Kapsamlı Siber Güvenlik El Kitabı

Teori, Uygulama ve En İyi Pratikler

Yusuf Talha ARABACI

24 Eylül 2025

İçindekiler

Bölüm 1

BİLGİ GÜVENLİĞİ VE VERİ KORUMA

Giriş

Bilgi güvenliği, bir kuruluşun en değerli varlıklarından biri olan bilgiyi korumaya odaklanan stratejik bir alandır. Bu koruma, yalnızca teknik önlemlerle sınırlı kalmayıp, aynı zamanda yönetimsel ve fiziksel kontrolleri de içerir. Bilgi güvenliğinin temelini oluşturan ve uluslararası alanda kabul görmüş ilkeler ve kavramlar, bu alanda uzmanlaşmak isteyen her profesyonel için kritik öneme sahiptir.

1.1 Bilgi Güvenliğinin Temel Kavramları ve İlkeleri

1.1.1 CIA Triad: Gizlilik (Confidentiality), Bütünlük (Integrity), Kullanılabilirlik (Availability)

CIA Triad, bilgi güvenliğinin temelini oluşturan üç ana unsuru ifade eder ve güvenli bir sistemin yapı taşlarını oluşturur. Bu üç unsur, birbirinden bağımsız düşünülemeyen ve birbirini tamamlayan bir bütündür.

- **Gizlilik (Confidentiality):** Bu ilke, bilginin yalnızca yetkili kişiler tarafından erişilebilir olmasını sağlamayı amaçlar. Yetkisiz kişilerin hassas verilere erişimini engellemek için şifreleme, erişim kontrol listeleri ve veri sınıflandırma gibi yöntemler kullanılır. Gizlilik ihlali, kredi kartı bilgileri veya kişisel olarak tanımlanabilir bilgilerin (PII) çalınması gibi saldırılarla gerçekleşebilir. Bilginin doğru kişilere, izin, resmi erişim onayı ve bilme ihtiyacı (need-to-know) prensiplerine uygun olarak sunulması esastır.

- **Bütünlük (Integrity):** Bütünlük, bilginin yetkisiz kişiler tarafından değiştirilmemesi, doğruluğunun, eksiksizliğinin ve güvenilirliğinin korunması anlamına gelir. İki temel bütünlük türü bulunmaktadır: veri bütünlüğü ve sistem bütünlüğü. Veri bütünlüğü, bilgilerin yetkisiz değişikliklere karşı korunmasını hedeflerken, sistem bütünlüğü sunucular, ağ cihazları ve uygulamalar gibi bir sistemin kendisinin yetkisiz değişikliklerden korunmasını amaçlar. Bir dosya üzerinde kimin, ne zaman, hangi değişikliği yaptığının kayıt altına alınması, bütünlüğün sağlanması için kritik bir adımdır.

- **Kullanılabilirlik (Availability):** Bu ilke, yetkili kullanıcıların ihtiyaç duydukları anda bilgiye ve sistemlere kesintisiz bir şekilde erişebilmesini ifade eder. Bir sistemin çökmesi veya bir siber saldırı sonucu hizmet verememesi durumunda, kullanılabilirlik ilkesi ihlal edilmiş olur. Yedekleme sistemleri, saldırı önleme çözümleri ve iyi planlanmış bir altyapı, kullanılabilirliğin güvence altına alınması için zorunludur.

Bu üç unsur arasındaki ilişki, tek birinin ihlalinin diğerlerini de olumsuz etkileyebilmesiyle yakından ilişkilidir. Örneğin, bir Dağıtılmış Hizmet Engelleme (DDoS) saldırısı doğrudan kullanılabilirlik ilkesini hedef alırken, bir siber saldırganın veritabanına sızarak hem müşteri bilgilerini çalması (gizlilik ihlali) hem de bu verileri değiştirmesi (bütünlük ihlali), ardından da veritabanı sunucusunun çökmesine neden olması (kullanılabilirlik ihlali) tüm triad'ın tehlikeye girdiğini gösterir. Bu örnek, bilgi güvenliğinin bir bütün olarak ele alınması gerektiğini ve güvenlik zincirinin en zayıf halkası koptuğunda tüm sistemin savunmasız kalabileceğini ortaya koymaktadır.

Bu üç temel unsurun yanı sıra, modern güvenlik yaklaşımları ek kavramları da temel bileşenler olarak kabul etmektedir. Kimlik doğrulama (Authentication), Yetkilendirme (Authorization) ve İnkâr Edilemezlik (Non-repudiation) gibi ilkeler, CIA üçlüsünü tamamlayarak daha sağlam bir güvenlik duruşu oluşturur. Kimlik doğrulama, bilgiye erişim talep eden kullanıcının kimliğinin doğrulanmasını sağlarken, yetkilendirme kullanıcının sistemde ne yapabileceğini belirler. İnkâr edilemezlik ise, bir işlemin veya olayın kaynağının ispatlanabilirliğini sağlayarak ilgili kişilerin bu olayı reddedememesini garanti eder. Bu ek unsurlar, güvenlik kontrol mekanizmalarının sadece veri üzerinde değil, aynı zamanda kullanıcılar ve süreçler üzerinde de yoğunlaşması gerektiğini gösterir.

1.1.2 Bilgi Güvenliği Risk Yönetimi ve Metodolojileri

Bilgi güvenliği risk yönetimi, bir kuruluşun bilgi varlıklarını koruma amacıyla potansiyel tehditleri, zafiyetleri ve bunların yaratabileceği etkileri sistematik bir şekilde ele alan bir süreçtir. Etkili bir risk yönetimi programı, riskleri en aza indirmek ve iş sürekliliğini sağlamak için stratejik kararlar alınmasına yardımcı olur. Modern risk yönetimi, çeşitli metodolojiler ve çerçeveler kullanılarak uygulanır:

- **CTEM (Continuous Threat Exposure Management) Metodolojisi:** Sürekli tehdit risklerini izlemeye ve yönetmeye odaklanan bir yaklaşımdır. Beş ana aşamadan oluşur:
  1. **Kapsam (Scope):** Risk değerlendirmesi için önceliklendirilen alanların belirlenmesi
  2. **Değerlendirme (Assess):** Belirlenen alandaki tehditlerin ve risklerin analizi
  3. **Mobilizasyon (Mobilize):** Risk azaltma planlarının geliştirilmesi ve kaynakların tahsisi
  4. **Doğrulama (Validate):** Alınan önlemlerin etkinliğinin test edilmesi
  5. **İzleme (Monitor):** Sürekli izleme ve yeni tehditlerin tespiti
- **FAIR (Factor Analysis of Information Risk):** Siber riskleri finansal terimlerle ölçmeye odaklanan bir metodoloji. FAIR, risk analizini kantitatif hale getirerek, risk yönetimi kararlarının daha objektif ve veri odaklı olmasını sağlar. Temel bileşenleri:
  - Risk = Olasılık x Etki formülünü detaylandırır
  - Kayıp Olayı Sıklığı (LEF - Loss Event Frequency)
  - Kayıp Büyüklüğü (LM - Loss Magnitude)
  - Risk skorlarını finansal değerlere dönüştürme
- **OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation):** Operasyonel hedeflere dayalı bir risk değerlendirme çerçevesidir. Üç ana fazdan oluşur:
  1. **Organizasyonel Görüş:** Varlıklar, tehditler ve mevcut güvenlik uygulamaları belirlenir
  2. **Teknoloji Görüşü:** Teknik zafiyetler analiz edilir
  3. **Strateji ve Plan Geliştirme:** Risk değerlendirme sonuçlarına göre güvenlik stratejisi oluşturulur

Risk yönetiminin temel kavramları şunlardır:

- **Tehdit:** Bir sistemin zafiyetini kullanarak zarara yol açma potansiyeli taşıyan bir olay veya durumdur. Örnek olarak kötü amaçlı yazılımlar, doğal afetler veya kötü niyetli insiderlar verilebilir.
- **Zafiyet:** Bir sistemdeki, ağdaki veya uygulamadaki güvenlik açığı veya zayıflıktır. Örneğin, yama yapılmamış bir işletim sistemi veya zayıf parola politikaları bir zafiyet oluşturur.
- **Risk:** Bir tehdidin bir zafiyeti kullanarak bir varlığa zarar verme olasılığı ve potansiyel etkisidir. Risk, genellikle şu formülle hesaplanır:

$$Risk = Tehdidin\ Etkisi \times Olma\ Olasl$$

- **Etki:** Bir riskin gerçekleşmesi durumunda ortaya çıkacak zararın boyutu ve ciddiyetidir. Bu zarar, finansal kayıplar, itibar zedelenmesi veya yasal yaptırımlar şeklinde olabilir.

Pratik Örnek Senaryo: Web Uygulaması Risk Analizi

Bir e-ticaret şirketinin, SQL Enjeksiyonu saldırılarına karşı müşteri veritabanı güvenliğini değerlendirdiği bir risk analizi senaryosu aşağıda adım adım incelenmiştir.

1. **Varlık Tanımlama:** Korunması gereken en kritik varlık, müşteri adı, adresi, kredi kartı bilgileri ve sipariş geçmişi gibi hassas verileri içeren müşteri veritabanıdır.

2. **Tehdit Tanımlama:** Potansiyel tehdit, bir siber saldırganın web uygulamasındaki bir zafiyeti kullanarak veritabanına erişim sağlamasıdır.

3. **Zafiyet Tanımlama:** Web uygulamasının, kullanıcıdan gelen girdileri yeterince filtrelemeden doğrudan veritabanı sorgusunda kullanması, bir SQL Enjeksiyonu zafiyetine yol açmaktadır.

4. **Etki Değerlendirmesi:** Bu zafiyetin kullanılmasıyla ortaya çıkabilecek etkiler:

   - Müşteri verilerinin çalınması: Şirketin itibarına büyük zarar verir ve yasal para cezalarına yol açar. **(Etki: Çok Yüksek)**
   - Veri manipülasyonu: Yanlış sipariş bilgileri veya finansal veriler, iş süreçlerini aksatır. **(Etki: Yüksek)**

5. **Olasılık Değerlendirmesi:** Saldırganların SQL Enjeksiyonu yöntemini yaygın olarak kullandığı ve web uygulamasının aktif olarak internete açık olduğu göz önüne alındığında, bu zafiyetin istismar edilme olasılığı **(Olasılık: Yüksek)** olarak değerlendirilir.

6. **Risk Hesaplaması:** Yukarıdaki formüle göre: ‘Risk = Çok Yüksek Etki x Yüksek Olasılık = Çok Yüksek Risk‘.

Bu analiz, şirketin bu riski acilen ele alması ve zafiyeti gidermesi gerektiğini gösterir. Risk yönetimi metodolojileri, bu tür analizleri sistematik bir şekilde yaparak güvenlik programlarının temelini oluşturur. Öte yandan, risk yönetimi artık statik, bir defalık bir görev olmaktan çıkıp, sürekli izlenmesi ve düzeltilmesi gereken dinamik bir süreç haline gelmiştir. Risk yönetimi göstergeleri (KPI’lar), risk maruziyetini ölçmek ve risk azaltma çabalarının etkinliğini değerlendirmek için kullanılır. Bu durum, siber güvenlik programlarının durağan ”kurallara dayalı” yaklaşımlar yerine, sürekli iyileştirme ve adaptasyon gerektiren ”risk tabanlı” bir modele dönüştüğünü ortaya koymaktadır.

1.1.3 Güvenlik Görevleri Ayrılığı (Segregation of Duties)

Güvenlik Görevleri Ayrılığı (SoD), bir kritik işlemin baştan sona tek bir kişi veya entity tarafından tamamlanmasını engelleyen temel bir güvenlik prensibidir. Bu ilkenin temel amacı, içeriden kaynaklanan kötüye kullanımı ve dolandırıcılığı, birden fazla kişinin iş birliği olmadan neredeyse imkansız hale getirmektir. SoD, sistemlerin gizliliğini, bütünlüğünü ve kullanılabilirliğini korumak için tasarlanmış bir kontrol mekanizmasıdır.

Teorik olarak basit bir fikir olmasına rağmen, SoD’nin pratikteki uygulaması, saldırıları yavaşlatan ve iç tehditleri önemli ölçüde azaltan bir dizi kontrol noktası oluşturur.

Pratik Uygulama Örnekleri:

- **Sistem Yönetimi:** Bir sistem yöneticisinin, bir sistemde hem kullanıcı hesabı oluşturma, hem bu hesaba ayrıcalık verme hem de bu işlemlerin izlendiği denetim kayıtlarını silme yetkisine sahip olması en yüksek riski taşır. Etkili bir SoD modeli, hesap oluşturma görevini bir yöneticiye, log izleme ve denetleme görevini ise bir başka role atar. Bu ayrım, hataların ve kötü niyetli değişikliklerin fark edilmesini kolaylaştırır.

- **Ağ Güvenliği:** Eğer bir ağ mühendisi hem güvenlik duvarı kurallarını yazma hem de bunları onaylama yetkisine sahipse, sisteme arka kapı (backdoor) yerleştirmesi ve bunu fark edilmeden yapması riski doğar. SoD ilkesi, bir kişinin kuralı oluşturmasını, bir diğerinin bu kuralı gözden geçirmesini ve bir güvenlik yöneticisinin de son onayı vermesini gerektirerek bu riski ortadan kaldırır.

- **Yazılım Geliştirme:** Bir geliştiricinin, kodunu doğrudan canlı (production) ortama geçirmesini önlemek için, kodun yazılması, test edilmesi ve canlıya alınması görevleri farklı kişilere veya ekiplere atanır. Bu sayede, güvenlik açığı içeren veya test edilmemiş kodların kritik sistemlere ulaşması engellenir.

SoD, HIPAA, SOX, GDPR ve ISO 27001 gibi düzenleyici çerçeveler için temel bir gerekliliktir. Metinlerde bahsedilen, bir hastanenin HIPAA denetimini, bir teknisyenin hem kullanıcı ekleyebilmesi hem de izleme sistemlerini devre dışı bırakabilmesi nedeniyle geçememesi örneği, bu ilkenin yasal zorunluluklardaki somut yansımasını gözler önüne serer. SoD, aynı zamanda ”en az ayrıcalık” (least privilege) ve ”ihtiyaç duyulduğu kadar bilme” (need-to-know) ilkelerinin bir uzantısıdır; bir kullanıcıya görevini tamamlaması için sadece gerekli yetkileri vermek ve bu yetkileri kritik bir sürecin tek bir aşamasına sınırlamak, iç tehditlere karşı güçlü bir savunma hattı oluşturur.

1.1.4 Savunma Derinliği (Defense-in-Depth) Stratejisi

Savunma Derinliği (DiD), bir kuruluşu korumak için birden fazla güvenlik önlemini bir arada kullanan kapsamlı bir siber güvenlik stratejisidir. Bu strateji, bir savunma hattı aşılırsa, ek katmanların tehdidi durdurmasını sağlamayı amaçlar. Bu yaklaşım, askeri bir stratejiden ilham alır ve bir orta çağ kalesinin çok katmanlı savunmasına benzetilir. Saldırganın hendek, kale kapısı ve surlar gibi birden fazla engeli aşması gerektiği gibi, siber saldırganların da bir dizi güvenlik kontrolünden geçmesi gerekir.

DiD stratejisinin üç ana katmanı bulunmaktadır:

- **Yönetimsel (Administrative) Kontroller:** Bunlar, bir kuruluşun güvenlik politikalarını, prosedürlerini ve süreçlerini içerir. Örnekler arasında risk değerlendirmeleri, çalışan güvenlik eğitimleri, erişim kontrol politikaları ve olay müdahale planları yer alır.

- **Fiziksel (Physical) Kontroller:** Bu katman, fiziksel varlıklara ve BT sistemlerine yetkisiz erişimi önlemeyi amaçlar. Kilitli sunucu odaları, güvenlik kameraları, giriş kartları ve güvenlik görevlileri bu kategoriye girer.

- **Teknik (Technical) Kontroller:** En karmaşık katman olan teknik kontroller, donanım ve yazılım tabanlı çözümleri içerir. Bu çözümler arasında güvenlik duvarları, antivirüs yazılımları, saldırı tespit/önleme sistemleri (IDS/IPS), şifreleme ve yedekleme sistemleri bulunur.

Modern Savunma Derinliği Stratejileri

Tarihsel olarak, savunma derinliği stratejileri, geleneksel çevre tabanlı (perimeter-based) güvenlik modelleri etrafında şekillenmiştir; bu modeller, bir ağın çevresini korumaya odaklanır. Ancak, dijital dönüşüm, uzaktan çalışma ve bulut hizmetlerinin yaygınlaşması, geleneksel ”güvenilir ağ” kavramını ortadan kaldırmıştır. Bu yeni ortamda, saldırganların zaten ağın içinde olduğu varsayımıyla hareket eden bir ”Sıfır Güven” (Zero Trust) yaklaşımı benimsenmiştir. Bu stratejide, her erişim talebi sürekli olarak doğrulanır ve yetkilendirilir.

Modern DiD stratejisi, geleneksel kontrolleri (güvenlik duvarı, antivirüs) yeni çözümlerle birleştirir:

- **Ayrıcalıklı Erişim Yönetimi (PAM):** Süper kullanıcılar ve etki alanı yöneticileri gibi yüksek ayrıcalıklı hesaplara erişimi izler ve güvence altına alır.

- **Uç Nokta Ayrıcalık Yönetimi (EPM):** Tüm uç noktalarda ayrıcalıklı erişimi kısıtlar, yanal hareketi (lateral movement) engeller ve fidye yazılımı gibi kötü amaçlı yazılımlara karşı koruma sağlar.

- **Uyarlanabilir Çok Faktörlü Kimlik Doğrulama (MFA):** Kullanıcının konumu, saati veya cihaz türü gibi bağlamsal bilgilere göre kimlik doğrulama faktörlerinin uygulanmasını sağlar.

Bu yeni yaklaşım, bir güvenlik stratejisinin, değişen teknolojik ve operasyonel koşullara nasıl adapte olması gerektiğini gösteren önemli bir evrimdir. Saldırganların birincil savunma hattını aşması durumunda, iç mekanizmaların onları durdurması beklenir. Bu çok katmanlı ve sürekli doğrulama yaklaşımı, tek bir kontrol noktasının başarısızlığının tüm sistemi riske atmasını önler.

1.1.5 Need-to-Know ve Least Privilege Prensipleri

En Az Ayrıcalık (Least Privilege) ve İhtiyaç Duyulduğu Kadar Bilme (Need-to-Know) ilkeleri, siber güvenlikte erişim kontrolünü yöneten iki temel prensiptir. Her ikisi de risk azaltmayı hedefler, ancak farklı odak noktaları vardır.

- **En Az Ayrıcalık (Least Privilege):** Bu ilke, bir kullanıcının, programın veya sürecin, görevini yerine getirmek için gereken minimum erişim yetkisine sahip olması gerektiğini belirtir. Bu, bir çalışanın işini yapmak için sadece gerekli olan yetkileri alması ve fazlasına sahip olmaması anlamına gelir. Örneğin, bir veritabanına veri girmekle görevli bir kullanıcı, bu verileri silme veya veritabanı şemasını değiştirme yetkisine sahip olmamalıdır. Bu prensip, kötüye kullanımdan veya yanlışlıkla yapılan hatalardan kaynaklanabilecek potansiyel zararı en aza indirmeyi amaçlar.
- **İhtiyaç Duyulduğu Kadar Bilme (Need-to-Know):** Bu ilke, bir kullanıcının, işini tamamlamak için bilmesi veya erişmesi gereken bilgiye sadece o anda erişiminin olması gerektiğini vurgular. Bu prensip, gizli bilgilerin yayılmasını kısıtlamak için kullanılır ve en az ayrıcalık ilkesini tamamlar. En az ayrıcalık **ne yapabileceğine** odaklanırken, ihtiyaç duyulduğu kadar bilme **neye erişebileceğine** odaklanır ve daha spesifiktir.

Bu ilkelerin güvenlik ve operasyonel açıdan birçok faydası bulunmaktadır:

- **Azaltılmış Risk:** Düşük ayrıcalıklı bir hesabın ele geçirilmesi durumunda, saldırganın sistemi genelinde yayılabileceği etki alanı sınırlı kalır. Edward Snowden'ın, en üst görevi veritabanı yedeklemesi almak olmasına rağmen, yönetici ayrıcalıkları sayesinde milyonlarca gizli dosyayı sızdırabilmesi, bu ilkenin neden bu kadar kritik olduğunu gösteren çarpıcı bir örnektir.
- **Artırılmış Stabilite:** Sistemlere ve uygulamalara yapılan yetkisiz veya yanlışlıkla yapılan değişikliklerin etkisi, bu ilkelerle sınırlanır.
- **Geliştirilmiş Denetlenebilirlik:** Denetim süreçleri, sistemin en az ayrıcalık ilkesine göre tasarlandığı durumlarda daha kolay ve daha hızlı yürütülebilir.

Pratik Uygulama Örnekleri

Bu ilkelerin pratikte uygulanması, hem işletim sistemleri düzeyinde hem de merkezi yönetim araçları kullanılarak gerçekleştirilir.

Linux Sistemlerinde Uygulama (`chown` ve `chmod`)

Linux'ta dosya ve dizin erişimini yönetmek için `chown` ve `chmod` komutları kullanılır. `chown`, dosya sahipliğini ve grubunu belirlerken, `chmod` sahiplik grubuna ve diğer kullanıcılara yönelik okuma, yazma ve çalıştırma izinlerini ayarlar. Bu iki komutun birlikte kullanımı, en az ayrıcalık ilkesinin uygulanmasının temelini oluşturur.

Senaryo: Kritik bir yapılandırma dosyasına sadece belirli bir hizmet hesabının erişmesini sağlamak.

1. **Dosya Sahipliğini Değiştirme:** Dosyanın sahipliğini, uygulamayı çalıştıran kullanıcıya (`service_user`) ve gruba (`service_group`) verin.

   ```
   $ sudo chown service_user:service_group /etc/critical_app/config.yaml
   ```

2. **Erişim Yetkilerini Kısıtlama:** Dosya için sadece sahibine okuma ve yazma (`6`) izni, diğer herkese ise hiçbir (`0`) izni verin.

   ```
   $ sudo chmod 600 /etc/critical_app/config.yaml
   ```

   Bu komut, `config.yaml` dosyasını sadece `service_user`'ın okuyup yazabilmesini, diğer kullanıcıların ise dosyaya erişimini tamamen engellemesini sağlar.

Özel İzinler ve Pratik Senaryolar

Linux'ta SGID (`Set Group ID`) ve Sticky Bit gibi özel izinler, en az ayrıcalık prensibini daha detaylı uygulamak için kullanılır.

- **SGID** (`chmod 2xxx`): Bir dizine SGID izni verildiğinde, o dizin içinde oluşturulan yeni dosyalar, dizinin grubuna ait olur. Bu, paylaşılan bir çalışma dizininde çalışan farklı kullanıcıların, dosyaların sahipliğini ve yetkilerini korumasına yardımcı olur.

```
$ sudo chmod 2775 /shared/project_dir
```

- **Sticky Bit** (`chmod 1xxx`): Bir dizine Sticky Bit izni verildiğinde, dizin içindeki dosyalar sadece kendi sahipleri, dizin sahibi veya `root` kullanıcısı tarafından silinebilir. Bu, özellikle `tmp` dizini gibi herkesin dosya oluşturabildiği ortak alanlarda yanlışlıkla veya kötü niyetli silmeleri önler.

```
$ sudo chmod 1777 /tmp
```

Windows Ortamında Uygulama

Windows sistemlerinde, yerel yönetici haklarının merkezi olarak yönetimi, Grup İlkesi (Group Policy) aracılığıyla yapılır. Bu, "ayrıcalık kayması" (privilege creep) olarak bilinen ve bir kullanıcının zamanla artan yetkileri nedeniyle gereksiz riskler biriktirmesini önlemeye yardımcı olur.

