Ama Microsoft Windows' ta portun açık veya kapalı olup olmadığı bakılmaksızın hatalı paketleniş herhangi bir TCP paketine bir RST ile yanıt verir. Bu tüm portların kapalı olmasına neden olur.
→Tüm bunlar hedef "**firewall evasion** (güvenlik duvarından kaçınma)".

3.5.ICMP Network Scanning

Bir hedefe bağlanıldığında ilk hedefin o ağın haritasını öğrenmek olmalıdır. Yani hangi IP adreslerin aktif hostları içerdiğini, hangilerinin içermediğini öğrenmekten bahsediyorum. Bunu yapmak içinde "**ping sweep**" adı altında **nmap** aracından faydalanacağız.

Nmap bu haritayı öğrenmek için belirtilen port için mümkün olabilen bütün IP adreslerine bir ICMP paketi gönderir ve hangi IP adresinden yanıt aldıysa o IP adresini "**alive**" diye işaretler fakat yanıt gelmesi alive olduğunu her zaman göstermez.

Bunun için ise "-**sn**" anahtarını kullanırız. Kullanımı ise şöyle;

-nmap -sn 192.168.0.1-254 ya da

-**nmap -sn 192.168.0.0/24**

Bu anahtar (-sn) herhangi bir portu taramamasını söyler. İlk başta ICMP echo paketlerini güvenmeye zorlar. Ek olarak -sn anahtarı nmapin 443 numaralı portuna bir TCP SYN paketi ve hedefin 80 numaralı portuna TCP ACK (kök olarak çalıştırılmazsa TCP SYN) paketi göndermesine neden olur.

3.6.Nmap Scripts Engine (NSE)

Nmap için işlevsellik sağlayan önemli bir ektir. Ayrıca Lua programlama diliyle yazılmış olup güvenlik açıklarını taramaktan veya onlardan otomatik yararlanmaya kadar çeşitli etkinlikler yapar. NSE keşif için yararlı olabilir.

safe: hedefi etkilemeyecek
intrusive: not safe
vuln: güvenlik açıklarını taramak için
exploit: bir güvenlik açığından yararlanma girişimi
auth: kimlik doğrulamayı atlamayı doğrulamak için
brute: çalışan hizmetler için kimlik bilgilerini brüte forcelama
discovery: ağ hakkında daha fazla bilgi için çalışan hizmetleri sorgulama

Bu nmap dosyalarını bulmak için 2 yolun var. Nmap web sitesini ziyaret edeceksin ya da linuxta **"/usr/share/nmap/scripts**" içindedirler.

Belirli bir script dosyasını çalıştırmak istiyorsan:

--script=<script-name> ya da

--script=http-fileupload-exploiter gibi

kullanabilirsin. Birden çok komut dosyasını aynı anda çalıştırmak istiyorsan virgülle ayırarak çalıştırabilirsin:

--script=smb-enum-users,smb-enum-shares gibi.

Bazı script dosyaları argüman gerektirir mesela kimliği doğrulanmış bir güvenlik açığından yaralanıyorsa kimlik bilgileri:

--script-args şeklinde kullanılabilir.

Eğer http-put scripti ile örnek verecek olursak argümanlar dosyayı yüklemek için URL ve dosyanın lokasyonu.

--nmap -p 80 --script http-put –script-args http-put.url='/dev/shell.php' , http-put.flie='./shell.php'

Şeklinde kullanılabilir. Bu script kullanımı hakkında daha detaylı bilgi görmek istiyorsan terminale

nmap --script-help <script-name> ile yardım alabilirsin.

Kali linuxta yüklü scriptleri aramak için 2 yol vardır. Bunlardan biri /usr/share/nmap/scripts/script.db dosyasıdır. Bu dosya script isimleri ve kategorilerini belirten bir metin dosyasıdır. Nmap "Script Engine" için scriptleri izlemek ve kullanmak için bu dosyayı kullanır.

Ama istersek scriptleri grep komutuyla da aratabiliriz:

grep “ftp” /usr/share/nmap/scripts/script.db

scriptleri aratmanın bir başka yolu ise ls komutu kullanarak aramak istediğimiz bir scripti bulabiliriz:

ls -l /usr/share/nmap/scripts/*ftp*

Yeni ve eksik scriptleri yüklemek için ise:

sudo apt update && sudo apt install nmap

işinizi görecektir ama eğer nmap üzerinden script indirmek istersen manuel olarak:

sudo wget -O /usr/share/nmap/scripts/script-name>.nse https://sun.nmap.org/nmap/scripts/script-name>.nse

yapıp bunun ardından script.db dosyasını yeni indirilen komut dosyasını içerecek şekilde güncelleyen

nmap –script-updatedb

komutu ile takip edilmeli.