Adım Adım Yönetim Süreci:

1. **GPO Oluşturma:** Active Directory'de, yerel yönetici haklarını kaldırmak için bir Grup İlkesi Nesnesi (GPO) oluşturulur.
2. **Politikayı Yapılandırma:** GPO düzenleme penceresinde `Computer Configuration > Preferences > Control Panel Settings > Local Users and Groups` yoluna gidilir.
3. **Yerel Yöneticiler Grubunu Düzenleme:** Buradan, "Administrators" grubuna yeni bir kural eklenir. `Update` aksiyonu kullanılarak, belirlenen kullanıcılar hariç tüm kullanıcıların bu gruptan silinmesi sağlanır. Bu işlem, yerel yönetici yetkilerini şirket genelinde temizleyerek yalnızca onaylanmış hesapların bu gruba dahil olmasını sağlar.

Pratikte, bazı eski veya kötü kodlanmış uygulamalar yönetici hakları olmadan çalışmayabilir. Bu tür durumlar için, kullanıcıya sürekli yönetici yetkisi vermek yerine, "just-in-time" (tam zamanında) ayrıcalıklar veren veya belirli uygulamalara anlık ayrıcalık yükseltme yetkisi tanıyan özel yazılımlar kullanılabilir. Bu çözümler, en az ayrıcalık ilkesini ihlal etmeden, çalışanların üretkenliğini sürdürmesine olanak tanır.

Bu prensiplerin uygulanması, teorik bir kavramdan pratik bir zorunluluğa dönüşen bir süreci temsil eder. İlk olarak bir olayı tanımlayan teorik kavramlar, Snowden sızıntıları gibi gerçek dünya örnekleri ile önem kazanmıştır. Bu önemin anlaşılması, eski uygulamalar gibi pratik zorlukları beraberinde getirmiş, bu zorluklar ise "just-in-time" erişim veya özel ayrıcalık yönetimi araçları gibi daha gelişmiş çözümlerin ortaya çıkmasını sağlamıştır. Bu dönüşüm, bir güvenlik ilkesinin, gerçek dünya sorunlarına yanıt olarak nasıl evrildiğinin ve bu evrimin yeni teknolojileri ve yönetim yaklaşımlarını nasıl doğurduğunun bir göstergesidir.

1.2 Bilgi Güvenliği Standartları ve Framework'ler

Bilgi güvenliği standartları ve çerçeveleri, bir kuruluşun bilgi varlıklarını etkili bir şekilde yönetmesi ve koruması için rehberlik sağlar. Bu çerçeveler, en iyi uygulamaları tanımlar ve denetlenebilir bir yapı sunar.

1.2.1 ISO 27001/27002 Information Security Management

ISO 27001, bir kuruluşun Bilgi Güvenliği Yönetim Sistemini (BGYS) kurması, uygulaması, sürdürmesi ve sürekli iyileştirmesi için gereklilikleri belirleyen uluslararası bir standarttır. Bu standart, bilgi güvenliğini yönetmeye yönelik risk tabanlı bir yaklaşım benimser. ISO 27001, gizlilik, bütünlük ve kullanılabilirlik ilkelerini benimseyerek bilginin güvenli bir şekilde yönetilmesini sağlamayı hedefler.

ISO 27001'in en önemli özelliklerinden biri sertifikasyon sürecidir. Kuruluşlar, standartlara uygunluklarını kanıtlamak için harici bir denetçi tarafından kapsamlı bir yerinde denetime tabi tutulurlar. Denetimi geçen kuruluşlar, üç yıl geçerli olan bir ISO 27001 sertifikası alırlar. Bu sertifika, kuruluşun güvenliğe olan bağlılığını uluslararası düzeyde gösteren bir güven sinyalidir.

1.2.2 NIST Cybersecurity Framework (CSF) Uygulamaları

Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) Siber Güvenlik Çerçevesi (CSF), siber güvenlik risklerini yönetmek ve azaltmak için kullanılan esnek bir rehberdir. Çerçeve, özellikle siber güvenlik programının ilk aşamalarında veya bir ihlali azaltmaya çalışırken kullanışlıdır ve daha teknik bir odak noktasına sahiptir. NIST CSF, beş ana işlevi etrafında yapılandırılmıştır:

- **Tanımlama (Identify):** Sistemlerin, varlıkların, verilerin ve yeteneklerin siber güvenlik risklerini anlamak.
- **Koruma (Protect):** Kritik hizmetlerin sunumunu sağlamak için koruyucu önlemler almak.
- **Tespit Etme (Detect):** Siber güvenlik olaylarının zamanında tespit edilmesini sağlayan faaliyetleri uygulamak.
- **Müdahale Etme (Respond):** Bir olay tespit edildiğinde, etkilerini sınırlamak için bir plan dahilinde harekete geçmek.
- **Kurtarma (Recover):** Siber bir olaydan etkilenen işlevleri geri yüklemek ve iyileştirmek için planlar yapmak.

Bu beş işlev, NIST CSF'nin ana omurgasını oluşturur ve kuruluşlara risk yönetimi stratejilerini iş ihtiyaçlarıyla uyumlu hale getirme konusunda yardımcı olur.

ISO 27001 vs. NIST CSF Karşılaştırması

ISO 27001 ve NIST CSF, bilgi güvenliği ve risk yönetimini farklı açılardan ele alan tamamlayıcı çerçevelerdir.

Özellik	ISO 27001	NIST CSF
Yapısı	Bir standarttır. Uygunluğu kanıtlamak için belirli ölçütleri karşılamanız gereken bir test gibidir.	Bir rehber veya kılavuzdur. Kuruluşlara siber güvenlik programı oluşturmaları için yol gösterir.
Amacı	Mevcut bir siber güvenlik programını güçlendirmek ve standardizasyon yoluyla güven oluşturmak için idealdir.	Siber güvenlik yolculuğunun erken aşamalarında olan veya yapılandırılmış bir yaklaşım arayan kuruluşlar için en iyisidir.
Kapsamı	Uluslararası kabul görmüştür. Genellikle büyük şirketler tarafından satıcılarından istenen bir gerekliliktir.	ABD federal kurumlarına yardımcı olmak için kurulmuştur, ancak herhangi bir kuruluş tarafından kullanılabilir. Müşteriler tarafından nadiren istenen bir gerekliliktir.
Maliyeti	Üçüncü taraf denetimleri ve sertifikasyon süreci nedeniyle maliyetlidir (5.000 ila 15.000 ABD Doları veya daha fazla).	Ücretsiz erişilebilir. Üçüncü taraf denetimi veya sertifikasyon gerektirmez.
Sertifikasyon	Resmi bir denetim süreci ve sertifika gerektirir.	Resmi bir sertifika süreci yoktur. Kuruluşlar uyumluluğu kendileri rapor edebilir.

Her iki çerçeve de benzer risk yönetimi süreçlerine dayanır ve önemli ölçüde örtüşür. Bir kuruluşa göre, NIST CSF uygulayan bir şirket, ISO 27001 uyumluluğuna %80 oranında yaklaşmış olur ve ISO 27001 de NIST CSF yönergelerinin yarısından fazlasını içerir.

1.2.3 COBIT 5 IT Governance Framework

COBIT (Control Objectives for Information and Related Technology), Bilgi Sistemleri Denetim ve Kontrol Birliği (ISACA) tarafından geliştirilen ve BT'nin iş hedefleriyle hizalanmasını sağlamak için kullanılan bir yönetişim çerçevesidir. COBIT 5, BT'nin kurumsal hedeflere değer katmasını sağlamak için beş temel prensibe dayanır:

1. **Paydaş İhtiyaçlarını Karşılama:** Çerçeve, BT'nin tüm paydaşların ihtiyaçlarını karşılayacak şekilde değer yaratmasına odaklanır.
2. **Kuruluşu Baştan Sona Kapsama:** COBIT 5, BT yönetişimini kuruluşun tüm süreçlerine, departmanlarına ve işlevlerine entegre etmeyi hedefler.
3. **Tek Bir Entegre Çerçeve Uygulama:** COBIT 5, diğer en iyi uygulama çerçeveleri ve standartlarla (ITIL, ISO 20000, ISO 27001 gibi) entegre çalışacak şekilde tasarlanmıştır. Bu, kuruluşların tek bir çerçeveye bağlı kalmak yerine, ihtiyaçlarına göre farklı standartların en iyi yönlerini birleştiren hibrit bir yaklaşım benimsemesini sağlar.
4. **Bütünsel Bir Yaklaşımı Etkinleştirme:** Bu ilke, BT yönetişiminin başarıya ulaşması için sadece süreçlere değil, aynı zamanda organizasyonel yapılara, kültüre, bilgiye, hizmetlere ve insan kaynaklarına da odaklanılması gerektiğini vurgular.
5. **Yönetişimi Yönetimden Ayırma:** COBIT, "yönetişim" (governance) ve "yönetim" (management) kavramlarını net bir şekilde ayırır. Yönetişim, paydaşların ihtiyaçlarını değerlendirme, yönlendirme ve performansı izleme ile ilgilenirken; yönetim, planlama, inşa etme, çalıştırma ve izleme ile ilgilenir.

Bu prensipler, bir kuruluşun BT yatırımlarından beklenen değeri gerçekleştirmesine yardımcı olur. COBIT'in diğer framework'lerle entegre bir şekilde çalışmak üzere tasarlanmış olması, modern BT yönetişiminin çok boyutlu ve karmaşık doğasını yansıtan bir yaklaşımdır.

1.2.4 COSO Internal Control Framework

COSO (Committee of Sponsoring Organizations of the Treadway Commission), bir kuruluşun iç kontrol süreçlerini tasarlamasına ve uygulamasına yardımcı olan bir çerçevedir. COSO'ya göre, iç kontrol, operasyonel, finansal raporlama ve uyumluluk hedeflerine ulaşma konusunda makul bir güvence sağlamak için tasarlanmış bir süreçtir. Çerçeve, beş ana bileşenden oluşur:

1. **Kontrol Ortamı (Control Environment):** Bir kuruluşun etik değerlere, dürüstlüğe ve liderlik taahhüdüne olan bağlılığını gösterir.
2. **Risk Değerlendirmesi (Risk Assessment):** Kuruluşun hedeflerine ulaşmasını etkileyebilecek riskleri tanımlama ve analiz etme sürecidir.
3. **Kontrol Faaliyetleri (Control Activities):** Riskleri azaltmak için tasarlanmış politikalar ve prosedürlerdir. Örneğin, erişim kontrolleri ve güvenlik politikaları bu kapsamdadır.
4. **Bilgi ve İletişim (Information and Communication):** İç ve dış iletişim kanallarının yasal, etik ve sektörel standartlara uygun olmasını sağlar. Bu, doğru bilginin zamanında ilgili taraflara iletilmesini içerir.
5. **İzleme (Monitoring):** Kontrollerin etkinliğinin sürekli olarak değerlendirilmesi ve gözden geçirilmesidir. Bu, dahili ve harici denetimlerle gerçekleştirilebilir.

COSO, kuruluşların iç kontrollerini iş süreçlerine entegre etmelerini sağlayarak, yasal ve düzenleyici gerekliliklere uyumlarını kolaylaştırır.

1.2.5 PCI DSS Payment Card Industry Standards

PCI DSS (Payment Card Industry Data Security Standard), ödeme kartı verilerini (örneğin, kredi kartı bilgileri) işleyen, saklayan veya ileten tüm kuruluşlar için geçerli olan bir güvenlik standartları setidir.

1.2.6 HITRUST Ortak Güvenlik Çerçevesi (CSF)

HITRUST CSF (Common Security Framework), özellikle sağlık sektörü için tasarlanmış kapsamlı bir güvenlik ve gizlilik çerçevesidir. Bu çerçeve, HIPAA, GDPR, ISO, NIST gibi birçok farklı standardı ve düzenlemeyi tek bir çatı altında birleştirir.

- **Yapı ve Bileşenler:**
  - 14 güvenlik kategorisi
  - 49 kontrol alanı
  - 156 kontrol referansı
  - 3 olgunluk seviyesi
- **Olgunluk Modeli:** HITRUST CSF, her kontrolün olgunluğunu 5 seviyede değerlendirir:
  1. Politika
  2. Prosedürler
  3. Uygulama
  4. Ölçüm
  5. Yönetim
- **Sertifikasyon Süreci:**
  - Öz-değerlendirme
  - Doğrulanmış değerlendirme
  - Sertifikalı değerlendirme
- **Faydaları:**
  - Birden fazla düzenleme ve standardın tek bir çerçevede birleştirilmesi
  - Risk bazlı yaklaşım
  - Ölçeklenebilir ve özelleştirilebilir kontroller
  - Sürekli iyileştirme modeli

Standardın temel amacı, kart verilerini koruyarak dolandırıcılığı önlemektir. PCI DSS, 12 ana gereksinimden oluşur. Bu gereksinimlerden biri, kullanıcı kimliklendirmesi ve erişim kontrolü ile ilgilidir.

Temel Gereksinimler ve Uygulama

- **Gereksinim 8: Kullanıcı Kimliklendirmesi:** Bu gereksinim, bilgisayara erişimi olan her bir kişiye benzersiz bir tanıtıcı atamasını zorunlu kılar. Ayrıca, güçlü parola politikaları gerektirir:
  - Parolalar en az 90 günde bir değiştirilmelidir.
  - Parola uzunluğu en az 7 karakter olmalıdır.
  - Parolalar hem sayısal hem de alfabetik karakterler içermelidir.
  - Kullanıcıların önceki dört parolasıyla aynı yeni bir parola belirlemesi engellenmelidir.
- **Kapsam Belirleme:** PCI DSS uyumluluğu için, öncelikle kart verilerinin işlendiği, saklandığı veya iletildiği tüm sistemleri ve uygulamaları içeren Kart Sahibi Veri Ortamının (Card Holder Data Environment - CDE) doğru bir şekilde belirlenmesi gerekir. Bu, gereksinimlerin hangi sistemlere uygulanacağını netleştirir.

PCI DSS, kuruluşların kart verilerini korumak için teknik ve yönetsel kontrolleri uygulamalarını zorunlu kılar.

1.3 Veri Sınıflandırması ve Yaşam Döngüsü Yönetimi

Veri sınıflandırması ve yaşam döngüsü yönetimi, verilerin doğru şekilde korunmasını, yönetilmesini ve nihayetinde güvenli bir şekilde imha edilmesini sağlamak için kritik öneme sahip süreçlerdir. Bu süreçler, kuruluşların yasal düzenlemelere (örneğin GDPR veya KVKK) uymasına ve veri kaybı riskini azaltmasına yardımcı olur.

1.3.1 Veri Kategorileri: Public, Internal, Confidential, Restricted

Veri sınıflandırması, verileri hassasiyetine ve önemine göre kategorize ederek her kategoriye uygun güvenlik önlemlerinin uygulanmasını sağlar. Tipik olarak, veriler dört ana kategoriye ayrılır:

- **Public (Genel):** Bu veriler, herhangi bir kısıtlama olmaksızın serbestçe kullanılabilir, yeniden kullanılabilir ve yeniden dağıtılabilir. Örnekler arasında basın bültenleri, şirket tanıtım materyalleri ve iş tanımları yer alır.
- **Internal (Kurum İçi):** Bu veriler yalnızca şirket personeli veya erişim yetkisi verilen çalışanlar için tasarlanmıştır. Yetkisiz ifşası genellikle büyük bir zarara yol açmaz, ancak yine de gizli tutulmalıdır. İç yazışmalar veya iş planları bu kategoriye örnek verilebilir.
- **Confidential (Gizli):** Gizli veriler, yetkili erişim ve özel yetkilendirme gerektirir. Bu verilerin yetkisiz ifşası veya kötüye kullanımı, şirkete önemli zararlar verebilir. Sosyal güvenlik numaraları (SSN) veya kart sahibi verileri (Cardholder Data) bu kategoride yer alır. Bu veriler, genellikle HIPAA veya PCI DSS gibi yasalarla korunur.
- **Restricted (Kısıtlı):** Bu, en yüksek hassasiyet seviyesine sahip veri kategorisidir. Kısıtlı verilerin yetkisiz erişimi veya ifşası, yasal suçlamalara, çok yüksek para cezalarına ve geri dönülemez itibar kaybına yol açabilir. Şirketin mülkiyetinde olan araştırma ve geliştirme verileri veya federal düzenlemelerle korunan veriler bu kategoriye girer.

1.3.2 Data Ownership ve Data Stewardship Modelleri

Veri sahipliği (Data Ownership) ve veri sorumluluğu (Data Stewardship), veri yönetişiminde iki farklı ancak birbirini tamamlayan roldür.

- **Veri Sahibi (Data Owner):** Bir veri kümesi üzerinde nihai karar verme yetkisine ve hesap verebilirliğe sahip olan kişidir. Veri sahipleri, verinin nasıl kullanılacağına, erişileceğine ve saklanacağına dair politikaları ve stratejik yönü tanımlarlar. Örneğin, bir finans departmanının verilerinin sahibi genellikle CFO'dur.
- **Veri Sorumlusu (Data Steward):** Veri sahibinin belirlediği politikaları operasyonel düzeyde uygulayan ve günlük veri yönetimi faaliyetlerini yürüten kişidir. Veri sorumluları, veri kalitesini, tutarlılığını ve uyumluluğunu sağlamaktan sorumludur.

Aşağıdaki tablo, bu iki rol arasındaki temel farkları özetlemektedir:

Özellik	Veri Sahibi (Data Owner)	Veri Sorumlusu (Data Steward)
Yetki	Veri kullanımı, erişimi ve saklanması hakkında nihai kararları verme yetkisi vardır.	Veri sahibinin belirlediği politikaları uygular ve operasyonel yönetimi yürütür.
Odak Alanı	Stratejik yön, risk yönetimi ve politika tanımına odaklanır.	Günlük veri kalitesini, tutarlılığını ve uyumluluğunu sağlamaya odaklanır.
Sorumluluk	Veri güvenliği politikalarını tanımlamak, riskleri yönetmek ve uyumluluğu sağlamaktan doğrudan sorumludur.	Veri kalitesini güvence altına almak, veriyi sınıflandırmak, dokümantasyonunu sağlamak ve uyumluluk kontrollerine yardımcı olmaktan sorumludur.

Veri sahipleri ve veri sorumluları, etkili bir veri yönetimi stratejisi için yakın iş birliği içinde çalışmalıdır.

1.3.3 Veri Yaşam Döngüsü Yönetimi

Veri yaşam döngüsü (Data Lifecycle), bir verinin ilk oluşturulduğu andan nihai olarak imha edildiği ana kadar geçtiği aşamalar dizisini ifade eder. Bu döngünün etkili bir şekilde yönetilmesi, verilerin güvenli, düzenlemelere uygun ve karar alma için erişilebilir kalmasını sağlar.

Veri Yaşam Döngüsü Aşamaları:

- **Yaratma veya Edinme (Creation/Acquisition):** Veri yaşam döngüsü, verilerin müşteri etkileşimleri, işlemler, IoT cihazları veya manuel girişler gibi çeşitli kaynaklar aracılığıyla üretilmesiyle başlar. Bu aşamada, toplanan verilerin kalitesi ve alakalılığı sonraki tüm aşamalar için temel oluşturur.

- **Depolama (Storage):** Veriler yaratıldıktan sonra, veritabanları, veri gölleri veya bulut depolama gibi ortamlarda saklanır. Hassas bilgileri korumak ve yetkili kullanıcıların erişimini kolaylaştırmak için sağlam yedekleme ve kurtarma süreçlerinin uygulanması bu aşamada kritiktir.

- **Kullanım (Use):** Depolanan veriler, iş stratejilerini yönlendiren içgörüler elde etmek için analiz edilir. Veri analitiği araçları, verilerdeki kalıpları, eğilimleri ve anormallikleri ortaya çıkarmada önemli bir rol oynar. Bu aşamada, yetkisiz erişimi veya kötüye kullanımı en aza indirmek için uygun kullanım politikalarının uygulanması gerekir.

- **Paylaşım (Share):** Yetkili kullanıcılar veya kuruluşlar arasında veri aktarımı ve paylaşımı yapılır. Bu aşamada, gizlilik ve bütünlük ilkelerinin korunması için şifreleme ve erişim kontrolleri gibi güvenlik önlemleri uygulanmalıdır.

- **Arşivleme (Archive):** Veri, aktif ortamdaki kaynakları boşaltmak amacıyla daha az sıklıkta erişildiği zamanlarda güvenli, düşük maliyetli bir depolama ortamına taşınır. Bu aşama, verilerin yasal saklama süreleri dolana kadar korunmasını sağlar.

- **İmha (Destroy):** Döngünün son aşamasıdır ve artık gerekmeyen verilerin güvenli bir şekilde yok edilmesini içerir. Bu süreç, yasal ve düzenleyici gerekliliklere uyum sağlamak için dikkatle yönetilmelidir.

Veri yaşam döngüsü modeli, verinin durağan bir varlık olmadığını, aksine sürekli değişen bir dizi aşamadan geçtiğini gösterir. Bu, her aşama için özel güvenlik politikaları ve kontrolleri gerektirir. Örneğin, veri ”kullanım” aşamasında DLP (Veri Kaybı Önleme) kontrolleri gerekirken, ”imha” aşamasında güvenli silme yöntemleri (fiziksel imha, üzerine yazma) zorunlu hale gelir. Bu dinamik, statik, tek boyutlu güvenlik çözümlerinin neden yetersiz kaldığını açıklar.

1.3.4 Metadata Yönetimi ve Otomatik Sınıflandırma Araçları

Metadata, verinin içeriği hakkında bilgi sağlayan veridir (örneğin, bir dosyanın oluşturulma tarihi, sahibi veya kaynağı gibi) ve otomatik veri sınıflandırması için kritik bir unsurdur. Otomatik sınıflandırma araçları, önceden tanımlanmış kural setleri veya içerik inceleme teknikleri kullanarak belirli bir dosya veya mesajın hassasiyetini belirler ve uygun şekilde etiketler.

Otomatik sınıflandırma, ERP sistemleri tarafından üretilen raporlar gibi, kullanıcı müdahalesi olmadan oluşturulan veriler için özellikle faydalıdır. Ancak, bu araçlar her zaman verinin bağlamını anlayamayabilir ve bu da yanlış eşleşmelere veya hassas verileri kaçırmaya neden olabilir. Bu tür zorlukları aşmak için, otomasyon ile kullanıcı odaklı yaklaşımlar birleştirilebilir. Örneğin, otomatik olarak bir etiket önerisi sunularak kullanıcıdan onay istenebilir. Bu yaklaşım, sistemin doğruluğunu büyük ölçüde artırır ve kullanıcı güvenini sağlar.

Otomatik sınıflandırma araçları, DLP yazılımları ile entegre bir şekilde çalışarak, verinin hassasiyetine göre otomatik olarak kontrol edilmesini ve uygun politikanın uygulanmasını sağlar. Örneğin, ”Gizli” olarak etiketlenmiş bir belgenin ağ üzerinden dışarıya aktarılması otomatik olarak engellenebilir.