3.7.Firewall Evasion

Windows ICMP paketlerini güvenlik duvarı ile bloklar. Bir paketin etkinliğini manuel olarak belirlemek için sadece “ping” komutu kullanılmıyor, bunu nmapde yapıyor. Yani nmap güvenlik duvarı yapılandırmasına sahip ana bilgisayarı ölü olarak kaydeder ve onu tarama zahmetine girmez.

Nmap bize güvenlik duvarı yapılandırmasını atlatmak için **“-Pn”** anahtarını sağlar.Ama bu kesin her zaman kesin çözüm değildir. Bu anahtar (burada anahtar ingilizcede switch diye de geçer) nmape, taramadan önce hosta ping atmakla uğraşmamasını söyler. Yani nmap hostu her zaman canlı olarak görecek ve böylece host gerçekten ölüyse bile her porta double-check yapacak ve bu da zaman alacak.

Eğer zaten yerel ağda isen, Nmap’ in hostu belirlemek için ARP isteklerini kullanabileceğini unutma.

Başka yararlı anahtarlarda var. Bunlar:

-f: Paketleri küçük parçalara ayırarak güvenlik duvarı veya IDS tarafından algılanma olasılığını azaltmaya yarar.

-mtu <number>: Paketlerin boyutu üzerinden daha fazla kontrol sağlar. Gönderilen paket için maksimum iletim birimi boyutunu kabul eder ve bu 8’ in katı olmalıdır.

--scan-delay <time>ms: Gönderilen paketlerin arasına gecikmem eklemek için kullanılır. Eğer ağ kararsızsa bu fazlasıyla yararlı olabilir. Ayrıca firewall ve IDS’ den kaçınmak içinde kullanılır.

--badsum: Paketler için geçersiz sağlama toplamı oluşturmak için kullanılır. Gerçek bir TCP/IP stack' i bu paketi düşürür ama firewall paketin sağlama toplamını kontrol etme zahmetine girmeden potansiyel olarak otomatik cevap verebilir. Bu switch firewall ya da IDS varlığını belirlemek için kullanılabilir.

NOT

→ **--data-lengt:** Bu switch paketlerin sonuna rastgele uzunlukta rastgele veri eklemenize izin verir.

4.NETWORK SERVİCES

4.1.SMB

Tam adı **Server Message Block Protocol** olan SMB, bir ağdaki dosyalara, yazıcılara vs. diğer kaynaklara erişebilmek için kullanılan bir **client-server protokol**üdür.

Serverlar dosya sistemleri ve diğer kaynakları ağ üzerindeki clientlara sunar. Clientların kendi hard diski olabilir ama onlar server üzerindeki paylaşılan dosya ve diğer kaynaklara ulaşmak ister.

SMB bir **response-request** protokoldür. Bu da server ve client arasında bağlantı sağlamak için çok mesaj döner anlamına gelir.

Client TCP/IP, NetBEUI ya da IPX/SPX kullanarak bağlanır. Bağlantı kurulduktan sonra, clientler sunucuya (server) paylaşılanlara erişmelerine, açmalarına, okumalarına, yazmalarına veya yapmak istediğinizle ilgili izinler içeren komut (SMB' ler) gönderir. Bunlar SMB durumunda ağ üzerinden yapılır.

Microsoft Windows bunu Windows 95' ten beri destekler. Ayrınca Samba ve Linux da destekler.

4.1.1 Enumeration

Enumeration dediğimiz şey potansiyel saldırı vektörlerini bulmak ve sömürüye yardımcı olmak için hedef hakkında bilgi toplama sürecine denilen isimdir. Bu süreç zaman ve enerji israfı olmaması için önemlidir. Enumeration ile kullanıcı adları, parolalar, ağ bilgileri, hostnames, uygulama verileri, servisler gibi bilgileri toplamak için önemlidir.

Bir sunucuda bağlanabilen ve dosyaları görüntülemek veya aktarmak için kullanılabilen SMB paylaşım sürücüleri vardır. SMB bu noktada hassas bilgileri öğrenmek isteyen bir saldırgan için harika bir başlangıçtır. Ama bazen karşısına çıkan sonuçlar onu şaşırtabilir çünkü bu yöntemi bilen bir kişi saldırganı şaşırtmak isteyecektir.

Enumerationın ilk adımı hedef makinenin dosyaları hakkında daha fazla bilgi sahibi olmak için bir port taraması yapmaktır. Bunu yapmak için de "**enum4linux**" toolunu kullanabiliriz.

Linux ve Windowsta SMB paylaşımlarını numaralandırmak için kullanılan bir tool olan enum4linux Samba paketlerindeki toolların etrafında bir sarıcıdır ve SMB ile ilgili hedeften hızlı bir şekilde bilgi çıkarmayı kolaylaştırır.