1.3.5 Veri Saklama ve İmha Politikaları

Veri saklama (data retention) ve imha (disposal) politikaları, hangi verilerin ne kadar süreyle saklanacağını ve artık gerek duyulmayan verilerin nasıl güvenli bir şekilde yok edileceğini belirleyen yazılı kurallar bütünüdür. Bu politikalar, yasal ve düzenleyici gerekliliklere uyum sağlamak, depolama maliyetlerini azaltmak ve şirketi potansiyel davalardan korumak için hayati öneme sahiptir.

Veri imha yöntemleri, verilerin hassasiyetine ve ilgili mevzuat gerekliliklerine bağlı olarak seçilmelidir. Başlıca imha yöntemleri şunlardır:

1. **Mantıksal Silme:** Bu yöntemler, veriyi geri getirilemez hale getirmek için yazılımsal teknikler kullanır.
    - **Temizleme (Clearing):** Veri depolama cihazlarının üzerine yeni veriler yazılarak eski verilerin kurtarılmasını zorlaştıran bir tekniktir. Bu, orta düzeyde bir güvenlik sağlar.
    - **Arındırma (Purging):** Fiziksel teknikler veya ileri teknoloji kullanarak verileri okunamaz ve laboratuvar ortamında bile kurtarılamaz hale getiren bir yöntemdir. Degaussing veya kriptografik parçalama gibi teknikler bu amaçla kullanılır.
2. **Fiziksel Yok Etme:** Bu yöntem, depolama ortamını tamamen yok ederek verilerin geri getirilemez hale gelmesini sağlar. Bu, hassas veriler için en yüksek güvenlik seviyesini sunar. Örnekler arasında optik veya manyetik medyanın eritilmesi, yakılması, toz haline getirilmesi veya parçalanması yer alır.

Aşağıdaki tablo, veri imha yöntemlerini ve güvenlik seviyelerini karşılaştırmaktadır:

Yöntem	Tanım	Güvenlik Seviyesi	Kullanım Alanı
Clearing (Temizleme)	Verilerin üzerine yeni veriler yazılarak kurtarılmasının zorlaştırılması.	Orta	Daha az hassas veriler veya dahili kullanım için.
Purging (Arındırma)	Fiziksel teknikler veya özel algoritmalarla verilerin kurtarılamaz hale getirilmesi.	Yüksek	Hassas veriler veya regülasyonlara tabi veriler iç
Physical Destruction (Fiziksel Yok Etme)	Depolama cihazının eritme, yakma, parçalama gibi yöntemlerle fiziksel olarak yok edilmesi.	En Yüksek	En hassas ve yasal yükümlü taşıyan veriler için.

Bu politikalar, depolama alanını boşaltmanın yanı sıra, gelecekteki operasyonlar için hayati varlıkların yanlışlıkla silinmesini de önler.

1.4 Şifreleme Teknolojileri ve Anahtar Yönetimi

Şifreleme, verileri okunamaz bir biçime dönüştürerek yetkisiz erişimi engelleyen temel bir bilgi güvenliği teknolojisidir. Bu teknolojinin doğru şekilde kullanılması, verilerin hem durağan (at rest) hem de hareket halindeyken (in transit) korunmasını sağlar.

1.4.1 Simetrik ve Asimetrik Şifreleme Algoritmaları (AES, RSA, ECC)

Şifreleme algoritmaları iki ana kategoriye ayrılır:

- **Simetrik Şifreleme:** Bu yöntemde, hem şifreleme hem de şifre çözme için tek ve aynı gizli anahtar kullanılır. Yüksek performans sunduğu için büyük veri setlerinin şifrelenmesinde idealdir. **AES (Advanced Encryption Standard)**, günümüzde en yaygın kullanılan simetrik algoritmadır. Çeşitli anahtar uzunlukları (128, 192, 256 bit) sunar ve kablosuz ağ güvenliği, SSL/TLS protokolleri ve VPN'ler gibi birçok alanda kullanılır.
- **Asimetrik Şifreleme:** Bu yöntem, **açık anahtar** (public key) ve **özel anahtar** (private key) olmak üzere, matematiksel olarak ilişkili iki farklı anahtar kullanır. Açık anahtar herkesle paylaşılabilirken, özel anahtar yalnızca sahibinde gizli kalır. Veri açık anahtarla şifrelenir ve yalnızca ilgili özel anahtarla çözülebilir. Simetrik şifrelemeye göre daha yavaştır, bu nedenle genellikle küçük veri setlerinin (oturum anahtarları gibi) şifrelenmesi ve dijital imzalar için kullanılır.

Asimetrik şifrelemede kullanılan ana algoritmalar şunlardır:

- **RSA (Rivest–Shamir–Adleman):** Güvenliği, büyük sayıları çarpanlarına ayırmanın zorluğuna dayanır. Yaygın olarak kullanılan bir algoritmadır, ancak modern tehditlere karşı daha uzun anahtar boyutları (örneğin, 2048 veya 3072 bit) gerektirebilir.
- **ECC (Elliptic Curve Cryptography):** RSA'ya göre daha modern ve verimli bir alternatiftir. Aynı kriptografik gücü, çok daha küçük anahtar boyutlarıyla sağlar. Bu, mobil cihazlar ve Nesnelerin İnterneti (IoT) gibi sınırlı işlem gücüne sahip ortamlar için idealdir.

Aşağıdaki tablo, bu algoritmaların teknik özelliklerini özetlemektedir:

Özellik	**Simetrik Algoritma (AES)**	**Asimetrik Algoritma (RSA)**	**Asimetrik Algor**
Kullanılan Anahtar Sayısı	1 (Gizli anahtar)	2 (Açık ve özel anahtar)	2 (Açık ve özel a
Anahtar Boyutu (NIST Önerisi)	128, 192, 256 bit	2048, 3072, 7680 bit	224, 256, 384 bit
Performans (Hız)	Çok Hızlı	Yavaş	Çok Daha Hızlı
Temel Fonksiyon	Toplu veri şifreleme ve şifre çözme	Dijital imza, anahtar değişimi	Dijital imza, anah

Adım Adım Kod Örnekleri (Python)

Simetrik Şifreleme (AES) Örneği:

`cryptography` kütüphanesi ile AES-GCM modunda şifreleme.

```
from cryptography.hazmat.primitives.ciphers.aead import AESGCM
import os

# Rastgele 256-bit anahtar ve nonce oluştur
key = os.urandom(32)
nonce = os.urandom(12)

# Şifrelenecek metin
plaintext = b"Gizli mesajimiz."

# Şifreleme işlemi
aesgcm = AESGCM(key)
ciphertext = aesgcm.encrypt(nonce, plaintext, None)

print(f"Şifrelenmiş metin: {ciphertext.hex()}")

# Şifre çözme işlemi
try:
    decrypted_text = aesgcm.decrypt(nonce, ciphertext, None)
    print(f"Şifresi çözülmüş metin: {decrypted_text.decode('utf-8')}")
except Exception as e:
    print(f"Hata: {e}")
```

Asimetrik Şifreleme (RSA) Örneği:

`cryptography` kütüphanesi ile RSA anahtar çifti oluşturma ve şifreleme.

```
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.primitives import hashes
```

```
# Özel anahtar oluşturma
private_key = rsa.generate_private_key(
    public_exponent =65537,
    key_size =2048,
)

# Açık anahtarı türetme
public_key = private_key.public_key()

# Şifrelenecek veri
message = b"Bu mesaj RSA ile sifrelenecek."

# Açık anahtar ile veriyi şifreleme
encrypted = public_key.encrypt(
    message,
    padding.OAEP(
        mgf =padding.MGF1(algorithm =hashes.SHA256()),
        algorithm =hashes.SHA256(),
        label =None
    )
)

print(f"Şifrelenmiş veri: {encrypted.hex()}")

# Özel anahtar ile şifre çözme
decrypted = private_key.decrypt(
    encrypted,
    padding.OAEP(
        mgf =padding.MGF1(algorithm =hashes.SHA256()),
        algorithm =hashes.SHA256(),
        label =None
    )
)

print(f"Şifresi çözülmüş veri: {decrypted.decode('utf-8')}")
```

1.4.2 Hash Fonksiyonları ve Dijital İmzalar

Hash fonksiyonları, verinin bütünlüğünü sağlamak için kullanılan tek yönlü matematiksel algoritmalardır. Bir girdi (mesaj) ne kadar büyük olursa olsun, sabit boyutlu bir çıktı (özet veya hash değeri) üretir. Bu çıktının en önemli özelliği, girdideki en ufak bir değişikliğin bile çıktıda tamamen farklı bir değere neden olmasıdır (”çığ etkisi”). Bu tek yönlü özellik, özet değerinden orijinal verinin elde edilememesini sağlar.

Hash fonksiyonları, **dijital imza** sürecinde kritik bir rol oynar. Dijital imza, bir belgenin bütünlüğünü ve kaynağını doğrulamak için kullanılır. Eğer belgenin kendisi çok büyükse, tüm belgeyi imzalamak zaman alıcı ve kaynak yoğundur. Bunun yerine, belgenin hash değeri alınır ve bu küçük boyutlu özet değeri, göndericinin özel anahtarıyla imzalanır. Alıcı, imzalı özet değerini göndericinin açık anahtarıyla doğrulayarak mesajın bütünlüğünü kontrol eder. Bu yöntem, hem operasyonel maliyeti hem de iletişim yükünü önemli ölçüde azaltır.

1.4.3 Açık Anahtar Altyapısı (PKI) ve Sertifika Yönetimi

Açık Anahtar Altyapısı (PKI), dijital sertifikaları ve anahtar çiftlerini oluşturmak, yönetmek, dağıtmak, saklamak ve iptal etmek için gerekli olan bir dizi rol, politika, donanım, yazılım ve prosedürdür. PKI’nın temel amacı, elektronik

iletişimin güvenli bir şekilde aktarılmasını sağlamak, tarafların kimliğini doğrulamak ve verinin bütünlüğünü güvence altına almaktır.

PKI'nın temel bileşenleri şunlardır:

- **Sertifika Otoritesi (CA - Certificate Authority):** Güvenilir bir üçüncü taraf olan CA, dijital sertifikaları düzenler, imzalar ve iptal eder. Bir CA, kendi özel anahtarını kullanarak sertifikaları dijital olarak imzalar, böylece son kullanıcı sertifikasına olan güven, CA'nın anahtarına olan güvene dayanır.

- **Kayıt Otoritesi (RA - Registration Authority):** RA, bir sertifika talebinde bulunan kişinin kimliğini doğrulayan bir bileşendir. RA, talepleri alır, gerekli doğrulamaları yapar ve ardından sertifikanın düzenlenmesi için talebi CA'ya iletir. Güvenlik ve erişilebilirlik nedenleriyle RA genellikle CA'dan ayrı bir birim olarak işlev görür.

- **Sertifika Veritabanı:** Düzenlenen ve iptal edilen tüm sertifikaların saklandığı bir depodur.

- **Güven Zinciri (Chain of Trust):** Bu, bir Kök CA'dan başlayıp, ara CA'lar aracılığıyla son kullanıcı sertifikalarına kadar uzanan hiyerarşik bir yapıdır. Bir sertifikanın doğruluğu, zincirdeki her bir sertifikanın bir üst otorite tarafından imzalanmış olmasıyla doğrulanır. Bu zincir, bir sertifikanın geçerliliğini kontrol etmek için kritik bir rol oynar.

1.4.4 Donanım Güvenlik Modülleri (HSM) ve Anahtar Yönetim Sistemleri (KMS)

Anahtar yönetimi, şifreleme teknolojilerinin güvenli bir şekilde uygulanabilmesi için hayati önem taşır. Bu alanda iki temel teknoloji öne çıkar: Donanım Güvenlik Modülleri (HSM) ve Anahtar Yönetim Sistemleri (KMS).

- **Donanım Güvenlik Modülü (HSM):** Bir HSM, şifreleme anahtarlarını fiziksel olarak koruyan, kurcalamaya dayanıklı bir donanım cihazıdır. Bir "banka kasası"na benzetilebilir; anahtarlar HSM içinde oluşturulur, depolanır ve imha edilir ve bu cihazdan asla çıkarılamaz. FIPS 140-2 Seviye 3 gibi düzenleyici standartlara uygun olan HSM'ler, fiziksel bir saldırı durumunda anahtarları otomatik olarak yok etme yeteneğine sahiptir. Bu özelliği, onları bir "Güvenin Kökü" (Root of Trust) olarak işlev görmesi gereken kritik altyapılarda vazgeçilmez kılar.

- **Anahtar Yönetim Sistemi (KMS):** KMS, şifreleme anahtarlarının yaşam döngüsünü (oluşturma, rotasyon, dağıtım, depolama ve imha) büyük ölçekte yönetmeye odaklanan bir yazılım çözümüdür. KMS, erişim politikalarını uygulamayı, kullanımı izlemeyi ve denetim kayıtları tutmayı kolaylaştırır.

Sadece birini kullanmak, güvenlikte önemli boşluklar yaratabilir. Sadece KMS kullanmak "uygun ama riskli" olabilir, çünkü anahtarlar yazılımda depolandığından yanlış bir yapılandırma veya bir zafiyet nedeniyle ifşa olma riski taşır. Sadece HSM kullanmak ise "güvenli ama yönetimi zor"dur, çünkü geniş ölçekte politika uygulamayı ve anahtar rotasyonunu manuel hale getirir. Optimal çözüm, her iki teknolojinin birlikte kullanıldığı hibrit bir modeldir. Bu modelde, KMS yönetim ve ölçeklenebilirlik sağlar, HSM ise anahtarlar için en yüksek düzeyde fiziksel korumayı sunar.

Özellik	Sadece HSM	Sadece KMS	KMS + HSM Birlikte Kullanımı
Anahtarın Depolandığı Yer	Fiziksel bir donanım cihazı içinde.	Genellikle yazılım veya sanal makine içinde.	Anahtarlar HSM'de güvenli bir şekilde depolanır, ancak KMS üzerinden yönetilir.
Güvenlik Riski	Fiziksel olarak güvenlidir, ancak ölçeklenebilirlik ve yönetim eksikliği riskleri barındırır.	Kullanışlıdır, ancak yanlış yapılandırma veya zafiyetler anahtarları ifşa edebilir.	Anahtarlar güvende kalır ve kolayca yönetilebilir.
Kontrol & Yönetim	Manuel operasyonlar gerektirir; politika uygulaması ve izleme zordur.	Kuralları belirlemek, anahtarları döndürmek ve kullanımı günlüğe kaydetmek kolaydır.	Politikaları belirlemek, izlemek ve otomatikleştirmek için güçlü bir koruma katmanı mevcuttur.
Uyum Hazırlığı	Yeterli değildir; politika kontrolü ve denetlenebilirlik eksiktir.	Yetersizdir; "güvenin kökü" eksiktir.	Entegre politika kontrolü ve donanım koruması sayesinde tam uyumluluk sağlar.

Bu yaklaşım, siber güvenlikte "tek bir çözüm her derde deva olmaz" prensibinin bir kanıtıdır. En yüksek güvenliğin, farklı teknolojilerin en iyi yönlerini birleştiren katmanlı ve entegre çözümlerle elde edildiğini gösterir.

1.4.5 Kuantum-Güvenli Kriptografi ve Post-Kuantum Algoritmaları

Kuantum bilgisayarlar, mevcut şifreleme yöntemlerinin güvenliğini tehdit eden yeni bir risk alanıdır. Shor's algoritması gibi kuantum algoritmaları, günümüzün yaygın asimetrik şifreleme algoritmalarının (örneğin RSA) dayandığı matematiksel problemleri (büyük sayıları çarpanlarına ayırma) çözme potansiyeline sahiptir. Grover's algoritması ise simetrik algoritmaların (AES gibi) kaba kuvvet saldırılarına karşı direncini azaltabilir, ancak bu algoritmalara karşı korunmak genellikle anahtar boyutunu iki katına çıkarmakla mümkün olabilir.

Kuantum-Güvenli Kriptografi (PQC), standart bilgisayarlarda çalışabilen ancak kuantum bilgisayarların saldırılarına karşı dirençli olan yeni algoritmaların geliştirilmesidir. Bu algoritmalar, kuantum bilgisayarların çözemeyeceği varsayılan matematiksel problemlere (örneğin, kafes tabanlı kriptografi) dayanır.

Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), kuantum-güvenli kriptografi için yeni standartlar belirlemek amacıyla bir yarışma başlatmıştır. Bu sürecin sonucunda, **CRYSTALS-Kyber** (şifreleme için) ve **CRYSTALS-Dilithium** (dijital imzalar için) gibi kafes tabanlı algoritmalar birincil standartlar olarak seçilmiştir. Bu algoritmalar, yüksek güvenlik sunmanın yanı sıra, dengeli anahtar boyutları ve güçlü performans özelliklerine sahiptir.

1.5 Veri Kaybı Önleme (DLP) ve Veri Koruma Teknolojileri

Veri Kaybı Önleme (DLP), hassas verilerin yetkisiz kişilere ifşasını, aktarımını veya sızdırılmasını tespit eden, önleyen ve yöneten bir dizi araç ve süreçtir. DLP çözümleri, bir kuruluşun fikri mülkiyetini, kişisel olarak tanımlanabilir bilgilerini (PII) korumasına ve dijital gizlilik yasalarına uymasına yardımcı olur.

1.5.1 Ağ Tabanlı, Uç Nokta Tabanlı ve Depolama Tabanlı DLP

DLP çözümleri, odaklandıkları koruma alanına göre üç ana türe ayrılır:

- **Ağ Tabanlı (Network-based) DLP:** Ağ trafiğini gerçek zamanlı olarak izler ve hassas verilerin e-posta veya diğer ağ protokolleri aracılığıyla yetkisiz bir şekilde dışarı aktarılmasını engeller. Geleneksel olarak, tüm çalışanların ofis ağına bağlı olduğu ve trafiğin merkezileştirildiği on-premise (yerinde) ortamlarda temel bir bileşendi. Ancak, uzaktan çalışma ve bulut tabanlı mimarilerin yaygınlaşmasıyla, ağ tabanlı DLP'nin etkinliği azalmıştır, çünkü ağın dışındaki etkinlikleri veya bulut uygulamaları arasındaki veri akışını izleyemez.

- **Uç Nokta Tabanlı (Endpoint-based) DLP:** Dizüstü bilgisayarlar, masaüstü bilgisayarlar ve mobil cihazlar gibi bireysel uç noktalardaki verileri kontrol eder. Bu çözümler, verilerin kaynağında nasıl kopyalandığını, taşındığını, yüklendiğini veya paylaşıldığını izler ve engeller. Kullanıcı ağa bağlı olmasa bile çalışır ve bu nedenle uzaktan çalışmanın yaygın olduğu günümüz ortamında vazgeçilmez bir çözüm haline gelmiştir.

- **Depolama Tabanlı (Storage-based) DLP:** Veritabanları, dosya sunucuları ve bulut depoları gibi "durağan" haldeki verileri tarar. Bu çözümler, hassas verileri keşfeder, sınıflandırır ve uygun koruma önlemlerini (örneğin, şifreleme) uygular.

Uzaktan çalışmanın yaygınlaşmasıyla birlikte, veri güvenliğinin odak noktası ağ çevresinden, verinin asıl bulunduğu yer olan uç noktalara kaymıştır. Bu, güvenlik teknolojilerinin, iş yapış biçimlerindeki köklü değişikliklere nasıl yanıt verdiğinin somut bir örneğidir. Modern bir DLP stratejisi, uç nokta tabanlı çözümlerin temel bir bileşen olmasını gerektirir.

1.5.2 İçerik İnceleme (Content Inspection) ve Örüntü Eşleştirme (Pattern Matching) Teknikleri

İçerik İnceleme, DLP çözümlerinin hassas verileri tanımlamak için veri paketlerinin içeriğini analiz ettiği bir tekniktir. Bu süreç, hassasiyet belirten anahtar kelimeleri ("gizli" gibi) ve belirli yapıları arar. İçerik incelemenin temelinde yatan en güçlü tekniklerden biri de **Örüntü Eşleştirme (Pattern Matching)**'dir. Bu teknik, daha büyük bir metin içinde belirli karakter dizilerini veya kalıpları tanımlamayı içerir ve genellikle **Düzenli İfadeler (Regular Expressions - Regex)** kullanılarak uygulanır.

Pratik Örnekler: Hassas Veri için Düzenli İfadeler (Regex)

DLP politikaları, finansal veriler, kişisel kimlik bilgileri veya sağlık bilgileri gibi hassas verileri tespit etmek için regex desenlerini kullanır.

- **Kredi Kartı Numarası Eşleştirme (MasterCard Örneği):**
  MasterCard numaraları genellikle 51 ile 55 arasında başlayan 16 haneli sayılardır. Metin içinde çeşitli formatlarda (boşluklu, tireli, vb.) yazılabilen bu numaraları tespit etmek için karmaşık regex desenleri kullanılır.

  - `5[1-5][0-9]{14}`: Bu desen, 5 ile başlayan, ikinci hanesi 1-5 arasında olan ve toplamda 16 haneli düz bir sayı dizisini eşleştirir.
  - `(5[1-5][0-9]{14}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[0-9]|2720)[0-9]{12}`: Bu daha kapsamlı bir ifadedir ve MasterCard'ın yeni numaralandırma aralıklarını da kapsar.

- **TC Kimlik Numarası Eşleştirme:**
  Türkiye Cumhuriyeti Kimlik Numarası (TCKN), 11 haneli bir sayıdır ve belirli kurallara uyar.

  - Regex deseni: `[1-9]{1}[0-9]{9}[0-9]{1}$`
  - **Mantık:**
    - : İfadenin başında olduğunu belirtir.
    - `[1-9]{1}`: İlk hane 0 olamaz. 1 ile 9 arasında tek bir rakamdır.
    - `[0-9]{9}`: Sonraki dokuz hane 0 ile 9 arasında herhangi bir rakam olabilir.
    - `[0-9]{1}`: Son hane de bir rakam olmalıdır.
    - `$`: İfadenin sonunda olduğunu belirtir.

Bu basit örnekler, DLP'nin hassas verileri otomatik olarak nasıl tanımlayabildiğini ve kurum politikalarını uygulayabildiğini gösterir.

1.5.3 Veri Sınıflandırma Entegrasyonu ve Politika Uygulama

DLP çözümleri, otomatik veri sınıflandırma araçlarıyla entegre çalışarak, bir verinin hassasiyet seviyesini temel alarak politika uygulaması yapar. Bu entegrasyon, bir verinin (örneğin bir Word belgesi veya e-posta) sınıflandırma etiketi ile işaretlenmesini ve bu etikete göre DLP'nin uygun kontrolü uygulamasını sağlar. Örneğin, bir belge "Gizli" olarak etiketlendiğinde, DLP politikası bu belgenin ağ üzerinden e-posta ile gönderilmesini veya USB diske kopyalanmasını otomatik olarak engelleyebilir. Bu, güvenlik politikalarının verinin kendisiyle ilişkilendirilmesini ve yetkisiz veri aktarımlarının gerçek zamanlı olarak önlenmesini sağlar.

1.5.4 Bulut DLP ve Çoklu Bulut Veri Koruma Stratejileri

Bulut Veri Kaybı Önleme (Cloud DLP), bir kuruluşun bulut depolama hizmetleri, veritabanları ve uygulamaları içindeki hassas verileri korumaya odaklanan bir veri güvenliği stratejisidir. Bulut ve çoklu bulut ortamlarının karmaşıklığı ve ölçeği, DLP'nin manuel olarak yönetilmesini zorlaştırır. Bu nedenle, bulut DLP çözümleri genellikle otomasyon ve yapay zeka (AI) araçlarını kullanır.

Bulut DLP stratejisinin temel adımları şunlardır:

1. **Veri Keşfi (Data Discovery):** Kuruluşun bulut altyapısı taranarak (örneğin, S3 depolama kovalıkları veya bulut veritabanları) PII, finansal kayıtlar veya fikri mülkiyet gibi hassas veriler keşfedilir.
2. **Veri Sınıflandırması (Data Classification):** Keşfedilen hassas veriler, önceden tanımlanmış kurallar ve politikalar doğrultusunda Public, Confidential, Restricted gibi kategorilere ayrılır.
3. **Politika Uygulama (Policy Enforcement):** Potansiyel bir politika ihlali tespit edildiğinde, bulut DLP çözümü, veri aktarımını engellemek, veriyi şifrelemek veya anonimleştirmek gibi önceden tanımlanmış eylemleri gerçekleştirir.

Manuel DLP süreçleri, geniş ve dinamik bulut ortamlarında yorucu ve hataya açık olabilir. Bu nedenle, otomasyon ve yapay zeka, bulut güvenliğinde sadece bir kolaylık değil, operasyonel bir zorunluluk haline gelmiştir. Otomasyon, insan hatasını en aza indirerek ve idari yükü azaltarak uzmanların daha stratejik görevlere odaklanmasını sağlar.

1.5.5 Veri Maskeleme (Data Masking) ve Anonimleştirme Teknikleri

Veri Maskeleme ve Anonimleştirme, hassas verilerin gizliliğini korumak için kullanılan tekniklerdir.

- **Veri Maskeleme:** Gerçek verinin, gerçek görünümlü ancak sahte verilerle değiştirilmesi işlemidir. Bu teknik, genellikle geliştirme, test veya analiz ortamlarında gerçek verilerin gizliliğini tehlikeye atmadan iş süreçlerinin devamını sağlamak için kullanılır. Örneğin, bir kullanıcının adı ve soyadı "J*** S****" gibi maskelenebilir.
- **Anonimleştirme:** Bir verinin, bir kişiyle ilişkilendirilemeyecek hale getirilmesi işlemidir. Kişisel Verilerin Korunması Kanunu'na (KVKK) göre, kişisel verinin kimliği belirlenebilir bir gerçek kişiyle ilişkilendirilemeyecek hale getirilmesi, bu verinin kişisel veri statüsünden çıkmasını sağlar.

Pratik Anonimleştirme Örnekleri:

- **Genelleştirme:** Verilerin daha genel bir kapsama indirgenmesidir. Örneğin, çalışanların tek tek yaşlarını belirtmek yerine, "X yaşında Z kadar çalışan bulunmaktadır" şeklinde genel bir ifade kullanılabilir.
- **Alt ve Üst Sınır Kodlama:** Verilerin önceden tanımlanmış kategorilere göre birleştirilmesidir. Örneğin, çalışanların kıdem yıllarını "5 yıldan az", "5 ile 10 yıl arasında" veya "10 yıldan çok" olarak anonim hale getirmek.
- **Değişken Çıkarma:** Veri setinden doğrudan kimlik belirleyici olan "Ad", "Soyad", "Adres" gibi değişkenlerin çıkarılması.

Veri imhası için ise, mantıksal veya fiziksel yok etme yöntemleri kullanılır. Örneğin, optik medyanın eritilmesi veya yakılması gibi fiziksel işlemler verilerin geri getirilemez hale gelmesini sağlar.

1.6 Uyum (Compliance) ve Düzenleyici Çerçeveler

Siber güvenlik politikaları, giderek artan bir şekilde yasal ve düzenleyici çerçeveler tarafından şekillendirilmektedir. Bu çerçeveler, kuruluşlara veri güvenliğini sağlama ve ihlallere karşı önlem alma konusunda yasal yükümlülükler getirir.

1.6.1 GDPR (General Data Protection Regulation) ve Tasarımla Gizlilik (Privacy by Design)

Genel Veri Koruma Tüzüğü (GDPR), Avrupa Birliği'nde kişisel verilerin korunmasını düzenleyen kapsamlı bir yasadır. GDPR, veri işlemeye yönelik temel ilkeleri belirler: hukuka uygunluk, dürüstlük, şeffaflık, amaç sınırlaması ve veri minimizasyonu.

GDPR'ın en önemli ilkelerinden biri, **Tasarımla Gizlilik (Privacy by Design)** kavramıdır. Bu ilke, gizliliğin bir ürünün veya sistemin tasarımının en başından itibaren düşünülmesi ve güvenlik önlemlerinin varsayılan olarak entegre edilmesi gerektiğini belirtir. Bu, gizliliğin sonradan eklenen bir özellik değil, temel bir mimari bileşen olmasını zorunlu kılar.

1.6.2 KVKK (Kişisel Verilerin Korunması Kanunu) Uygulamaları

Türkiye'de kişisel verilerin korunması, 6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK) ile düzenlenmiştir. KVKK, GDPR ile benzer ilkeleri benimsemekle birlikte, bazı önemli farklar bulunmaktadır.

- **Rıza Şartları:** KVKK, kişisel verilerin açık rıza olmadan işlenmesini genel bir kural olarak yasaklar ve belirli sınırlı durumlarda istisnalara izin verir. GDPR ise veri işlemeye yönelik rıza dışında daha geniş yasal zeminler (sözleşme, yasal yükümlülük, meşru menfaat gibi) sunar.
- **Hesap Verebilirlik:** GDPR, veri sorumlularının veri işleme faaliyetlerinin kanuna uygunluğunu ispatlamakla yükümlü olduğunu açıkça belirtir. KVKK'da bu ilke açıkça belirtilmemiştir, ancak veri sorumlularının kişisel verilerin güvenliğini sağlamak için gerekli önlemleri alma yükümlülüğü bu kavramı dolaylı olarak içerir.

1.6.3 HIPAA Sağlık Bilgileri Gizliliği

Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası (HIPAA), ABD'deki sağlık kuruluşları ve iş ortakları tarafından hasta bilgilerinin (Korunan Sağlık Bilgisi - PHI) gizliliğini ve güvenliğini korumayı hedefler. HIPAA, sağlık verilerinin toplanmasını, kullanılmasını ve ifşa edilmesini düzenleyen katı kurallar içerir.

HIPAA, okullardaki öğrenci sağlık kayıtları söz konusu olduğunda genellikle Aile Eğitimi Hakları ve Gizlilik Yasası (FERPA) ile birlikte değerlendirilir. Bir öğrencinin sağlık kayıtları, eğer okul tarafından tutuluyorsa, genellikle HIPAA değil, FERPA kapsamında yer alır.

1.6.4 SOX (Sarbanes-Oxley) Finansal Veri Koruması

Sarbanes-Oxley Yasası (SOX), halka açık şirketlerin finansal raporlamalarını ve iç denetim mekanizmalarını düzenleyen bir ABD yasasıdır. SOX, BT altyapılarında iç dolandırıcılığı önlemeye yönelik katı iç denetim ve erişim kontrolü gereklilikleri getirmiştir. SOX uyumluluğu, kullanıcı hesaplarına erişimin izlenmesini ve hassas sistemlerdeki aktivitelerin denetlenmesini zorunlu kılar.

1.6.5 Türkiye Siber Güvenlik Kanunu ve İşletmelere Etkileri

Mart 2025 itibarıyla yürürlüğe giren Türkiye Siber Güvenlik Kanunu, ülkenin dijital güvenliğini güçlendirmeyi ve siber tehditlere karşı daha dirençli bir altyapı oluşturmayı hedefler. Kanun, yalnızca kamu kurumlarını değil, kritik altyapı olarak tanımlanan finans, sağlık, enerji, ulaştırma ve telekomünikasyon gibi özel sektör kuruluşlarını da doğrudan etkilemektedir.

Bu kanun, işletmelere bir dizi yeni yükümlülük getirmektedir:

- **Siber Güvenlik Sorumlusu Atama Zorunluluğu:** 50'den fazla çalışanı olan özel şirketler ve kamu kurumları, organizasyon yapısı içinde bir siber güvenlik sorumlusu veya birimi oluşturmakla yükümlüdür.
- **Olay Bildirim Süresi:** Siber güvenlik olaylarının, olayın tespit edilmesini takiben 48 saat içinde ilgili birimlere bildirilmesi zorunludur.
- **Zorunlu Teknik Önlemler:** İşletmelerin güçlü kimlik doğrulama sistemleri, sızma testleri (penetration test), güncel antivirüs ve güvenlik duvarları gibi temel teknik tedbirleri alması zorunluluk haline gelmiştir.
- **Bağımsız Denetim:** Kritik altyapı sağlayıcıları, yılda en az bir kez bağımsız siber güvenlik denetimine tabi tutulacak ve bu denetim sonucunda "Siber Güvenlik Uygunluk Sertifikası" alacaktır.
- **Cezai Yaptırımlar:** Yükümlülüklere uymayan şirketlere ciddi idari para cezaları uygulanabilir (brüt satış gelirinin %5'ine kadar). Siber güvenlikle ilgili yanlış bilgi yayarak kamuoyunda panik yaratanlar için 2 ila 5 yıl arasında hapis cezası öngörülmüştür.

Bu kanun, siber güvenliği "tercih edilebilir bir iyileştirme" olmaktan çıkarıp, uyulması zorunlu ve ciddi yaptırımları olan bir hukuki yükümlülük haline getirmiştir. Bu durum, siber güvenlik politikalarının ve yatırımlarının üst düzey yönetim tarafından stratejik bir öncelik olarak ele alınmasını gerektirir. Kanun, sadece teknolojik çözümlerin değil, aynı zamanda kurumsal farkındalığın, risk yönetiminin ve sürekli denetimin de zorunlu hale geldiği yeni bir dönemi başlatır. Bu, regülasyonların teknolojik ve yönetsel dönüşümü nasıl tetiklediğinin önemli bir göstergesidir.

Bölüm 2

AĞ GÜVENLİĞİ MİMARİSİ VE ALTYAPI KORUMA

Giriş

Bu kapsamlı eğitim belgesi, siber güvenlik alanındaki uzmanlar için tasarlanmıştır ve modern ağ güvenliği mimarilerinin temelini oluşturan prensipleri, teknolojileri ve pratik uygulamaları detaylandırmaktadır. Bu bölüm, ağ altyapısının korunması için hem teorik hem de uygulamalı bir çerçeve sunmaktadır.

2.1 Ağ Güvenlik Mimarisi ve Tasarım Prensipleri

Modern ağ güvenliği, tek bir savunma hattına dayanmaktan ziyade, stratejik olarak katmanlanmış ve birbirini tamamlayan mekanizmalarla oluşturulan sağlam bir mimariyi gerektirir. Bu yaklaşım, bir siber saldırganın ağa erişimini zorlaştırmak, ihlalin etki alanını sınırlamak ve tespit edilme şansını artırmak için tasarlanmıştır.

2.1.1 Defense-in-Depth Network Architecture (Katmanlı Savunma Ağ Mimarisi)

Defense-in-Depth (DiD), bir bilgi güvenliği felsefesidir ve bir ağdaki gizlilik, bütünlük ve erişilebilirliği korumak için bir dizi güvenlik mekanizmasının ve kontrolünün katmanlar halinde yerleştirilmesini ifade eder. Hiçbir bireysel önlem tüm siber tehditleri durduramaz, ancak birlikte çalışarak çok çeşitli tehdit vektörlerine karşı koruma sağlarlar ve bir mekanizma başarısız olduğunda yedeklilik sunarlar. Bu strateji başarılı olduğunda, ağın güvenliğini birçok farklı saldırı vektörüne karşı önemli ölçüde güçlendirir. Bu katmanlı yaklaşım, bir ağda birden fazla güvenlik kontrolünün stratejik olarak konumlandırılmasını içerir. Bu katmanlar, geleneksel ağ güvenlik duvarlarından (firewalls), kötü niyetli trafiği tespit edip engelleyen Saldırı Tespit ve Önleme Sistemlerine (IDS/IPS) kadar uzanır. Ayrıca, kullanıcıların dizüstü bilgisayarları veya mobil cihazları gibi uç noktalara yerleştirilen antivirüs koruması, tehdit tespiti ve analizi sağlayan Uç Nokta Tespit ve Yanıt (EDR) yazılımlarını da kapsar. Temel bir adım olan ağ segmentasyonu, ağı iş ihtiyaçlarına göre alt ağlara ayırarak yanal hareketin önlenmesine yardımcı olurken, Çok Faktörlü Kimlik Doğrulama (MFA) ve şifreleme gibi önlemler, kullanıcı ve veri düzeyinde koruma sağlar. Bu güvenlik felsefesinin altında yatan temel mantık, siber güvenlikte ”tek bir gümüş kurşun”un olmadığıdır. Bir saldırgan, bir katmanı aşsa bile, bir sonraki katmanla karşılaşır. Bu durum, başarılı bir ağ ihlali için gereken süreyi ve karmaşıklığı artırır, böylece aktif bir saldırının tamamlanmadan önce tespit edilme ve durdurulma olasılığını yükseltir. Bu katmanlı savunma yaklaşımı, fiziksel güvenlikte de rutin olarak uygulanmaktadır. Örneğin, değerli varlıkları korumak için güvenlik kameraları, kurşun geçirmez cam ve kasaların kullanılması, aynı prensibin somut bir örneğidir. Bu paralellik, siber güvenlik uzmanlarının hem dijital hem de fiziksel güvenlik problemlerinin temelinde yatan ortak mantığı anlaması gerektiğini göstermektedir. Bir ağın savunmasını, bir kalenin kapıları, duvarları ve hendekleri gibi katman katman korumaya benzetmek, sadece bir ağa rastgele güvenlik araçları eklemekten daha fazlası olduğunu, bir güvenlik mimarisi oluşturma yaklaşımı olduğunu vurgular.

2.1.2 Network Segmentation ve Micro-segmentation Stratejileri

Ağ segmentasyonu, bir ağı daha küçük, izole edilmiş segmentlere ayırma uygulamasıdır ve siber güvenlik riskini azaltmanın hayati bir adımıdır. Mikro-segmentasyon ise bu yaklaşımın daha granüler bir formudur; ağı iş yükleri, uygulamalar ve hatta bireysel cihazlar düzeyinde izole eder. Bu strateji, bir saldırganın ağa ilk erişimi sağladıktan sonra yanal hareketini (lateral movement) zorlaştırarak bir ihlalin etki alanını (”blast radius”) sınırlar ve genel saldırı yüzeyini azaltır. Bu stratejiyi başarılı bir şekilde uygulamak, metodolojik bir süreç gerektirir:

1. **Değerli Veri ve Varlıkları Tanımlama:** Bir kuruluştaki tüm veri ve varlıklar aynı değere sahip değildir. Müşteri veritabanı gibi operasyonlar için hayati olan sistemlerin belirlenmesi ilk adımdır.
2. **Sınıflandırma Etiketleri Atama:** Hassasiyet seviyelerine göre varlıklara etiketler atayın (örneğin, ”Gizli,” ”Özel”). Bu etiketler, daha sonra ağ içindeki güven bölgelerini tanımlamak için kullanılır.
3. **Veri Akışlarını Haritalama:** Ağ genelindeki veri akışlarını haritalayarak uygulamalar ve sistemler arasındaki iletişim ve bağımlılıkları belirleyin. Bu, segmentasyon politikalarının oluşturulması için temel bir adımdır.
4. **Varlık Grupları Tanımlama:** Benzer amaçlara ve hassasiyet seviyelerine sahip varlıkları ayrı segmentlerde gruplandırın.
5. **Erişim Kontrol Politikaları Oluşturma:** En az ayrıcalık prensibine dayalı olarak segmentler arası iletişimi düzenleyen politikalar oluşturun. Bu, bir uygulamanın veya kullanıcının işini yapmak için gerekli olan minimum izin seviyesine sahip olmasını sağlar.

Ağ segmentasyonu, sadece bir saldırı kontrolü değildir; aynı zamanda bir **Sıfır Güven (Zero Trust) modelinin teknik temelidir**. Sıfır Güven, ağ içinde hiçbir şeye varsayılan olarak güvenmemeyi gerektirir. Bir ağ, mantıksal olarak izole edilmiş segmentlere ayrıldığında, her bir segment sınırı, bu ”hiçbir şeye güvenme” politikasının uygulanabileceği bir kontrol noktası haline gelir. Bu, teorik bir modelin pratik bir mimariye dönüşmesini sağlayan kritik bir adımdır.

2.1.3 Zero Trust Network Architecture (ZTNA) Modeli

Zero Trust Network Architecture (ZTNA), geleneksel güvenlik modellerinden önemli bir sapma gösteren bir yaklaşımdır. Bu model, bir ağda herhangi bir iç veya dış kullanıcıya ya da cihaza varsayılan olarak güvenmek yerine, her erişim isteğini sürekli olarak doğrular ve yetkilendirir. Geleneksel güvenlik, ağ çevresini bir kale gibi savunurken, ZTNA bir ihlalin zaten meydana geldiğini varsayar (”assume a breach”) ve güveni ağın her noktasına yerleştirir. ZTNA, geleneksel VPN’lerden farklı olarak, kimlik doğrulamayı kullanıcının IP adresine değil, daha gelişmiş kimlik tekniklerine dayandırır. Bu, kuruluşa, kullanıcı konumuna veya cihaz türüne göre erişim isteklerini otomatik olarak reddetme gibi özelleştirilmiş ve granüler güvenlik politikaları oluşturma esnekliği sunar. Bir kullanıcının kimliği bir kez doğrulandıktan sonra tüm ağa erişim sağlayan VPN’lerin aksine, ZTNA en az ayrıcalık prensibini uygulayarak yalnızca kullanıcının ihtiyaç duyduğu uygulamalara ve hizmetlere erişim sağlar. Bir ZTNA modelini uygulamak, teknolojik araçlardan daha fazlasını gerektiren metodolojik bir süreçtir. Aşağıdaki beş adım, Zero Trust yaklaşımını uygulamak için yaygın olarak kullanılan bir kılavuzdur:

1. **Koruma Yüzeyini Tanımlama:** Saldırı yüzeyini tanımlayın ve en değerli dijital varlıklara (hassas veriler, kritik uygulamalar ve hizmetler) odaklanın.
2. **Ağ Kontrolleri Uygulama:** Ağınızdaki trafik akışını ve sistem bağımlılıklarını anlayın.
3. **Bir Sıfır Güven Ağı Mimarisi Oluşturma:** Ağ segmentasyonunu uygulayın ve çok faktörlü kimlik doğrulamayı (MFA) dahil edin.
4. **Bir Sıfır Güven Politikası Oluşturma:** Her bir erişim isteği için ”Kim, Ne, Ne Zaman, Nerede, Neden ve Nasıl” (Kipling Metodu) sorularını sorarak politikalar tasarlayın.
5. **Ağı İzleme:** Potansiyel sorunları daha erken tespit etmek için ağ aktivitesini sürekli olarak izleyin; bunun için raporlar, analizler ve loglar kullanılır.

ZTNA'nın uygulanması, organizasyonel bir zihniyet değişikliği ve paydaş katılımı gerektiren kaynak yoğun bir süreçtir. Bu, güvenlik politikalarını otomatikleştirme ve sürekli doğrulama için gerekli süreç ve araçları yeniden tasarlamayı içerir. ZTNA, bir teknoloji satın almaktan ibaret değildir; bu, bir kuruluşun güvenliğe yaklaşımını baştan aşağı yeniden tanımlayan stratejik bir operasyonel yeniden yapılanmadır.

2.1.4 Software-Defined Perimeter (SDP) Yaklaşımları

Software-Defined Perimeter (SDP), ağ altyapısını internete karşı görünmez hale getirerek, DDoS, fidye yazılımı ve sunucu taraması gibi ağ tabanlı saldırılara karşı savunmasızlığı azaltan bir yaklaşımdır. SDP, bir kullanıcının kimliğini ve cihaz duruşunu doğrulayarak kaynaklara sanal bir sınır oluşturur ve yalnızca yetkili kullanıcıların erişimini sağlar. Bu yöntem, bir Zero Trust güvenlik modelini uygulamak için kritik bir rol oynar. SDP, geleneksel Sanal Özel Ağlara (VPN) göre önemli avantajlar sunar. Geleneksel VPN'ler, kullanıcılara ağa tam erişim sağlarken, SDP'ler özelleştirilmiş politikalara dayalı olarak yalnızca uygulamalara ve hizmetlere erişim sağlar. Bu, bir kullanıcının bir ağ varlığına erişimi olmaması durumunda onu görmesini de engeller. Ayrıca, SDP her kullanıcı için ayrı ve şifreli bir ağ bağlantısı oluşturarak bir saldırganın ağ içinde serbestçe dolaşma (roam) yeteneğini sınırlar. SDP, Zero Trust'ın "karanlık bulut" (dark cloud) konseptini hayata geçirir; yani kullanıcılar, izinleri olmayan uygulama ve hizmetleri göremezler. Bu, potansiyel bir saldırganın ağdaki kaynakları taramasını ve yanal hareket etmesini önleyerek saldırı yüzeyini önemli ölçüde azaltır. SDP, esnek, tutarlı ve merkezi politika yönetimi sağlar. Geleneksel sistemlerin aksine, SDP politikaları otomasyonu destekler, bu da dinamik BT ortamlarında ölçeklenebilirlik sağlar. Bu teknoloji, dizüstü bilgisayarlar, mobil cihazlar ve IoT aygıtları dahil olmak üzere geniş bir cihaz yelpazesini destekler. Özünde, SDP, geleneksel ağ merkezli (network-centric) VPN'lerden farklı olarak kimlik ve uygulama merkezli (identity- and application-centric) bir model sunar. Bu, onu modern, dağıtık ve bulut tabanlı çalışma ortamları için tasarlanmış, doğası gereği bir ZTNA platformu yapar.

2.1.5 DMZ (Demilitarized Zone) Tasarımı ve Best Practices

Demilitarized Zone (DMZ), bir kuruluşun yerel ağı (LAN) ile halka açık internet arasında ek bir güvenlik katmanı olarak işlev gören bir alt ağdır. DMZ, web sunucuları, e-posta sunucuları ve DNS sunucuları gibi, dış kullanıcıların erişmesi gereken, dışa dönük hizmetleri barındırır. Bu yapılandırmanın amacı, bu hizmetler ihlal edilse bile, saldırganın ana iç ağa doğrudan erişimini engellemektir. DMZ'yi tasarlamak için kullanılan iki ana mimari model bulunmaktadır:

- **Tek Güvenlik Duvarı (Üç Ayaklı Model):** En az üç ağ arayüzüne sahip tek bir güvenlik duvarı kullanılır. Bir arayüz dış ağa, bir arayüz iç ağa ve bir arayüz DMZ'ye bağlanır. Bu yapı, basitlik sağlar, ancak DMZ'deki bir ihlal durumunda iç ağın doğrudan korunması için ek bir güvenlik duvarı kadar sağlam olmayabilir.
- **Çift Güvenlik Duvarı Mimarisi:** İki ayrı güvenlik duvarı kullanılır. Birincisi, dış ağ ile DMZ arasında bir "ön uç" güvenlik duvarı olarak işlev görür ve yalnızca DMZ'ye yönelik trafiğe izin verir. İkincisi, DMZ ile iç ağ arasında "arka uç" güvenlik duvarı olarak çalışır ve yalnızca DMZ içindeki onaylanmış kaynaklardan gelen trafiğe izin verir. Bu daha güvenli bir yaklaşımdır ve hassas kaynakları barındıran kuruluşlar için tavsiye edilir.