Syntaxi ise basittir:

enum4linux [options] ip

-U: Kullanıcı bilgilerini al
-M: Makine listelerini al
-N: İsim listesi dökümü
-S: Paylaşım listesini al
-P: Şifre politikası bilgisini al
-G: Grup ve üye listesini al
-a veya **-A**: Yukarıdakilerin tümü

4.1.2 Exploiting

Bir sisteme girmenin en iyi yolu yanlış yapılandırmadan kaynaklandığı bir durumla karşılaştığınızda daha yüksektir. Yani bir kabuk bulduğunuzda sistemi girmenin en iyi yolu odur. Bu durumda anonim SMB paylaşım erişimden yararlanılabilir. Bir kabuğa yol açacak bilgileri elde etmemize izin verecek yaygın bir yanlış yapılandırmadır. Bunun için Samba' nın bir parçası olan SMBCLİENT kullanılabilir:

smbclient //[IP]/[SHARE]

Söz dizimini kullanarak SMB paylaşımına uzaktan paylaşabiliriz.

-U: Kullanıcıyı belirlemek için
-p: port belirlemek için

Ayrıca ana sunucuya ssh yapmak için gereken şey özel anahtarlardır.

4.2.Telnet

TCP/IP protokollerinden olan ve tam adı Telecommunication Network olan Telnet application katmanında işlev görür. İnternet ağı üzerindeki bir makineye uzaktaki bir başka makineden bağlanmak için geliştirildi. Varsayılan portu 23 olan Telnet günümüzde kullanılmamaktadır çünkü iletişimi **açık metin** halinde yaptığı için güvenli değildir. Yerini SSH almıştır. Kullanımından bahsedecek olursak:

telnet [IP] [port]

şeklinde kullanımı vardır. Ayrıca ek bilgi olarak CVE (Common Vulnerabilities and Exposures), açıklanan bilgisayar güvenlik açıklarının listesidir. Bunu internette arama yaparak bulabilirsiniz.

NOT

→ **Shell**, bir cihazda kod ya da komut yürütme elde etmek için kullanılabilecek bir kod veya program parçası olarak tanımlanabilir.
→**Reverse Shell,** hedef makinenin saldıran makineyle iletişim kurduğu Shell tipi.

4.3.FTP

File Transfer Protocol' un kısaltması olan FTP, ağ üzerinden dosyaların uzaktan transferine izin vermek için kullanılan protokoldür. Client-server modeli kullanılır. Komutları ve verileri çok verimli bir şekilde ileten FTP standart port olarak ise 21. portu kullanır.

FTP iki kanal kullanır. Bunlar:

1. **Command Channel:** Komut dosyalarını transfer etmek için ve komutlara yanıt vermek için
2. **Data channel:** Verileri transfer etmek için kullanılır.

Client, server ile bağlanmak için bir bağlantı başlatır. Server, kimlik bilgilerini doğrular ve sonra oturum açar. Oturum açıkken client sunucuda FTP komutlarını çalıştırabilir.

FTP' de active ve passive olmak üzere iki farklı bağlantı türü vardır:

Active FTP connectionda, client bir port açar ve dinler, serverin aktif olarak ona bağlanması gerekir.

Passive FTP connectionda, server bir port açar ve dinler, client ona bağlanır.

İki farklı kanaldan veri transferi gerçekleştirmenin sayesinde veri transferinin bitmesini beklemeden servere komutlar gönderebiliriz. Aksi takdirde sadece veri aktarımları sırasında komutlar girebilirsin ve bu da büyük dosya aktarımı veya yavaş internet bağlantıları için verimli olmaz.

Bir enumerating olayında bir tool hakkında bilgi almak için şu komutu kullanabilirsin:

tool [-h / -help / -help]

ya da

man [tool]

kullanabilirsin. Ayrıca Telnette olduğu gibi FTP' de de veriler şifrelenmez.

Şimdi çok hızlı bir şifre kırma toolu olan Hydra'dan basitçe bahsedeceğiz.

Hydra çok hızlı bir şifre kırma tooludur (online). Telnet, SSH, FTP, HTTPS, SMB gibi protokollerde kullanılabilir. Kullanımı ise basitçe şöyledir:

hydra -t 4 -l [user] -P /usr/share/wordlists/rockyou.txt -vV [machine IP] ftp

gibidir.

hydra: Hydra toolunu çalıştırmak için

-t 4: hedef başına paralel bağlantı sayısı

-l [user]: hedefinizdeki hesabın kullanıcısını gösterir.

-P: Olası şifrelerin listesini içeren dosyayı işaret eder.

-vV: Ayrıntılı modu çok ayrıntılı hale getirir ve her deneme için oturum açma + geçiş kombinasyonunu gösterir.

[machine IP]: Hedef makinenin IP' si

ftp/protocol: Protokolleri düzenleme

Hydra toolu ile şifreyi bulduktan sonra "**ftp [IP]**" komutuyla da hedef makinenin adresinden dosya transfer işlemine başlayabiliriz. Burada ftp üzerinden işlem gösterilmiştir. Başka bir protokol kullanıldığında onun üzerinden işlem yapılmalıdır.

"**get [filenames]**" komutu ile dosya alırız ve artık kendi makinemizden bu dosyaya bakabiliriz.