DMZ'nin etkinliği, güvenlik duvarı kurallarının doğru bir şekilde uygulanmasına bağlıdır. İşte en iyi uygulamalardan bazıları:

- **Sıkı Güvenlik Duvarı Kuralları:** Dış güvenlik duvarı, yalnızca web sunucuları için TCP 80 ve 443 gibi belirli hizmet portlarına izin verecek şekilde yapılandırılmalıdır. Tüm diğer istekler engellenmelidir. İç güvenlik duvarı kuralları daha katı olmalıdır; yalnızca DMZ'deki onaylanmış kaynaklardan gelen trafiğe iç ağa erişim izni verilmelidir.
- **Ağ Adres Çevirisi (NAT):** Özel IP adreslerinin ve yönlendirme bilgilerinin yetkisiz kişilere ifşa edilmemesi için NAT'ı kullanın.
- **Düzenli Denetim:** DMZ'yi ve güvenlik duvarı kurallarını, amaçlandığı gibi çalıştıklarından emin olmak için düzenli olarak test edin ve denetleyin. Gereksiz, güncel olmayan veya yanlış kuralları temizleyin.

DMZ tasarımı, basit bir araç uygulamasından daha fazlasıdır; risk toleransına ve iş mantığına dayalı bir mimari kararıdır. Bir siber güvenlik uzmanı, bir DMZ'nin nasıl tasarlanacağını seçerken, kuruluşun barındırılan hizmetlerin hassasiyetine ve potansiyel bir ihlalin riskine dayalı olarak güvenlik ve karmaşıklık arasındaki dengeyi anlamalıdır.

2.2 Next-Generation Firewall (NGFW) ve Güvenlik Duvarları

Next-Generation Firewalls (NGFW), geleneksel güvenlik duvarlarının ötesine geçerek gelişmiş özellikler sunan üçüncü nesil güvenlik duvarı teknolojisidir. NGFW'ler, paket filtreleme ve durum bilgili denetim gibi temel yetenekleri, uygulama seviyesinde tehditleri ele alan akıllı ve bağlam odaklı güvenlik özellikleriyle birleştirir. Bu, modern siber tehditlere karşı daha kapsamlı bir koruma sağlar.

2.2.1 Stateful Packet Inspection vs Deep Packet Inspection

Geleneksel güvenlik duvarı teknolojisinin temelinde durum bilgili paket denetimi (Stateful Packet Inspection) bulunur. Bu yöntem, bir paketin yalnızca başlık bilgisini (kaynak/hedef IP adresi ve port numarası gibi) kontrol ederek ağ trafiğini kontrol eder. Bu, bir bağlantıdaki tüm paketlerin durumunu izleyerek trafik akışlarını daha etkili bir şekilde yönetir. Ancak, bu yaklaşım, paketin içeriğini incelemediği için başlık bilgilerini gizleyebilecek tehditlere karşı savunmasız kalır. Derin paket denetimi (Deep Packet Inspection - DPI), bu sınırlamanın üstesinden gelir. DPI, bir paketin başlığının yanı sıra taşıdığı verinin tamamını da inceler. Bu sayede, geleneksel filtreleme yöntemlerinin gözden kaçırabileceği, veri sızdırma girişimleri, içerik politikası ihlalleri veya zararlı yazılımlar gibi gizli tehditleri bulabilir. DPI, aynı zamanda, daha yüksek öncelikli trafiğe (VoIP veya Zoom gibi) öncelik vermek için de kullanılabilir. Ancak, DPI'nin bu gelişmiş güvenliği, bazı ödünleşmelerle birlikte gelir. DPI, ağ trafiği üzerinde büyük bir işlem yükü oluşturur, bu da ağ hızını ve performansını düşürebilir. Ayrıca, mevcut güvenlik duvarlarının karmaşıklığını artırır ve optimum etkinlik için periyodik güncellemeler ve revizyonlar gerektirir. Bu, siber güvenlikte "güvenlik, performans ve kullanılabilirlik" arasındaki temel bir ödünleşimi gösterir. Bir siber güvenlik uzmanı, DPI'nin sağladığı güvenlik faydalarını operasyonel maliyetlerinden ayrı düşünemez.

2.2.2 Application-aware Firewalling ve Layer 7 Security

OSI modelinin en üst seviyesi olan Layer 7 (Uygulama Katmanı), web tarama, e-posta ve dosya transferi gibi son kullanıcı deneyimlerini mümkün kılan protokollere ev sahipliği yapar. Bu katman, SQL injection, XSS ve API suistimali gibi uygulama katmanı saldırıları için birincil hedeftir. Geleneksel güvenlik duvarları yalnızca alt katmanlarda çalıştığı için, bu tür saldırılara karşı yetersiz kalır. NGFW'ler, DPI yetenekleri sayesinde Layer 7'de paketleri filtreleyebilir ve uygulama bazında karmaşık kurallar uygulayabilir. Bu, NGFW'lere uygulama farkındalığı ("application awareness") kazandırır. Örneğin, bir NGFW, bir port numarasına göre trafiği engellemek yerine, Facebook gibi belirli bir uygulamadan gelen trafiği tanıyabilir ve bu trafiğin içinde Facebook oyunlarını veya anlık mesajlaşma özelliklerini engelleyebilir. Web Uygulama Güvenlik Duvarı (WAF) gibi özel Layer 7 güvenlik çözümleri, web uygulamalarına yönelik saldırıları (OWASP Top 10 listesindeki saldırılar gibi) tespit etmek ve engellemek için tasarlanmıştır. Bu WAF'lar, web isteklerini izleyerek ve filtreleyerek çalışır. Modern NGFW'lerin ve WAF'ların yetenekleri arasında önemli bir örtüşme bulunmaktadır. Her ikisi de Layer 7 trafiğini işleyebilir ve kural motorları kullanır. Bu durum, güvenlik donanımlarının ve yazılımlarının tek bir platformda birleştiği bir trendi göstermektedir. Bu yakınsama, ağ altyapısı karmaşıklığını azaltır ve tek bir merkezi konsol üzerinden daha tutarlı bir güvenlik politikası yönetimini mümkün kılar.

Özellik	Geleneksel Güvenlik Duvarı	Yeni Nesil Güvenlik Duvarı (NGFW)
Paket Denetimi	Durum Bilgili (Stateful)	Durum Bilgili ve Derin Paket Denetimi (DPI)
Görünürlük	Yüzeysel, yalnızca alt TCP/IP katmanları (L3-L4)	Derinlemesine, tüm TCP/IP katmanları (L7 dahil)
Hizmetler	Temel, paket filtreleme	Kapsamlı (UTM hizmetleri: IDS/IPS, anti-virüs, içerik filtreleme)
Koruma	Sınırlı	Geliştirilmiş, çok çeşitli saldırıları algılar ve engeller
Uygulama Farkındalığı	Yok	Var (uygulama tabanlı filtreleme ve kontrol)
Tehdit Zekası	Yok	Dış tehdit zekası ağlarıyla iletişim kurar

2.2.3 Intrusion Prevention System (IPS) Integration

Saldırı Önleme Sistemi (IPS), kötü niyetli aktiviteyi sürekli olarak izleyen ve tespit ettiğinde otomatik olarak engelleyen bir ağ güvenlik aracıdır. Saldırı Tespit Sistemi (IDS), yalnızca kötü niyetli aktiviteyi algılayıp bir yöneticiyi uyarırken, IPS ek olarak bu aktiviteyi durdurmak için önleyici eylemde bulunur. IPS, tipik olarak ağ trafiği akışının içinde, genellikle güvenlik duvarının hemen arkasında konumlandırılır. Bu stratejik yerleşim, IPS'in ağ trafiğini gerçek zamanlı olarak analiz etmesini ve güvenlik duvarının gözden kaçırabileceği tehditleri yakalamasını sağlar. IPS, ağ trafiğini önceden tanımlanmış tehdit imzalarıyla (signature-based) veya normal davranışa karşı sapmaları izleyerek (anomaly-based) analiz eder. Bu, güvenlik duvarının birincil savunma hattını tamamlayarak, bir saldırının ilk katmandan sızması durumunda ikinci bir şans sunar. NGFW'ler, birleşik bir tehdit yönetimi (UTM) çözümü olarak IPS yeteneklerini sıklıkla bünyesinde barındırır. Bu entegrasyon, birden fazla güvenlik cihazının yönetim karmaşıklığını azaltır ve daha tutarlı bir güvenlik duruşu sağlar. IPS'in otomatik yanıt yetenekleri, BT ekipleri üzerinde yük oluşturmadan tehditlere hızla yanıt vermeyi mümkün kılar.

2.2.4 SSL/TLS Decryption ve Content Filtering

Modern internet trafiğinin büyük bir çoğunluğu SSL/TLS ile şifrelenmiştir. Bu, verilerin gizliliğini korurken, aynı zamanda zararlı yazılımların ve tehditlerin şifreli tünellerin içinde gizlenmesine de olanak tanır. Saldırganlar, şifreli kanalları komuta ve kontrol (C2) trafiği için kullanarak güvenlik araçlarından kaçınabilirler. NGFW'ler, bu zorluğun üstesinden gelmek için SSL/TLS sonlandırma proxy'leri olarak işlev görebilir. Bu, NGFW'nin hedefine ulaşmadan önce gelen ve giden şifreli trafiği şifrelemesini çözmesini ve içeriğini incelemesini sağlar. Şifresi çözülmüş trafik daha sonra kötü niyetli içerik, veri sızdırma girişimleri veya politika ihlalleri açısından derinlemesine incelenebilir. İnceleme tamamlandıktan sonra, trafik tekrar şifrelenir ve hedefine güvenli bir şekilde yönlendirilir. Bu yetenek, NGFW'lerin URL filtreleme, içerik filtreleme ve tehdit zekası entegrasyonu gibi gelişmiş güvenlik özelliklerini uygulamasını sağlar. İçerik filtreleme, kullanıcıların belirli türdeki web sitelerine veya içeriğe erişimini kısıtlar. Ancak, SSL/TLS şifre çözme işleminin uygulanması, güvenlik ve gizlilik arasında hassas bir denge gerektirir. Bu işlem, hassas verilerin, hatta kişisel bilgilerin bir güvenlik cihazı tarafından okunmasına neden olabilir, bu da yasal ve etik sonuçlar doğurabilir. Bu nedenle, bir kuruluşun bu yeteneği uygulamaya koymadan önce şeffaf politikalar oluşturması ve çalışanların güvenini yönetmesi kritik öneme sahiptir.

2.2.5 Firewall Rule Optimization ve Policy Management

Etkin bir güvenlik duruşu, yalnızca gelişmiş bir güvenlik duvarına sahip olmaktan değil, aynı zamanda onun kural tabanını ve politikasını ustaca yönetmekten geçer. Kötü yönetilen bir kural tabanı, performans sorunlarına, artan yönetim karmaşıklığına ve ciddi güvenlik açıklarına yol açabilir. Güvenlik duvarı kural optimizasyonu ve politika yönetimi için en iyi uygulamalar şunları içerir:

- **Kural Temizliği:** Süresi dolmuş, kullanılmayan ve gölgelenmiş (shadowed) kuralları düzenli olarak silin. "Herhangi bir" (Any) kaynak, hedef veya porta sahip kurallar güvenlikte boşluklar yaratabilir.

- **Belgeleme ve Adlandırma Kuralları:** Her kurala, amacını ve işlevini belirten açıklayıcı yorumlar ve adlar ekleyin. Tutarlı adlandırma kuralları, sorun giderme ve okunabilirliği büyük ölçüde artırır.
- **Politika Yapılandırması:** Kuralları hiyerarşik katmanlara ayırarak yönetimi kolaylaştırın. En çok isabet alan (top-hit) kuralları listenin en üstüne taşıyarak performansı optimize edin.
- **Otomasyon ve Denetim:** Kural değişiklikleri için resmi bir değişim kontrol süreci oluşturun ve güvenlik duvarı yapılandırmalarını düzenli olarak denetleyin. Otomatikleştirilmiş araçlar, fazlalıkları ve uyumsuzlukları tespit edebilir.

Zamanla biriken, kötü yönetilen kural tabanları, bir güvenlik duvarını yavaşlatır ve gözden kaçırılan politikalar nedeniyle savunmasız bırakabilir. Bu, operasyonel ihmallerin zamanla ciddi güvenlik risklerine dönüşebileceğini gösteren bir ”güvenlik borcu” kavramının örneğidir.

2.3 Network Intrusion Detection ve Prevention Systems

Ağ saldırı tespit ve önleme sistemleri (IDS/IPS), kötü niyetli aktiviteyi tespit etmek ve engellemek için ağ trafiğini sürekli olarak izleyen kritik güvenlik kontrolleridir. Bu sistemler, bilinen veya daha önce görülmemiş tehditleri tanımlamak için çeşitli metodolojiler kullanır.

2.3.1 Signature-based vs Anomaly-based Detection Methods

Siber tehditleri tespit etmek ve bunlara karşı uyarı vermek için kullanılan iki ana yöntem bulunmaktadır:

- **İmza Tabanlı Tespit:** Bu yöntem, bilinen tehditlerin önceden programlanmış bir listesine veya ”imzalarına” (Indicators of Compromise - IOCs) dayanır. Bir sistem, ağ trafiğindeki belirli bir desen veya kod dizisini bu listedeki imzalarla karşılaştırarak kötü niyetli aktiviteyi hızlı ve doğru bir şekilde belirler. Yüksek işlem hızına ve düşük yanlış pozitif oranına sahiptir, ancak daha önce görülmemiş sıfır-gün (zero-day) saldırılarını tespit edemez.
- **Anomali Tabanlı Tespit:** Bu yöntem, normal ağ davranışına ait bir ”baseline” oluşturur ve bu baseline'dan önemli sapmaları tespit eder. Örneğin, bir kullanıcının normal mesai saatleri dışında oturum açması veya yeni IP adreslerinin ağa bağlanmaya çalışması bir anomali olarak işaretlenebilir. Anomali tabanlı tespit, imza tabanlı sistemlerin kaçırabileceği sıfır-gün saldırılarını yakalamada etkilidir. Ancak, normal aktivitenin yanlışlıkla tehdit olarak algılanması nedeniyle daha yüksek yanlış pozitif oranları üretebilir.

Özellik	**İmza Tabanlı Tespit**	**Anomali Tabanlı Tespit**
Çalışma Prensibi	Bilinen kalıpları eşleştirme	Normal davranıştan sapmaları tespit etme
Tespit Odak Noktası	Bilinen tehditler, IOC'ler	Bilinmeyen ve sıfır-gün tehditler
Yanlış Pozitif Oranı	Düşük	Daha yüksek (ayarlamaya bağlı)
İşlem Hızı	Yüksek, bilinen saldırılar için hızlı	Değişken, analiz gerektirir
Gereken Kaynaklar	Düzenli imza güncellemesi	Sürekli ayarlama ve insan müdahalesi

2.3.2 Network Behavior Analysis (NBA) Teknikleri

Ağ Davranışı Analizi (NBA), ağ trafiği kalıplarını izleyerek, modelleyerek ve analiz ederek kötü niyetli aktiviteyi gösterebilecek anormallikleri tespit eden bir siber güvenlik tekniğidir. Geleneksel imza tabanlı sistemlerden farklı olarak, NBA, normal davranışın baseline'larını oluşturmak için istatistiksel modelleme ve makine öğrenimi gibi teknikleri kullanır. NBA'nın temel faydaları şunlardır:

- **Gelişmiş Tehdit Tespiti:** Geleneksel savunmaları aşan sıfır-gün saldırılarını, Gelişmiş Kalıcı Tehditleri (APTs) ve içeriden gelen tehditleri tespit edebilir.

- **Yanal Hareket Görünürlüğü:** Dahili trafik akışlarını analiz ederek kimlik bilgisi suistimali, yetkisiz veri erişimi veya VLAN'lar arası yanal hareket gibi tehditleri belirler.
- **Zero Trust Desteği:** NBA, davranışları sürekli olarak doğrulayarak ve güven sınırlarını uygulayarak Sıfır Güven mimarilerinde temel bir rol oynar. Bu, ZTNA'nın "sürekli güven doğrulaması" ve "sürekli güvenlik denetimi" yeteneklerinin altında yatan ana teknolojik bileşendir.

2.3.3 Threat Intelligence Integration ve IOC Matching

Siber güvenlikte, tehdit göstergeleri (Indicators of Compromise - IOCs), bir ağda devam eden veya potansiyel bir saldırıyı gösteren adli kanıtlardır. Bunlar, kötü amaçlı IP adresleri, şüpheli dosya hashleri veya anormal ağ trafiği gibi somut veriler olabilir. IOC'ler, saldırganların geride bıraktığı "dijital ekmek kırıntıları" olarak düşünülebilir ve güvenlik profesyonellerinin güvenlik olaylarını hızlı bir şekilde tanımlamasına, tespit etmesine ve yanıt vermesine yardımcı olur. Bir kuruluş, tehdit istihbaratı entegrasyonu sayesinde proaktif bir savunma duruşu benimseyebilir. Bu, tehdit istihbaratı beslemelerinden gelen IOC'leri SIEM sistemleri ve IDS/IPS gibi güvenlik araçlarına otomatik olarak besleyerek, bilinen kötü niyetli göstergelerin ağ trafiğinde taranmasını sağlar. Bu entegrasyon, saldırıların daha fazla zarar vermeden önce tespit edilmesini ve engellenmesini sağlar. Ancak, manuel IOC takibi ölçeklenemez bir süreçtir ve güvenlik ekiplerini yorabilir. Bu nedenle, otomatikleştirilmiş platformlar ve araçlar, gelen tehdit verilerini işleyerek ve yalnızca ilgili, yüksek öncelikli göstergeleri güvenlik kontrollerine ileterek bu süreci kolaylaştırır. Bu yaklaşım, tehdit istihbaratını pasif bir bilgi kaynağı olmaktan çıkarıp, aktif bir güvenlik eylemine dönüştürür.

2.3.4 False Positive Reduction ve Tuning Strategies

Yanlış pozitifler, meşru aktivitelerin kötü niyetli olarak yanlış bir şekilde etiketlenmesi durumudur. Bu durum, güvenlik analistlerini gerçek tehditlerden uzaklaştırarak "uyarı yorgunluğuna" ("alert fatigue") neden olabilir ve bir saldırının gözden kaçmasına yol açabilir. Anomali tabanlı tespitin birincil dezavantajı yüksek yanlış pozitif oranıdır. Bu riskleri azaltmak için aşağıdaki stratejiler uygulanmalıdır:

- **Sistem Ayarlaması (Tuning):** IDS/IPS araçları, her kuruluşun benzersiz trafik modellerine göre ayarlanmalıdır. Örneğin, planlı veri yedeklemeleri veya iş sürekliliği aktiviteleri sırasında meydana gelen normal trafik artışları istisna olarak işaretlenmelidir.
- **Ağ Segmentasyonu:** Ağı daha küçük, izole bölgelere ayırmak, güvenlik araçlarının daha yüksek öncelikli hedeflere odaklanmasına yardımcı olur ve yanlış pozitif gürültüsünü azaltır.
- **SSL/TLS Denetimi:** Şifreli trafik içindeki kötü niyetli yükleri (payload) tespit etmek için SSL/TLS şifre çözme çözümleri kullanılmalıdır.
- **Beyaz Liste (Whitelisting) Kullanımı:** Güvenilir IP adresleri veya hizmetleri beyaz listeye eklemek, gereksiz alarmları azaltabilir. Ancak, bu strateji, saldırganların yararlanabileceği kör noktalar oluşturmamak için dikkatli uygulanmalı ve düzenli olarak denetlenmelidir.

Bir IDS/IPS sisteminin etkinliği, teknolojinin kendisinden çok, onu doğru bir şekilde ayarlama çabasına bağlıdır. Başarılı bir siber güvenlik programı, teknolojik araçların sürekli bakım ve optimizasyonunu içerir. Bir uzman, bir aracın teknik kapasitesini olduğu kadar, operasyonel yükünü ve doğru sonuçlar vermesi için gereken insan çabasını da anlamalıdır.

2.4 Secure Remote Access ve VPN Teknolojileri

Uzaktan çalışma modelinin yaygınlaşmasıyla birlikte, ağa güvenli erişim sağlamak her zamankinden daha önemli hale gelmiştir. VPN'ler bu alanda geleneksel bir çözüm sunarken, Zero Trust Network Access (ZTNA) gibi daha modern yaklaşımlar, yeni güvenlik paradigmasına uygun alternatifler sunmaktadır.

2.4.1 IPSec VPN: Site-to-Site ve Remote Access Configurations

IPSec (Internet Protocol Security), genel ağlar üzerinden verilerin güvenli bir şekilde iletilmesine yardımcı olan bir protokol grubudur. VPN'ler, iki ana yapılandırma türüyle bu protokolü kullanır:

- **Siteden Siteye VPN:** İki farklı ağ arasında (örneğin, bir merkez ofis ile bir şube ofisi arasında) şifreli bir tünel oluşturur. Bu, her bir istemci makinenin şifreleme/şifre çözme yapmasını veya VPN istemci yazılımı yüklemesini gerektirmez.
- **Uzaktan Erişim VPN'i:** Bireysel bir kullanıcıyı (örneğin, evden çalışan bir çalışanı) kurumsal ağa güvenli bir şekilde bağlar. Bu, kullanıcının cihazına özel bir VPN istemci yazılımı kurulmasını gerektirir.

Siteden siteye bir VPN tüneli kurmak, hem fiziksel arayüzleri hem de tünel arayüzlerini yapılandırmayı içerir. Örneğin, Palo Alto Networks cihazında bu işlem, fiziksel arayüzlerin Layer 3 olarak tanımlanmasını, tünel arayüzlerinin oluşturulmasını ve IKE ile IPSec için kripto profillerinin tanımlanmasını içerir. Geleneksel IPSec VPN'lerin önemli bir sınırlaması, doğaları gereği kullanıcılara ağa tam erişim verme eğiliminde olmalarıdır. Bu durum, en az ayrıcalık (least privilege) prensibiyle çelişir ve bir ihlal durumunda saldırganın ağ içinde serbestçe hareket etme riskini artırır.

2.4.2 SSL/TLS VPN ve Web-based Remote Access

SSL/TLS VPN'ler, IPSec'ten farklı olarak, OSI modelinin uygulama katmanında (Layer 7) çalışır ve web tabanlı uygulamalara erişim için idealdir. En büyük avantajları, çoğu web tarayıcısının bu protokolü desteklemesi nedeniyle kullanıcıların özel bir istemci (clientless) yazılımı kurmasına gerek kalmamasıdır. Kullanıcılar, sadece bir web tarayıcısı üzerinden kimlik bilgileriyle oturum açarak kurumsal kaynaklara erişebilirler. Bu yaklaşım, özellikle web tabanlı uygulamalara erişim sağlarken yönetim ve dağıtım kolaylığı sunar. Ayrıca, BT ekiplerinin kullanıcılara uygulama bazında granüler erişim kontrolleri tanımlamasını kolaylaştırır. Ancak, istemci tabanlı VPN'ler, temel olarak HTTP-/HTTPS trafiğiyle sınırlıdır, bu da onları web tabanlı olmayan kaynaklara veya hizmetlere erişim için uygunsuz hale getirebilir.

Özellik	**IPSec VPN**	**SSL/TLS VPN (Clientless)**
OSI Katmanı	Ağ Katmanı (Layer 3)	Uygulama Katmanı (Layer 7)
Erişim Modeli	Ağ merkezli (ağa tam erişim)	Uygulama merkezli (uygulama bazlı erişim)
İstemci Gereksinimi	Genellikle özel yazılım gerektirir	Genellikle web tarayıcısı üzerinden erişim
Uygulama Alanı	Siteden siteye ve uzak kullanıcılara tam erişim	Uzaktan web tabanlı uygulamalara erişim
Granüler Kontrol	Düşük (birden fazla VPN kurulumu gerektirebilir)	Yüksek (uygulama bazlı politikalar)

2.4.3 Software-Defined WAN (SD-WAN) Security

Software-Defined WAN (SD-WAN), bir kuruluşun birden fazla WAN bağlantısı (MPLS, geniş bant, LTE) üzerinde merkezi olarak yönetilen bir yazılım katmanı oluşturarak ağ performansını ve verimliliğini artıran bir teknolojidir. Geleneksel WAN'ların aksine, SD-WAN, bulut uygulamalarına ve dağıtık şubelere yönelik trafiği optimize eder. SD-WAN'ın benimsenmesi, ağ sınırlarının geleneksel veri merkezinden dağıtık, bulut tabanlı bir modele kaydığını gösterir. Bu yeni mimari, güvenliğin de uç noktalara (edge) yakın bir şekilde uygulanmasını gerektirir. SD-WAN, yerleşik güvenlik yetenekleri sayesinde bu gereksinimi karşılar. Bunlar arasında, güvenli site-to-site bağlantıları için IPSec tabanlı VPN'ler, gelişmiş tehdit koruması için NGFW'ler ve güvenlik ile ağ fonksiyonlarını tek bir hizmette birleştiren SASE (Secure Access Service Edge) bulunur. Bu yaklaşım, güvenlik politikalarının merkezi bir kontrol düzleminden yönetilmesini sağlayarak karmaşıklığı azaltır ve tutarlı bir politika uygulamasını garanti eder.

2.4.4 Zero Trust Network Access (ZTNA) Platforms

Zero Trust Network Access (ZTNA), geleneksel VPN'lere modern bir alternatiftir ve tanımlanmış erişim kontrol politikalarına dayalı olarak uygulamalara ve hizmetlere güvenli uzaktan erişim sağlar. VPN'ler tüm LAN'a tam erişim

verirken, ZTNA çözümleri varsayılan olarak reddeder ve yalnızca kullanıcının açıkça erişim izni olan hizmetlere erişim sağlar.

ZTNA'nın temel faydaları şunlardır:

- **Saldırı Yüzeyini Azaltma:** Tıpkı SDP gibi, ZTNA da uygulamaları kullanıcılardan gizleyerek ("dark cloud") yanal hareket riskini azaltır.
- **Kimlik Tabanlı Kontrol:** ZTNA, erişim kontrolünü IP adresine değil, kullanıcı kimliğine ve cihaz duruşuna dayandırır. Bu, yalnızca yamalı ve güvenli cihazların kurumsal hizmetlere bağlanabilmesini sağlar.
- **Sürekli Doğrulama:** ZTNA, "bir kez doğrulandın, içeridesin" yaklaşımını terk ederek, bir kullanıcının oturumu boyunca güveni sürekli olarak doğrular ve şüpheli davranışları tespit ettiğinde erişimi gerçek zamanlı olarak iptal edebilir.

2.4.5 Mobile VPN ve BYOD Security Considerations

Kendi Cihazını Getir (BYOD) politikaları, esneklik ve maliyet avantajları sunarken, aynı zamanda zayıf parolalar, güvenli olmayan halka açık Wi-Fi ağları ve güncel olmayan işletim sistemleri gibi önemli güvenlik riskleri de taşır. Mobil VPN'ler, bu riskleri hafifletmek için hayati bir araçtır ve halka açık ağlar üzerinden hassas verilerin iletilmesini korur. Bir BYOD ortamını güvence altına almak için aşağıdaki hususlar dikkate alınmalıdır:

- **Net BYOD Politikası:** İzin verilen cihaz türlerini, veri sahipliğini, parola standartlarını ve mobil VPN kullanımını belirten net bir politika oluşturun.
- **Mobil Cihaz Yönetimi (MDM) Yazılımı:** Politikaları uygulamak, cihazları izlemek ve kayıp veya çalıntı cihazlardaki şirket verilerini uzaktan silmek için MDM çözümlerini kullanın.
- **Şifreleme:** Cihazların kendisinde ve bulutta depolanan verilerin şifrelenmesini zorunlu kılın.
- **Çalışan Eğitimi:** Çalışanları, halka açık Wi-Fi ağlarının riskleri, kimlik avı saldırıları ve veri koruma en iyi uygulamaları hakkında düzenli olarak eğitin.
- **Uzaktan Silme (Remote Wipe):** Kayıp veya çalıntı bir cihaz durumunda, şirket verilerini uzaktan silebilme yeteneği kritik öneme sahiptir.

BYOD güvenliği, teknolojik kontroller, açık politikalar ve insan faktörünün birleşimini gerektiren karmaşık bir alandır.

2.5 Wireless Network Security ve 802.11 Protokolleri

Kablosuz ağlar, esneklik ve hareketlilik sağlarken, aynı zamanda kötü niyetli aktörlerin hedefi olabilecek benzersiz güvenlik zorlukları da sunar. Modern güvenlik protokolleri ve araçları, bu riskleri yönetmek için hayati öneme sahiptir.

2.5.1 WPA3 ve Enterprise Wireless Security (802.1X/EAP)

WPA3, WPA2 standardına göre önemli güvenlik iyileştirmeleri getiren en son Wi-Fi güvenlik protokolüdür. En önemli yeniliklerinden biri, zayıf parolalara karşı koruma sağlayan ve "Simultaneous Authentication of Equals" (SAE) adı verilen yeni bir anahtar değişimi mekanizmasıdır. SAE, geleneksel önceden paylaşılan anahtar (PSK) yöntemlerinin aksine, çevrimdışı kaba kuvvet (brute-force) saldırılarına karşı koruma sağlar.

Kurumsal ortamlarda, Wi-Fi güvenliği için 802.1X/EAP protokolleri kullanılır. Bu yaklaşım, her kullanıcı için benzersiz, dinamik bir şifreleme anahtarı sağlamak için bir RADIUS sunucusuyla entegre olur. WPA3-Enterprise modu, 192-bit'lik bir şifreleme gücünü zorunlu kılarak daha sağlam güvenlik kontrolleri sunar.

2.5.2 Wireless Intrusion Detection Systems (WIDS)

Kablosuz Saldırı Tespit Sistemi (WIDS), kablosuz ağ trafiğini sürekli olarak izleyerek yetkisiz erişimi, sahte cihazları veya şüpheli aktiviteyi tespit eden bir teknolojidir. Bir Kablosuz Saldırı Önleme Sistemi (WIPS) ise, tespitten öteye geçerek tehdidi durdurmak için aktif önlemler alır. Bu önlemler, yetkisiz cihazı izole etmeyi veya ağdan ayırmayı içerebilir. WIDS ve WIPS, sahte erişim noktaları ve ortadaki adam (”evil twin”) saldırıları gibi kablosuz ağa özel tehditleri tespit etmekte ve bunlara karşı koruma sağlamakta etkilidir.

2.5.3 Rogue Access Point Detection ve Mitigation

Sahte erişim noktaları (Rogue APs), bir kuruluşun ağına yetkisiz olarak takılan ve ağa bir giriş noktası oluşturan kablosuz cihazlardır. Bunlar kasıtlı olarak kötü niyetli bir saldırgan tarafından veya iyi niyetli ancak bilgisiz bir çalışan tarafından (örneğin, bir Wi-Fi ölü noktasını gidermek için) kurulabilir. Her iki durumda da, sahte AP'ler veri ihlali, kötü amaçlı yazılım yayılması ve ortadaki adam saldırıları için bir güvenlik açığı oluşturur.

Bu tehditleri azaltmak için çok katmanlı bir yaklaşım benimsenmelidir:

- **Fiziksel Güvenlik:** Sunucu odaları ve çalışma alanları gibi ağ altyapısının bulunduğu alanlarda düzenli fiziksel denetimler yapın.
- **Kablosuz Tarama:** Kablosuz tarama araçları ve WIDS/WIPS sistemleri kullanarak yetkisiz AP'leri gerçek zamanlı olarak tespit edin.
- **Politika ve Eğitim:** Güvenlik ekibinin onayı olmadan herhangi bir kablosuz cihazın kurulmasını yasaklayan açık bir politika oluşturun ve çalışanları yetkisiz ağ değişikliklerinin riskleri hakkında eğitin.

2.5.4 Guest Network Isolation ve Captive Portal Security

Misafir ağları, ziyaretçilere internet erişimi sağlarken, kurumsal ağın güvenliğini korumak için tasarlanmıştır. Bu izolasyon, Sanal Yerel Ağlar (VLAN'lar) ve istemci izolasyonu (client isolation) kullanılarak gerçekleştirilir. VLAN'lar, ağ trafiğini mantıksal olarak bölerek misafir trafiğini dahili ağlardan ayırır. İstemci izolasyonu ise aynı ağa bağlı misafir cihazlarının birbirleriyle iletişim kurmasını engeller.

Misafir ağları genellikle bir kimlik doğrulama veya kabul ekranı sunan bir ”captive portal” kullanır. Captive portal güvenliği için en iyi uygulamalar şunlardır:

- **Güvenlik Duvarları:** Portal trafiğini kötü niyetli aktiviteye karşı izlemek için saldırı tespit sistemleri (IDS) ve güvenlik duvarları kullanın.
- **Log Yönetimi:** Ağ aktivitelerini takip etmek ve anormal davranışları tespit etmek için logları toplayın ve analiz edin.

2.5.5 IoT Device Wireless Security Challenges

Nesnelerin İnterneti (IoT) cihazları, genellikle sınırlı işlem gücü ve varsayılan kimlik bilgileri nedeniyle benzersiz güvenlik zorlukları sunar. Bu kısıtlamalar, cihazların kendisinde gelişmiş güvenlik kontrolleri (örneğin, yerleşik güvenlik duvarları) barındırmasını engeller.

Bu zorlukları aşmak için aşağıdaki çözümler uygulanmalıdır:

- **Ağ Tabanlı Güvenlik Duvarları:** IoT cihazlarının kendi güvenlik duvarlarını barındıramaması nedeniyle, ağ seviyesinde güvenlik duvarları kullanılarak kötü niyetli trafiğin cihazlara ulaşmadan önce engellenmesi sağlanır.
- **Sağlam Kimlik Doğrulama:** Varsayılan parolaları değiştirmek ve dijital sertifikalar gibi daha güçlü kimlik doğrulama mekanizmaları kullanmak esastır.
- **Özel Ağlar:** Hassas verilerin genel Wi-Fi ağları üzerinden gönderilmesi yerine, VPN'ler veya özel ağlar kullanılmalıdır.

- **Sınırlı Bağlantı Profili:** Cihazın ağ bağlantısını yalnızca temel işlevleriyle sınırlandırarak saldırı yüzeyini azaltın.

Bu yaklaşım, güvenliğin sadece cihazın kendisine değil, bağlı olduğu ağa da devredilmesini gerektirir ve siber güvenliğin paylaşımlı bir sorumluluk modeli olduğunu gösterir.

2.6 Network Monitoring, Analysis ve Forensics

Ağ izleme, analiz ve adli bilişim (forensics), bir siber güvenlik programının proaktif ve reaktif unsurlarını oluşturan hayati disiplinlerdir. Bu süreçler, ağın durumunu anlamak, tehditleri tespit etmek ve bir ihlal sonrası kanıt toplamak için gereklidir.

2.6.1 Network Traffic Analysis (NTA) ve Flow Monitoring

Network Traffic Analysis (NTA), ağ trafiği modellerini izleyerek ve değerlendirerek trafiğin ağ içinde nasıl aktığına dair içgörü kazanma uygulamasıdır. Bu, ağ güvenliği ve operasyonel ekiplerin tehditleri ve performans sorunlarını tespit etmesine yardımcı olur.

NTA için iki ana yöntem kullanılmaktadır:

- **Akış Tabanlı Analiz:** NetFlow, IPFIX ve sFlow gibi protokollerden gelen özet trafik kayıtlarını toplar. Bu yöntem, hangi kaynak/hedef adreslerin en çok bant genişliğini kullandığı gibi bilgilere odaklanarak geniş ölçekte görünürlük sağlar. Veri hacmini önemli ölçüde azalttığı için ölçeklenebilirdir.
- **Paket Tabanlı Analiz:** Ağdaki tüm paketleri yakalar ve içeriklerini inceler. Bu, en derinlemesine görünümü sağlar ve protokol el sıkışma hataları veya şifrelenmemiş iletişimlerin içeriği gibi düşük seviyeli sorunları teşhis etmek için çok değerlidir. Ancak, veri yoğun ve uzun vadede uygulanabilir değildir.

Etkili bir ağ izleme stratejisi, geniş ölçekli akış tabanlı izlemeyi, anormallik tespit edildiğinde belirli noktalarda tetiklenen paket yakalama yeteneği ile birleştirir.

2.6.2 SIEM Integration ve Log Correlation

Güvenlik Bilgi ve Olay Yönetimi (SIEM) sistemi, ağ cihazlarından, sunuculardan, uygulamalardan ve veritabanlarından güvenlik verilerini toplayan, normalleştiren ve ilişkilendiren merkezi bir platformdur. SIEM'in gücü, tek başlarına zararsız görünen farklı kaynaklardan gelen olaylar arasındaki ilişkileri ve kalıpları belirleme yeteneğidir. SIEM'de log ilişkilendirme, çok aşamalı saldırıları tespit etmek için kritik öneme sahiptir. Örneğin, bir sunucudaki başarısız oturum açma denemelerinin, ardından bir veritabanına başarılı bir erişimin gerçekleşmesi bir kaba kuvvet (brute-force) saldırısı olarak ilişkilendirilebilir. Bu süreç, tek tek güvenlik araçlarının yakalayamayacağı karmaşık saldırı senaryolarını ortaya çıkarır. SIEM, güvenlik verisine bağlam ekleyerek ve tehditlerin nasıl geliştiğine dair bir hikaye anlatarak, güvenlik ekiplerinin en kritik olaylara odaklanmasına olanak tanır.

2.6.3 Packet Capture ve Deep Packet Analysis

Paket yakalama (Packet Capture - PCAP), ağ üzerinden akan veri paketlerinin sistematik olarak kaydedilmesidir. Bu, ağ sorunlarını gidermek, performansı analiz etmek veya bir güvenlik olayı sonrası adli kanıt toplamak için paha biçilmez bir araçtır.

- **Araçlar:** Wireshark, paket yakalamayı ve detaylı protokol bilgilerini gösteren popüler bir grafik arayüzlü araçtır. `Tcpdump` ve `TShark` gibi komut satırı araçları, otomasyon ve sürekli paket yakalama için idealdir.
- **Filtreleme:** Performansı korumak ve ilgili verileri izole etmek için paket yakalama sırasında filtreler kullanılmalıdır. Filtreler, kaynak/hedef IP adresleri, port numaraları veya belirli protokoller gibi parametrelere göre ayarlanabilir.

Özellik	Wireshark	Tcpdump	TShark	NetworkMiner
Arayüz	GUI	CLI	CLI	GUI
Kullanım Odak Noktası	Manuel inceleme, sorun giderme	Düşük seviyeli yakalama, betik oluşturma	Otomatik analiz, sürekli kayıt	Adli analiz, eser çıkarma
Canlı Yakalama	Evet	Evet	Evet	Evet
Önemli Artıları	Detaylı protokol desteği, güçlü filtreler	Hafif, varsayılan olarak Linux'ta bulunur	Başsız (headless) Wireshark, otomasyon için uygun	Otomatik olarak dosya, resim, kimlik bilgisi çıkarır

2.6.4 Network Forensics Methodologies

Ağ adli analizi, bir ihlal sonrası ağ trafiğini ve loglarını inceleyerek saldırganın yolunu, yöntemlerini ve saldırının kapsamını yeniden inşa etmeyi amaçlayan bir disiplindir. Dijital adli bilişim ve olay yanıtı (DFIR) sürecinin bir parçasıdır ve veri toplama, inceleme, analiz ve raporlama aşamalarını içerir.

- **Veri Kaynakları:** Analistler, sistem logları, ağ trafiği yakalamaları (PCAP dosyaları) ve depolama cihazlarından elde edilen veriler gibi çeşitli kaynaklardan dijital kanıt toplar.
- **Analiz:** Toplanan kanıtlar incelenerek anormallikler ve şüpheli aktiviteler belirlenir. Örneğin, bir PCAP dosyasının incelenmesiyle bir "drive-by malware" saldırısı ortaya çıkarılabilir; anormal bir TCP portu üzerinden gerçekleşen şüpheli bir HTTP isteği, kötü amaçlı bir sunucuya (IOC) işaret edebilir.
- **Zorluklar:** Ağ adli analizi, şifreli trafik, yüksek veri hacimleri ve cihaz çeşitliliği gibi zorluklarla karşı karşıyadır. Modern ağların neredeyse tamamının şifreli olması, derin paket incelemesini engeller ve adli analiz yeteneklerini ciddi şekilde sınırlar. Bu durum, güvenlik profesyonellerinin veri akışını açığa çıkarabilen SSL/TLS denetimi gibi araçları neden kullanmak zorunda olduklarını açıklar.

2.6.5 Bandwidth Management ve QoS Security Implications

Bant genişliği yönetimi ve Hizmet Kalitesi (QoS), ağ performansını ve kullanılabilirliğini optimize etmek için kullanılan temel ağ operasyonel kontrolleridir. Ancak, bu kontroller güvenlik açısından da önemli sonuçlar doğurabilir. Örneğin, bir NGFW üzerindeki DPI, VoIP trafiği gibi kritik uygulamalara öncelik vermek için kullanılabilir ve bu, normal web trafiğinin yavaşlamasına neden olabilir. Güvenlik politikaları, bir ağın güvenlik duruşunu güçlendirmek için bant genişliği ve QoS mekanizmalarıyla entegre edilebilir ve olası DoS saldırılarına karşı koruma sağlayabilir.

Bölüm 3

ENDPOINT VE SİSTEM GÜVENLİĞİ

Giriş

Endpoint güvenliği, modern siber güvenlik stratejilerinin temel taşlarından biridir. Bu bölümde endpoint koruma platformları, sistem güvenliği teknolojileri ve uç nokta güvenlik yönetimi konularını detaylı olarak inceleyeceğiz.

3.1 Endpoint Protection Platform (EPP) Teknolojileri

Endpoint Protection Platform (EPP), bir kuruluşun ağındaki uç noktaları (endpoint) siber tehditlere karşı korumak için tasarlanmış entegre bir güvenlik çözümüdür. Uç noktalar, dizüstü bilgisayarlar, masaüstü bilgisayarlar, sunucular, mobil cihazlar ve IoT cihazları gibi ağa bağlanan herhangi bir cihazı içerir. EPP'ler, geleneksel antivirüs yazılımlarının ötesine geçerek, çok katmanlı bir savunma yaklaşımı sunar. Bu çözümler, proaktif tehdit avlama, davranışsal analiz, makine öğrenimi ve istihbarat entegrasyonu gibi gelişmiş teknikler kullanarak bilinmeyen tehditleri tespit etme yeteneğine sahiptir. EPP'nin temel bileşenleri arasında gelişmiş antivirüs koruması, uygulama kontrolü, cihaz kontrolü ve veri kaybı önleme (DLP) yer almaktadır.

3.1.1 Next-Generation Antivirus (NGAV) ve Machine Learning

Geleneksel antivirüs (AV) çözümleri, bilinen tehditlere karşı bir imza veritabanını kullanarak çalışmaktadır. Bu model, yeni ve bilinmeyen tehditlerin ortaya çıkma hızı karşısında yetersiz kalmıştır. Bu yetersizlik, modern siber tehditlerin sürekli evrilen doğasına yanıt olarak Next-Generation Antivirus (NGAV) teknolojisinin geliştirilmesini tetiklemiştir. NGAV, geleneksel AV'nin imza tabanlı yaklaşımının aksine, makine öğrenmesi (ML), yapay zeka (AI), davranışsal tespit ve istismar önleme gibi ileri teknolojileri kullanarak bilinmeyen tehditleri dahi öngörmekte ve anında engellemektedir. Bu teknolojik değişim, güvenliğin sadece "bilinen kötüye karşı savunma"dan, "anormalin proaktif olarak tespiti"ne doğru evrildiğini göstermektedir.

NGAV'ın temel gücü, AI ve ML algoritmalarını kullanmasından kaynaklanmaktadır. Bu algoritmalar, dosya çalıştırılmadan önce milyonlarca dosya karakteristiğini gerçek zamanlı olarak analiz etmekte ve bir dosyanın kötü amaçlı olup olmadığını belirlemektedir. Bu imzasız teknoloji, hem bilinen hem de bilinmeyen kötü amaçlı yazılımları, uç nokta internete bağlı olmasa bile tespit etme ve engelleme yeteneği sunmaktadır. Ayrıca, NGAV çözümleri, dosyasız (fileless) saldırılar gibi geleneksel antivirüsleri atlatmak için tasarlanmış, sisteme yerleşik araçları (örneğin, PowerShell) kötüye kullanan tehditlere karşı da koruma sağlamaktadır. NGAV çözümleri, genellikle bulut tabanlı bir mimariyle çalıştığından, tek bir hafif ajan (lightweight agent) aracılığıyla uç noktalara dağıtılmakta ve sistem performansı üzerinde minimal etki yaratmaktadır. Bu bulut mimarisi, çözümün saatler içinde kurulmasına olanak tanırken, altyapı bakımı ve imza veritabanı güncelleme yükünü de ortadan kaldırmaktadır. NGAV, sadece daha iyi tespit yetenekleri sunmakla kalmayıp, aynı zamanda daha hızlı yanıt süreleri (MTTR) ve güvenlik personeli için daha az operasyonel yük sağlamaktadır.

3.1.2 Behavioral Analysis ve Heuristic Detection

Davranışsal analiz, bir sistemdeki kullanıcıların ve varlıkların normal aktivitelerini öğrenerek ve bu normal davranıştan sapmaları (anomalileri) tespit ederek çalışmaktadır. Bu yaklaşım, saldırganların geleneksel imza tabanlı sistemleri atlatmak için kullandığı yeni veya gizli yöntemlere karşı özellikle etkilidir. Sezgisel (heuristic) tespit ise, bilinen bir tehdit imzasıyla eşleşmeyen, ancak kötü amaçlı davranış kalıpları sergileyen kodları veya dosyaları proaktif olarak analiz etme yeteneğini ifade etmektedir. Bu iki teknik, özellikle sıfır-gün (zero-day) zafiyetlerini ve bilinmeyen tehditleri belirlemek için hayati öneme sahiptir.

Kullanıcı ve Varlık Davranış Analizi (UEBA), davranışsal analizin ileri bir formudur. UEBA, makine öğrenmesi algoritmalarını kullanarak kullanıcı ve varlık hareketlerinin temel bir modelini oluşturmaktadır. Bu sayede, güvenlik ekipleri anormal oturum açma girişimlerini, yetkisiz erişim denemelerini veya bir kullanıcının normalde erişmediği verilere ani erişimini daha erken aşamada tespit edebilmektedir. Bu durum, hem içeriden gelebilecek tehditlere hem de dış saldırganların kaba kuvvet veya diğer sinsi yöntemlerle sistemlere sızma girişimlerine karşı erken uyarı sağlamaktadır. Bir sistem, tek bir anomaliyi hemen kötü niyetli olarak işaretlemese de, bir saldırı döngüsü boyunca birden fazla anormal davranışın varlığı daha büyük bir riskin göstergesi olabilmektedir. Bu da güvenlik analistlerinin hangi uyarılara öncelik vermesi gerektiğini belirlemesine yardımcı olmaktadır. Sonuç olarak, davranışsal analiz ve UEBA, güvenlik analistlerine eyleme geçirilebilir içgörüler sunarak olaylara daha hızlı ve etkili bir şekilde yanıt verilmesini sağlamaktadır.

3.1.3 Application Control ve Software Restriction Policies

Uygulama kontrolü, bir sistemde hangi yazılımların çalışabileceğini yöneten bir siber güvenlik önlemidir. Bu önlemler, yetkisiz veya kötü amaçlı yazılımların çalıştırılmasını önleyerek güvenliği artırmayı amaçlamaktadır. Temel olarak iki ana strateji bulunmaktadır: **blacklisting (engellenenler listesi)** ve **whitelisting (izin verilenler listesi)**. Blacklisting, bilinen zararlı yazılımların çalışmasını engellerken, whitelisting yalnızca güvenilen ve onaylanmış uygulamaların çalışmasına izin vermektedir. Whitelisting, daha katı bir güvenlik duruşu sunmasına rağmen, yönetimi daha karmaşık olabilmektedir.

Windows ortamında, uygulama kontrolü için kullanılan çeşitli yerleşik mekanizmalar bulunmaktadır. Software Restriction Policies (SRP), Grup İlkesi tabanlı bir özellik olup, yazılımların dosya adı, hash değeri, yayıncısı veya dosya yolu gibi özelliklerine göre çalışmasını kontrol etmektedir. Ancak, Microsoft, Windows 10'un belirli sürümlerinden ve Windows Server 2019'dan itibaren SRP yerine daha modern ve gelişmiş olan **AppLocker** ve **Windows Defender Application Control (WDAC)** çözümlerini kullanmayı önermektedir.

WDAC, sanallaştırma tabanlı güvenlik (VBS) kullanarak çekirdek (kernel) seviyesindeki kodlar dahil olmak üzere, sistemde hangi uygulamaların çalışacağını kısıtlamaktadır. Bu sayede, kötü amaçlı komut dosyaları ve dosyasız saldırılar gibi modern tehditlere karşı daha güçlü bir koruma sağlanmaktadır. WDAC politikaları, PowerShell komutları veya Microsoft tarafından sağlanan özel bir sihirbaz aracılığıyla oluşturulabilmekte ve yönetimsel araçlarla (örneğin, GPO veya MEMCM) uç noktalara dağıtılabilmektedir.

3.1.4 Pratik Yönergeler ve Komut Örnekleri

Windows ortamında uygulama kontrolü politikalarını uygulamak için izlenebilecek adımlar ve örnek komutlar aşağıda sunulmuştur.

1. **AppLocker ile Varsayılan Kural Oluşturma:**
    - Bir yönetici hesabıyla Windows'a oturum açın ve `secpol.msc` komutunu çalıştırarak Yerel Güvenlik Politikası Düzenleyicisi'ni açın.
    - **Application Control Policies** bölümünden **AppLocker**'a gidin.
    - **Executable Rules** öğesine sağ tıklayarak **Create Default Rules** seçeneğini belirleyin. Bu işlem, yöneticilerin ve Program Files/Windows klasörlerindeki programların çalışmasına izin veren üç varsayılan kural oluşturacaktır.
2. **Belirli Bir Uygulamayı Engellemek İçin Kural Oluşturma (Yayıncı Kuralı):**

- Aynı konsolda **Executable Rules** öğesine sağ tıklayın ve **Create New Rule** seçeneğini seçin.
- İzin türü olarak **Deny**'ı seçin ve sonraki adımda ana koşul olarak **Publisher**'ı belirleyin.
- **Browse** düğmesine tıklayarak engellemek istediğiniz uygulamanın (`.exe`) dosyasını seçin. Örneğin, `C:\Program Files\Google\Chrome\Application\chrome.exe`.
- Yayıncı (Publisher) bilgileri otomatik olarak doldurulacaktır. Kuralı tüm Chrome sürümlerine uygulamak için kaydırıcıyı **File name** seviyesine getirin ve kuralı kaydedin. Bu kural, Google Chrome'un tüm sürümlerinin çalışmasını engelleyecektir.

3. **WDAC Politikası Oluşturma ve Dönüştürme:**

   - WDAC Wizard'ı kullanarak bir XML politikası oluşturun.
   - Yönetici haklarıyla PowerShell ISE'yi açın ve oluşturulan XML politikasını dağıtılabilir ikili (.p7b) formata dönüştürmek için aşağıdaki komutu çalıştırın:

     ```
     ConvertFrom-CIPolicy -XmlFilePath "C:\wdac\wdacpolicy.xml" -BinaryFilePath "C:\wdac\siPolicy
     ```

3.1.5 Device Control ve Removable Media Protection

Cihaz kontrolü ve taşınabilir medya koruması, organizasyonun güvenliğini artırmak ve veri sızıntısını önlemek için kritik önlemlerdir. Bu stratejiler, USB bellekler, harici diskler, CD/DVD'ler gibi taşınabilir medya cihazlarının uç noktalarda kullanımını düzenlemeyi ve izlemeyi amaçlamaktadır. Bu kontroller, kötü amaçlı yazılımların bu cihazlar aracılığıyla sisteme sızmasını engellediği gibi, hassas verilerin yetkisiz bir şekilde dışarıya aktarılmasının da önüne geçmektedir.

Windows ortamında, cihaz kontrolü genellikle Grup İlkesi Yönetimi Konsolu (GPMC) veya Microsoft Intune gibi modern yönetim araçları aracılığıyla gerçekleştirilmektedir. GPMC, tüm taşınabilir depolama cihazlarını engellemek için kolayca uygulanabilir bir merkezi politika sunmaktadır. Bu, veri sızıntısı riskini önemli ölçüde azaltmaktadır.

3.1.6 Pratik Yönergeler ve Komut Örnekleri

Bir GPO kullanarak tüm kullanıcılar için USB bellek erişimini engellemek için izlenecek adımlar aşağıda verilmiştir:

1. **Grup İlkesi Yönetimi Konsolunu Açma:** Sunucu Yöneticisi'nden (Server Manager) veya `gpmc.msc` komutuyla GPMC'yi açın.

2. **Yeni Bir GPO Oluşturma:**

   - **Group Policy Objects** öğesine sağ tıklayın ve **New** seçeneğini seçin.
   - GPO'ya açıklayıcı bir ad verin (örneğin, **USB Erişimi Engelleme**) ve **OK**'a tıklayın.

3. **GPO'yu Düzenleme:**

   - Yeni oluşturduğunuz GPO'ya sağ tıklayın ve **Edit** seçeneğini seçerek Grup İlkesi Yönetim Düzenleyicisi'ni açın.
   - Soldaki menüden aşağıdaki yolu izleyin: `Computer Configuration > Policies > Administrative Templates > System > Removable Storage Access`.

4. **Erişimi Engelleme Politikasını Yapılandırma:**

   - Sağdaki bölmede, **All Removable Storage classes: Deny all access** ayarını bulun.
   - Bu ayara çift tıklayın, açılan iletişim kutusunda **Enabled** seçeneğini işaretleyin. Bu, tüm taşınabilir depolama cihazlarına erişimi engelleyecektir.
   - **Apply** ve **OK**'a tıklayarak ayarı kaydedin.

5. **GPO'yu Bağlama ve Uygulama:**

    - GPMC'de, USB erişiminin engelleneceği ilgili organizasyonel birime (OU) sağ tıklayın ve **Link an existing GPO** seçeneğini seçin.
    - Açılan listeden **USB Erişimi Engelleme** GPO'sunu seçin ve **OK**'a tıklayın.
    - Client makinelerde politikayı hemen uygulamak için komut isteminde yönetici olarak aşağıdaki komutu çalıştırın: `gpupdate /force`

Bu adımlarla, yöneticiler ve kullanıcılar için tüm taşınabilir medya cihazlarına erişim etkin bir şekilde engellenmektedir. Güvenlik yöneticileri, daha sonra seçilen cihazlara izin vermek için cihaz örneği kimlikleri (device instance IDs) gibi parametreleri kullanarak istisnalar tanımlayabilmektedir.

3.1.7 Cloud-based vs On-premises EPP Deployment Models

Endpoint Protection Platform (EPP) çözümleri, iki ana dağıtım modeliyle sunulmaktadır: bulut tabanlı ve şirket içi (on-premises). Bu modeller, bir organizasyonun altyapı ihtiyaçlarına, güvenlik gereksinimlerine ve bütçe kısıtlamalarına göre farklı avantajlar ve dezavantajlar sunmaktadır.

Bulut Tabanlı Dağıtım (Cloud-based):

- **Mimari:** EPP çözümü, satıcının bulut altyapısı üzerinde barındırılmakta ve yönetilmektedir. Uç noktalara kurulan hafif ajanlar, satıcının bulut sunucularıyla iletişim kurmaktadır.
- **Avantajları:**
    - **Hızlı Dağıtım:** Çözüm, ek donanım veya altyapı kurulumu gerektirmediği için saatler içinde devreye alınabilmektedir.
    - **Düşük Operasyonel Yük:** Bakım, güncelleme ve ölçeklendirme gibi işlemler satıcı tarafından yönetilmektedir. Bu, organizasyonun IT ekibinin yükünü önemli ölçüde azaltmaktadır.
    - **Esneklik ve Ölçeklenebilirlik:** Çalışan sayısı arttıkça veya azaldıkça çözüm kolayca ölçeklenebilmektedir. Uzaktan çalışanlar ve mobil cihazlar için idealdir.
- **Dezavantajları:**
    - **Veri Egemenliği:** Veri, üçüncü bir tarafın (satıcının) bulutunda barındırıldığı için bazı regülasyonlar veya şirket politikalarıyla uyumsuzluklar ortaya çıkabilmektedir.

Şirket İçi Dağıtım (On-premises):

- **Mimari:** EPP çözümünün tüm bileşenleri (yönetim konsolu, veritabanı, sunucular) organizasyonun kendi veri merkezinde barındırılmaktadır.
- **Avantajları:**
    - **Tam Kontrol:** Organizasyon, tüm verilerin ve altyapının tam kontrolüne sahiptir. Bu, özellikle hassas verilerle çalışan ve katı uyumluluk gereksinimleri olan sektörler için kritik bir faktördür.
    - **Özelleştirme:** Çözüm, organizasyonun özel ihtiyaçlarına göre daha fazla özelleştirilebilmektedir.
- **Dezavantajları:**
    - **Yüksek Maliyet:** Yüksek bir ön yatırım maliyeti gerektirmektedir. Sunucu, depolama ve lisanslama gibi maliyetler söz konusudur. **Yönetim Yükü:** Çözümün bakımı, güncellemeleri ve ölçeklendirilmesi organizasyonun kendi IT ekibi tarafından yapılmalıdır. Bu, ek personel ve kaynak gerektirmektedir.

Sonuç: Bulut tabanlı çözümler, esneklik, hız ve maliyet etkinliği arayan organizasyonlar için idealdir. Şirket içi çözümler ise veri güvenliği ve regülasyon uyumu konusunda tam kontrol sağlamak isteyen büyük kuruluşlar veya kritik altyapıya sahip sektörler için daha uygun olabilmektedir.

3.2 Endpoint Detection and Response (EDR) Solutions

Endpoint Detection and Response (EDR), uç noktalardaki şüpheli aktiviteleri sürekli olarak izleyen, tespit eden ve bu tehditlere müdahale eden bir siber güvenlik teknolojisidir. Geleneksel EPP'ler, bilinen tehditleri engellemeye odaklanırken, EDR, geleneksel savunmaları atlatabilen gelişmiş ve bilinmeyen tehditleri tespit etmek için tasarlanmıştır. EDR, bir ihlalin zaten gerçekleştiği varsayımıyla çalışır ve güvenlik analistlerine bir saldırının kök nedenini anlama ve müdahale etme yeteneği kazandırır.

EDR çözümleri, bir uç nokta cihazındaki faaliyetleri sürekli olarak izleyen bir yazılım ajanı kurarak çalışmaktadır. Bu ajan, süreç yürütme, ağ bağlantıları, dosya değişiklikleri ve diğer sistem aktiviteleri hakkında telemetri verilerini toplamaktadır. Toplanan veriler, bir olay günlüğüne kaydedilmekte ve şüpheli faaliyetler için yapay zeka ve makine öğrenmesi algoritmalarıyla analiz edilmektedir. Bu, EDR'ın yalnızca bilinen tehdit imzalarını taramakla kalmayıp, aynı zamanda normalin dışındaki davranışları da tespit etmesini sağlamaktadır.

EDR'ın en önemli yeteneklerinden biri, **tehdit avcılığı (threat hunting)**dır. Tehdit avcılığı, güvenlik analistlerinin otomatik güvenlik araçları tarafından gözden kaçırılmış olabilecek gizli tehditleri proaktif olarak aradığı, hipotez tabanlı bir yaklaşımdır. Bu proaktif süreç, reaktif tehdit algılama modelinden (bir uyarıya yanıt verme) farklıdır ve kuruluşun BT ortamındaki gizli tehditleri, desenleri veya anomalileri ortaya çıkarmaya yardımcı olmaktadır. Tehdit avcıları, EDR'ın sağladığı zengin telemetri verilerini kullanarak karmaşık sorgular yazabilmekte ve olası saldırgan taktikleri, teknikleri ve prosedürleri (TTP'ler) hakkında hipotezler oluşturabilmektedir.

3.2.1 Pratik Senaryo ve KQL Örnekleri

Senaryo: Bir sistem yöneticisi, bir çalışanın bilgisayarında dosyasız (fileless) bir zararlı yazılımın çalıştığından şüphelenmektedir. Saldırgan, meşru sistem araçlarından biri olan PowerShell'i kötüye kullanıyor olabilir.

Hipotez: Saldırgan, PowerShell'i kullanarak kötü amaçlı bir betiği gizlice çalıştırıyor olabilir. Bu betik, genellikle şifrelenmiş veya base64 kodlanmış bir komut satırı ile başlatılır.

Tehdit Avcılığı Sorgusu (KQL - Kusto Query Language):

Aşağıdaki KQL sorgusu, bu hipotezi doğrulamak için `DeviceProcessEvents` tablosundaki verileri kullanarak şüpheli PowerShell kullanımını aramaktadır:

```
// Base64 kodlu ve şüpheli PowerShell komutlarını tespit etme
DeviceProcessEvents

| where InitiatingProcessFileName in~ ("powershell.exe", "pwsh.exe")
| where ProcessCommandLine has_any ("-e", "-enc", "-encodedcommand")
| where not (ProcessCommandLine has "Set-ExecutionPolicy")
| project Timestamp, DeviceName, InitiatingProcessFileName, ProcessCommandLine, AccountName
```

Bu sorgu, `powershell.exe` veya `pwsh.exe` ile başlayan, `-e`, `-enc` veya `-encodedcommand` gibi şüpheli parametreleri içeren tüm süreçleri listelemektedir. Bu sorgu, güvenlik analistinin, meşru sistem yöneticisi betiklerinden ayrışan potansiyel kötü amaçlı faaliyetleri hızlı bir şekilde belirlemesine olanak tanımaktadır. Analist, bu sonuçları inceleyerek saldırının doğasını ve kapsamını anlayabilmektedir.

3.2.2 Forensic Data Collection ve Analysis

Siber güvenlik olayları, bir olayın kök nedenini, yayılımını ve etkisini anlamak için derinlemesine bir soruşturma gerektirmektedir. EDR, bu süreçte kritik bir rol oynamaktadır çünkü uç noktalardan adli analiz için gerekli olan zengin verileri toplamaktadır. EDR sistemleri, bir tehdit tespit edildiğinde otomatik olarak bellek dökümleri (memory dumps), ağ bağlantı bilgileri, süreç etkinlikleri ve dosya sistemi değişiklikleri gibi kanıtları toplamaktadır. Bu veriler, saldırının bir zaman çizelgesini oluşturmaya yardımcı olmakta ve araştırmacıların saldırının nasıl gerçekleştiğini, nasıl yayıldığını ve hangi varlıkları etkilediğini anlamasını sağlamaktadır.

Adli Veri Toplama ve Analiz Araçları

- **Disk İmajları:** Saldırının kalıcı izlerini (kayıt defteri değişiklikleri, silinen dosyalar) incelemek için disk imajları alınmaktadır. Bu veriler, saldırganın sistemi nasıl ele geçirdiğini ve neler yaptığını ortaya koymaktadır.

- **Bellek Dökümleri:** Dosyasız (fileless) zararlı yazılımlar genellikle bellekte çalışmaktadır. Bu tür tehditleri tespit etmek için EDR, şüpheli süreçlerin anlık bellek dökümlerini alarak analiz etmektedir.

- **Log Yönetimi:** Sistem ve uygulama logları, olayların kronolojik sırasını belirlemek için kullanılmaktadır. EDR, bu logları merkezi bir depoda toplayarak analizi kolaylaştırmaktadır.

Siber adli analiz uzmanları, toplanan verileri analiz etmek için EnCase, FTK (Forensic Toolkit) veya Autopsy gibi özel adli bilişim yazılımları kullanmaktadır. Bu araçlar, verilerin bütünlüğünü koruyarak, hukuki süreçlerde geçerli olabilecek deliller sunmaktadır. EDR'ın bu adli yetenekleri, olay müdahalesini hızlandırmakta ve kuruluşların tehditleri daha hızlı bir şekilde bertaraf etmesine olanak tanımaktadır.

3.2.3 Automated Response ve Remediation Actions

EDR çözümlerinin en önemli avantajlarından biri, bir tehdit tespit edildiğinde insan müdahalesine gerek kalmadan otomatik olarak yanıt verme yeteneğidir. Bu otomasyon, tehditlerin yayılmasını ve sistemler üzerindeki hasarı en aza indirmek için kritik bir rol oynamaktadır. Gelişmiş EDR sistemleri, önceden belirlenmiş kurallara ve tetikleyicilere göre bir dizi otomatik iyileştirme eylemi gerçekleştirebilmektedir.

Otomatik Yanıt Eylemleri Örnekleri:

- **Uç Nokta İzolasyonu:** Şüpheli veya virüs bulaşmış bir uç nokta tespit edildiğinde, EDR o cihazı otomatik olarak ağdan izole edebilmektedir. Bu, saldırının yatay hareket (lateral movement) yapmasını ve diğer cihazlara yayılmasını engellemektedir.

- **Süreci Sonlandırma:** Kötü amaçlı olduğu belirlenen bir süreç veya komut dosyası anında sonlandırılabilmektedir.

- **Dosyaları Karantinaya Alma:** Tespit edilen zararlı dosyalar, sistemin geri kalanına zarar vermesini engellemek için otomatik olarak karantinaya alınmaktadır.

- **Sistem Geri Yükleme:** Bazı EDR çözümleri, saldırının etkilerini geri almak için sistemin bir önceki temiz haline geri yüklenmesini sağlayabilmektedir.

Bu otomatik eylemler, güvenlik ekiplerinin üzerindeki yükü hafifletmekte ve saldırılara karşı daha hızlı bir savunma hattı oluşturmaktadır. EDR'ın bu yetenekleri, bir fidye yazılımı (ransomware) saldırısı gibi hızla yayılan tehditlerde, manuel müdahaleye ihtiyaç duymadan tehdidi anında durdurarak büyük hasarları önleyebilmektedir.

3.2.4 Threat Intelligence Integration ve IOC Matching

Tehdit istihbaratı (Threat Intelligence), bir kuruluşun BT ortamını tehdit edebilecek potansiyel veya mevcut riskler hakkında toplanan, analiz edilen ve eyleme dönüştürülebilen bilgileri ifade etmektedir. Bu bilgiler, bilinen kötü amaçlı IP adreslerini, alan adlarını, dosya hash'lerini ve saldırganların kullandığı taktikleri, teknikleri ve prosedürleri (TTP'ler) içermektedir. EDR çözümleri, proaktif bir güvenlik duruşu sağlamak için bu tehdit istihbaratı beslemeleriyle entegre edilmektedir.

Gösterge Eşleştirme (IOC Matching), tehdit istihbaratı entegrasyonunun temel bir bileşenidir. IOC (Indicator of Compromise), bir güvenlik ihlalinin kanıtı olan dijital verilerdir. EDR, uç noktalardan topladığı verileri (ağ bağlantıları, dosya hash'leri vb.) sürekli olarak tehdit istihbaratı beslemelerindeki IOC'lerle karşılaştırmaktadır. Bir eşleşme bulunduğunda, bu durum otomatik olarak bir uyarıyı tetiklemekte ve güvenlik ekibinin olaya anında yanıt vermesini sağlamaktadır. Bu süreç, bilinen tehditlerin hızlı bir şekilde belirlenmesini sağlamakta ve güvenlik analistlerinin manuel olarak arama yapma yükünü azaltmaktadır. Tehdit istihbaratı, EDR'a yalnızca bir uyarı değil, aynı zamanda saldırının bağlamını ve potansiyel kaynağını anlamak için değerli bilgiler sağlamaktadır.

3.2.5 EDR Data Retention ve Compliance Considerations

EDR, olay analizi, tehdit avcılığı ve adli soruşturmalar için kritik olan büyük miktarda telemetri verisi toplamaktadır. Bu verilerin ne kadar süreyle saklanacağı, depolama maliyetleri, yasal uyumluluk gereksinimleri ve olay müdahale süreçlerinin etkinliği açısından stratejik bir karardır. Veri saklama (data retention) politikaları, hem yasal düzenlemelerle (örneğin, GDPR, HIPAA) uyumun sağlanması hem de bir saldırının kök nedenini belirlemek için yeterli verinin bulunmasını garanti etmek amacıyla oluşturulmaktadır.

Uyum (Compliance) İçin En İyi Uygulamalar

- **Yasal Gereklilikleri Araştırma:** Organizasyonun faaliyet gösterdiği sektördeki ve coğrafi bölgedeki yasal veri saklama gereklilikleri belirlenmelidir. Örneğin, sağlık sektöründe HIPAA düzenlemeleri, hasta verilerinin belirli bir süre saklanmasını zorunlu kılabilmektedir.

- **Veri Sınıflandırması:** Toplanan EDR verileri, hassasiyetine ve iş değerine göre sınıflandırılmalıdır. Daha kritik ve hassas veriler için daha uzun saklama süreleri belirlenirken, daha az önemli veriler için daha kısa süreler uygulanabilmektedir.

- **Otomasyon:** Veri saklama politikalarının uygulanması, otomatik araçlarla yönetilmelidir. Otomatik veri silme veya arşivleme süreçleri, manuel hataları önlemekte ve uyumluluğu tutarlı bir şekilde sağlamaktadır.

- **Güvenli Depolama:** Toplanan veriler, yetkisiz erişime karşı korunmak için şifreleme ve granüler erişim kontrolleriyle güvenli bir ortamda saklanmalıdır.

- **Düzenli Denetim (Auditing):** Veri saklama politikalarına uyum, periyodik olarak denetlenmeli ve raporlanmalıdır. Bu denetimler, hem iç süreçlerin doğru işlediğini doğrulamakta hem de dış denetçilere kanıt sunmaktadır.

3.3 Extended Detection and Response (XDR) Platforms

Extended Detection and Response (XDR), EDR'nin yeteneklerini uç noktaların ötesine taşıyarak, bir kuruluşun tüm güvenlik altyapısını kapsayan daha bütünsel bir tehdit görünürlüğü ve müdahale platformudur. XDR, e-posta, ağ, bulut ve kimlik yönetimi gibi farklı güvenlik katmanlarından gelen verileri bir araya getirir ve bunları tek bir merkezi platformda analiz eder. Bu, güvenlik ekiplerinin daha karmaşık ve dağıtık saldırıları tespit etmesine ve bunlara daha hızlı müdahale etmesine olanak tanır.

XDR, geleneksel EDR'ın sınırlı odağını genişleterek, bir saldırının sadece uç noktadaki belirtilerini değil, aynı zamanda ağdaki anormallikleri, buluttaki yetkisiz erişim denemelerini veya e-posta yoluyla gelen tehditleri de eşzamanlı olarak izleyebilmektedir. Bu genişletilmiş görünürlük, güvenlik stratejisinin temel bir bileşeni haline gelmektedir.

3.3.1 Çoklu Vektör Tehdit Tespiti ve Korelasyon

Modern saldırılar nadiren tek bir saldırı vektörü ile sınırlı kalmaktadır. Saldırganlar, bir uç noktayı ele geçirdikten sonra ağda yanal hareket etmekte ve bulut kaynaklarına erişmeye çalışmaktadır. Geleneksel güvenlik araçları, her bir katman için ayrı çözümler sunduğundan, bir saldırının tüm aşamalarını takip etmek için güvenlik analistlerinin bu farklı sistemlerden gelen uyarıları ve logları manuel olarak birleştirmesi gerekmektedir. Bu parçalı yaklaşım, olay görünürlüğünü azaltmakta ve yanıt sürelerini uzatmaktadır.

XDR, bu sorunu çözmek için farklı güvenlik katmanlarından (uç nokta, ağ, e-posta, bulut) telemetri verilerini toplamakta, normalleştirmekte ve korele etmektedir. Bu korelasyon yeteneği, birbiriyle ilişkili, ancak farklı kaynaklardan gelen yüzlerce uyarıyı tek bir bütünleşik olay altında birleştirmektedir. Bu sayede, güvenlik analistleri bir saldırının tüm hikayesini baştan sona (örneğin, bir phishing e-postasından başlayıp bir sunucuda veri sızıntısıyla biten bir saldırıyı) tek bir konsoldan görebilmektedir. Bu "tam hikaye" yaklaşımı, tehditleri daha doğru bir şekilde belirlemeyi, yanlış pozitifleri azaltmayı ve yanıt eylemlerini hızlandırmayı sağlamaktadır.

3.3.2 Platformlar Arası Görünürlük: Uç Nokta, Ağ, Bulut, E-posta

Dijital altyapıların giderek çeşitlenmesiyle birlikte, bir organizasyonun güvenlik duruşunda ”kör noktalar” oluşabilmektedir. Güvenlik platformları, bu kör noktaları ortadan kaldırmak için bulut, uç nokta, sunucular, e-posta, ağ ve mobil cihazlar gibi tüm vektörlerden zengin veri toplama yetenekleri sağlamaktadır.

XDR platformları, bu platformlar arası görünürlüğü tek bir merkezi arayüzde birleştirerek, güvenlik ekiplerinin tüm dijital ekosistem üzerinde bütünsel bir denetime sahip olmasını sağlamaktadır. Bu bütünleşik görünüm, güvenlik analistlerinin her bir olayın kök nedenini, saldırganın ilk giriş noktasını ve organizasyon genelinde nereye yayıldığını daha hızlı anlamasına olanak tanımaktadır.

3.3.3 Yapay Zeka (AI)/Makine Öğrenmesi (ML) Destekli Analizler ve Otomatik Yanıt

XDR platformlarının kalbinde, geniş veri kümelerini işlemek ve anlamlı güvenlik içgörüleri elde etmek için kullanılan yapay zeka (AI) ve makine öğrenmesi (ML) algoritmaları yer almaktadır. AI/ML, geleneksel kural tabanlı sistemlerin ötesine geçerek, daha önce görülmemiş tehditleri ve karmaşık davranışları tespit edebilmektedir.

AI/ML’in Temel Rolleri:

- **Anomali Tespiti ve Davranışsal Analiz:** ML, kullanıcıların, cihazların ve uygulamaların normal davranışlarına dair bir temel oluşturmaktadır. Bu temelden herhangi bir sapma, potansiyel bir güvenlik olayına işaret eden bir anomali olarak değerlendirilmektedir. Bu, özellikle içeriden gelen tehditleri veya ele geçirilmiş hesapları belirlemek için kritiktir.

- **Korelasyon ve Önceliklendirme:** AI destekli otomasyon, çok sayıda uyarının birleştirilerek daha az sayıda, ancak daha eyleme geçirilebilir olay haline getirilmesini sağlamaktadır. Bu, ”uyarı yorgunluğunu” (alert fatigue) en aza indirmekte ve güvenlik ekiplerinin en yüksek riskli olaylara odaklanmasını kolaylaştırmaktadır.

- **Otomatik Yanıt:** XDR, yüksek güvenilirlikli tehditler tespit ettiğinde, AI algoritmalarını kullanarak otomatik yanıt eylemlerini tetikleyebilmektedir. Örneğin, saldırganın kullandığı bir cihazı veya kimliği ağdan otomatik olarak izole edebilmekte, kötü amaçlı süreçleri durdurabilmektedir. Bu otomasyon, özellikle insan tarafından işletilen ve hızlı yayılan fidye yazılımı saldırılarına karşı yanıt süresini önemli ölçüde kısaltmaktadır.

3.3.4 SOAR Entegrasyonu ve Orkestrasyonu

Güvenlik Orkestrasyonu, Otomasyonu ve Yanıtı (SOAR), güvenlik operasyonlarını otomatikleştirmeyi ve düzenlemeyi amaçlayan bir teknoloji setidir. SOAR, güvenlik ekiplerinin verimliliğini artırmak ve karmaşık olaylara karşı tutarlı bir şekilde yanıt vermek için ”playbook” adı verilen önceden tanımlanmış iş akışlarını kullanmaktadır.

XDR ile SOAR Entegrasyonunun Önemi

XDR, çoklu vektörlerden tehditleri birleştirerek ve korele ederek ”ne olduğunun” kapsamlı bir resmini sunmaktadır. SOAR ise bu resim üzerine inşa edilen ”nasıl yanıt verileceğinin” pratik ve otomatik adımlarını sağlamaktadır. Bu entegrasyon, güvenlik operasyonları için güçlü bir sinerji yaratmaktadır. XDR, bir tehdidi yüksek doğrulukla tespit ettiğinde, SOAR platformu otomatik olarak bir playbook’u tetikleyebilmektedir. Bu, güvenlik analistlerinin manuel müdahaleye gerek kalmadan anında harekete geçmesini sağlamaktadır.

Pratik SOAR Playbook Senaryoları:

- **Phishing Saldırısı Yanıtı:**

  1. **Tetkik:** XDR, bir kullanıcının oltalama (phishing) e-postasına tıkladığını ve kimlik bilgilerini girdiğini tespit eder.
  2. **Otomasyon (Playbook):** SOAR playbook’u otomatik olarak başlatılır:
     - Tehdit istihbaratı entegrasyonuyla kötü amaçlı URL’yi sorgular ve bu URL’yi güvenlik duvarlarında ve web proxy’lerinde anında engeller.
     - Kullanıcının hesabını geçici olarak kilitler veya parola sıfırlama talebinde bulunur.
     - Oltalama e-postasını alan tüm kullanıcıların gelen kutusundan bu e-postayı otomatik olarak siler.

  - Olay yönetimi sisteminde (örneğin, Jira veya ServiceNow) yeni bir olay bileti oluşturur ve ilgili güvenlik analistini bu biletle ilişkilendirir.

- **Yanal Hareket (Lateral Movement) Tespiti ve Yanıtı:**

  1. **Tetkik:** XDR, ele geçirilmiş bir kimlik kullanarak bir saldırganın ağ içinde yanal hareket etmeye çalıştığını tespit eder.
  2. **Otomasyon (Playbook):** SOAR playbook'u otomatik olarak başlatılır:
     - Saldırganın kullandığı hesabı anında devre dışı bırakır.
     - Saldırının başladığı uç noktayı ağdan izole eder.
     - Saldırganın erişmeye çalıştığı kritik sunuculara giden ağ trafiğini engeller.
     - Güvenlik ekibine olayın acil bir durum olduğunu bildiren bir uyarı gönderir.

Bu entegre yaklaşım, güvenlik operasyonlarının etkinliğini artırırken, güvenlik analistlerinin rutin ve tekrarlayan görevlerden kurtularak daha stratejik tehdit avcılığına odaklanmasını sağlamaktadır.

3.3.5 XDR Vendor Ekosistemi ve Entegrasyon Zorlukları

XDR ekosistemi, genellikle iki ana model etrafında şekillenmektedir: yerel (native) XDR ve açık (open) XDR. Yerel XDR platformları, tek bir satıcının kendi ürünlerini (EDR, ağ güvenliği, e-posta güvenliği vb.) bir araya getirmesiyle oluşmaktadır. Bu model, sıkı entegrasyon ve tek bir yönetim konsolu sunduğu için kurulum ve yönetim kolaylığı sağlamaktadır. Ancak, bu yaklaşım organizasyonları tek bir satıcıya bağımlı hale getirmekte ve "vendor lock-in" riskini ortaya çıkarmaktadır.

Öte yandan, açık XDR platformları, farklı satıcılardan gelen güvenlik ürünlerini entegre etmeye olanak tanıyan bir mimariye sahiptir. Bu model, organizasyonlara daha fazla esneklik sunmakta ve mevcut güvenlik yatırımlarını koruma imkanı sağlamaktadır. Ancak, farklı satıcıların ürünlerini entegre etmek, karmaşıklığı artırabilmekte ve beklenmedik sorunlara yol açabilmektedir. Bu nedenle, bir XDR çözümü seçerken organizasyonun mevcut altyapısı, güvenlik olgunluğu ve entegrasyon yetkinlikleri dikkatle değerlendirilmelidir.

3.3.6 EPP, EDR ve XDR Karşılaştırması

Kriterler	**Endpoint Protection Platform (EPP)**	**Endpoint Detection and Response (EDR)**	**Extended Detection and Response (XDR)**
Kapsam	Yalnızca uç nokta cihazları (PC, sunucu).	Yalnızca uç nokta cihazları (PC, sunucu).	Uç nokta, ağ, e-posta, bulut, kimlik.
Temel Yetenek	Önleme odaklı koruma (NGAV, uygulama kontrolü).	Tespit, araştırma ve yanıt.	Çoklu katmanlarda tehdit tespiti, korelasyon ve yanıt.
Hedef	Bilinen ve bilinmeyen tehditlerin uç noktaya ulaşmasını engellemek.	EPP'nin atladığı tehditleri bulmak, kök nedeni belirlemek ve iyileştirmek.	Güvenlik silolarını ortadan kaldırmak, saldırının tüm hikayesini sunmak.
Mimari	Bulut tabanlı veya şirket içi.	Bulut tabanlı veya şirket içi.	Genellikle bulut tabanlı bir hizmet (SaaS).
Yanıt Mekanizması	Otomatik engelleme ve karantinaya alma.	Otomatik karantina-/izolasyon ve manuel iyileştirme.	Otomatik ve orkestrasyonlu yanıt (SOAR ile).
Önemli Notlar	İlk savunma hattı. Modern tehditlere karşı yetersiz kalabilir.	İnsan odaklı tehdit avcılığını destekler. Kapsamı uç nokta ile sınırlıdır.	Saldırının tüm vektörlerini birleştirir. Daha karmaşık ve kapsamlıdır.

3.4 İşletim Sistemi Güvenliği ve Sertleştirme (Hardening)

Bir sistemin sertleştirilmesi (hardening), saldırı yüzeyini en aza indirmek ve güvenlik açıklarını gidermek için sistemin konfigürasyonunu ve ayarlarını güçlendirme sürecidir. Bu süreç, sadece güvenlik yazılımları kurmakla sınırlı kalmamakta, aynı zamanda işletim sistemi seviyesindeki tüm varsayılan ve potansiyel zafiyetlerin giderilmesini içermektedir. CIS Benchmarks gibi endüstri standartları, bu süreç için yol gösterici rol oynamaktadır.

Windows işletim sistemini güvenli hale getirmek, hem yerleşik güvenlik özelliklerini doğru bir şekilde yapılandırmayı hem de sürekli güncellemelerle bilinen zafiyetleri gidermeyi gerektirmektedir. Bir Windows sisteminin sertleştirilmesi için aşağıdaki temel adımlar izlenebilmektedir:

- **Parola Politikaları:** Güçlü parola politikaları, bir sistemin temel güvenlik duruşu için hayati öneme sahiptir. Bu politikalar, parolaların minimum uzunluğunu, karmaşıklık gereksinimlerini, maksimum ve minimum geçerlilik sürelerini ve hesap kilitleme eşiğini belirlemektedir.
- **Kullanıcı Hesap Yönetimi:** Varsayılan yönetici ve misafir hesaplarının yeniden adlandırılması veya devre dışı bırakılması, saldırganların otomatik araçlarla bu hesapları hedef almasını zorlaştırmaktadır.
- **Uzak Erişimin Kısıtlanması:** RDP gibi uzak erişim servisleri, yalnızca gerektiğinde ve güvenli kanallardan (VPN veya jump server) erişime açık olmalıdır.
- **Yerleşik Güvenlik Özellikleri:** Windows Defender Antivirus, SmartScreen ve Exploit Protection gibi yerleşik özelliklerin etkinleştirilmesi, kötü amaçlı yazılımlara ve web tabanlı tehditlere karşı temel koruma sağlamaktadır.
- **PowerShell Sertleştirmesi:** PowerShell, Windows ortamında sıklıkla kötü amaçlı komut dosyalarının çalıştırılması için kullanılmaktadır. PowerShell'in kendisini güvence altına almak için çalıştırma politikaları (Execution Policy) uygulanmakta ve komut betiği loglaması (Script Block Logging) etkinleştirilmektedir.

3.4.1 Pratik PowerShell Sertleştirme Script Örnekleri

Windows sistemleri için güvenlik ayarlarını otomatikleştirmek üzere PowerShell scriptleri kullanılabilmektedir. Aşağıda, temel bir sertleştirme senaryosu için örnekler sunulmuştur.

1. **Parola Politikalarını Yapılandırma:**

```
# Parola geçmişini 24 olarak ayarla
secedit /configure /cfg C:\temp\policy.inf /db C:\temp\policy.sdb /areas SECURITYPOLICY
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit\Settings\Passw

# Minimum parola yaşı 1 gün
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit\Settings\Passw

# Maksimum parola yaşı 60 gün
Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit\Settings\Passw
```

2. **PowerShell Çalıştırma Politikasını Ayarlama:** `Set-ExecutionPolicy` komutu, PowerShell'in scriptleri çalıştırmasına izin veren koşulları kontrol etmektedir. `RemoteSigned` politikası, yerel olarak oluşturulan scriptlerin çalışmasına izin verirken, internetten indirilenlerin güvenilir bir yayıncı tarafından imzalanmasını zorunlu kılmaktadır. `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine`
3. **PowerShell Loglamasını Etkinleştirme:** Script bloklarının ve modül faaliyetlerinin Windows olay günlüğüne kaydedilmesini sağlamak, adli inceleme ve tehdit avcılığı için kritik bir adımdır. Bu ayar GPO üzerinden kolayca yapılandırılabilmektedir.

3.4.2 Linux Security: SELinux, AppArmor, ve Container Security

Linux işletim sistemleri, geleneksel isteğe bağlı erişim kontrolü (DAC) modelinin ötesine geçen, zorunlu erişim kontrolü (MAC) mekanizmalarıyla güçlendirilmektedir. Bu mekanizmalar, kötü amaçlı yazılımların ve yetkisiz kullanıcıların sisteme zarar vermesini engellemektedir. Bu alandaki en yaygın iki teknoloji **SELinux** ve **AppArmor**'dur.

- **SELinux (Security-Enhanced Linux):** ABD Ulusal Güvenlik Ajansı (NSA) tarafından geliştirilen SELinux, "label-based" (etiket tabanlı) bir model kullanmaktadır. Her dosya, süreç ve kullanıcı, güvenlik bağlamı (security context) adı verilen bir etiketle etiketlenmektedir. SELinux politikaları, bu etiketler arasındaki etkileşimleri sıkı bir şekilde tanımlamaktadır. Bu model, son derece granüler ve katı bir kontrol sağladığı için kurumsal ve yüksek güvenlik gereksinimleri olan ortamlar için idealdir. Ancak, öğrenme eğrisi daha diktir ve politika yönetimi karmaşık olabilmektedir.
- **AppArmor (Application Armor):** "Path-based" (yol tabanlı) bir model kullanan AppArmor, uygulamalara, dosya yollarına dayalı profiller atamaktadır. Bu profiller, bir uygulamanın hangi dosyalara ve kaynaklara erişebileceğini net bir şekilde belirlemektedir. AppArmor, SELinux'a göre daha kolay yönetilebilir ve daha basit bir öğrenme eğrisine sahiptir, bu da onu masaüstü sistemler ve daha basit sunucu ortamları için iyi bir seçenek haline getirmektedir.

Konteyner Güvenliği: Docker ve Kubernetes gibi konteyner teknolojileri, uygulamaları izole etse de, aynı işletim sistemi çekirdeğini (kernel) paylaşmaktadırlar. Bu durum, bir güvenlik açığının tüm konteynerli sistemi etkileme potansiyelini taşımaktadır. SELinux ve AppArmor, konteyner güvenliğini artırmak için kullanılmaktadır. Örneğin, Kubernetes, konteynerlerin AppArmor profilleriyle çalışmasını sağlayarak ek bir izolasyon katmanı ekleyebilmektedir.

3.4.3 Pratik Komut Örnekleri

- **SELinux Modunu Kontrol Etme:** `getenforce` komutu, SELinux'un mevcut modunu (`Enforcing`, `Permissive` veya `Disabled`) göstermektedir. `sestatus` komutu, daha detaylı bir durum raporu sunmaktadır.
- **SELinux Modunu Değiştirme:** Modu geçici olarak `Permissive` (yalnızca loglama, engelleme yok) moda geçirmek için: `sudo setenforce 0` Modu tekrar `Enforcing` moda getirmek için: `sudo setenforce 1`. Kalıcı değişiklikler için `/etc/selinux/config` dosyası düzenlenmektedir.
- **AppArmor Durumunu Kontrol Etme:** `sudo aa-status` komutu, etkin ve şikayet modundaki profilleri listelemektedir.

3.4.4 SELinux vs. AppArmor Karşılaştırması

Kriterler	**SELinux (Security-Enhanced Linux)**	**AppArmor (Application Armor)**
Güvenlik Modeli	Etiket tabanlı (Label-based). Her varlığa güvenlik etiketi atar.	Yol tabanlı (Path-based). Kuralları dosya yollarına atar.
Öğrenme Eğrisi	Oldukça dik ve karmaşıktır. Detaylı politika yönetimi gerektirir.	Genellikle daha kolay ve basittir. Profiller daha anlaşılırdır.
Kontrol Seviyesi	Çok granüler ve katıdır. Kullanıcılar, süreçler ve objeler üzerinde detaylı kurallar tanımlar.	Uygulama bazlı daha statik kontrol sağlar. Önceden tanımlanmış davranışlar için daha uygundur.

Esneklik	Dinamik ortamlar için daha esnektir.	Dinamik ortamlarda daha az idealdir.
Varsayılan Dağıtım	Fedora, CentOS/RHEL gibi dağıtımlarda varsayılan olarak gelir.	Ubuntu, SUSE gibi dağıtımlarda varsayılan olarak gelir.
İdeal Kullanım	Yüksek güvenlikli ve karmaşık kurumsal ortamlar, konteyner platformları.	Masaüstü sistemleri ve yönetim kolaylığının öncelikli olduğu sunucular.

3.4.5 macOS Security Architecture ve Enterprise Management

macOS, donanım ve yazılım katmanına entegre edilmiş sağlam bir güvenlik mimarisi sunmaktadır. Bu mimari, kullanıcı verilerini korumaya ve kötü amaçlı yazılımlara karşı sistemi sertleştirmeye odaklanmaktadır. Apple'ın kendi tasarladığı M serisi çipler, bu güvenlik altyapısının temelini oluşturmaktadır.

Temel Güvenlik Özellikleri:

- **Secure Enclave:** Ana işlemciden fiziksel olarak izole edilmiş, hassas kullanıcı verilerini (parola ve biyometrik veriler) koruyan özel bir yardımcı işlemcidir. Bu yonga, ana işlemci saldırıya uğrasa bile hassas verilerin güvenliğini sağlamaktadır.
- **System Integrity Protection (SIP):** SIP, yönetici yetkilerine sahip süreçlerin bile kritik sistem dosyalarını ve klasörlerini değiştirmesini engellemektedir. Bu, çekirdek ve sistem dosyalarına yönelik saldırıları önlemektedir.
- **Gatekeeper:** İnternetten indirilen uygulamaların, bilinen kötü amaçlı yazılımlara karşı Apple tarafından imzalanmış ve onaylanmış olup olmadığını denetlemektedir. Bu, kullanıcıları güvenilmeyen yazılımlara karşı korumaktadır.
- **FileVault 2:** Diskin tamamını şifreleyerek cihaz çalınsa veya kaybolsa bile verilerin güvenliğini sağlamaktadır. Apple Silicon çipler, bu şifreleme işlemini donanım seviyesinde destekleyerek daha da ileri bir güvenlik katmanı sunmaktadır.

Kurumsal Yönetim:

macOS cihazlarının kurumsal ortamda yönetimi, Mobil Cihaz Yönetimi (MDM) çözümleri aracılığıyla gerçekleştirilmektedir. MDM, yöneticilere merkezi bir platformdan cihazlara güvenlik politikaları uygulamalarına, uygulama dağıtmalarına ve envanter yönetimi yapmalarına olanak tanımaktadır. Bu araçlar, macOS'in yerleşik güvenlik özelliklerini tamamlayarak cihazların kuruluş politikalarına uygunluğunu sağlamaktadır.

3.4.6 Mobil İşletim Sistemleri: iOS/Android Güvenlik Modelleri

Mobil cihazlar, kurumsal kaynaklara erişimin ana kapısı haline geldiğinden, mobil işletim sistemlerinin güvenlik modelleri hayati önem taşımaktadır. iOS ve Android, farklı yaklaşımlarla güvenliği sağlamaktadır.

- **iOS Güvenlik Modeli:**
  - **Kapalı Ekosistem (Walled Garden):** iOS, uygulamaların yalnızca Apple'ın App Store'u üzerinden dağıtılmasına izin vermektedir. Her uygulama, yayınlanmadan önce katı bir inceleme sürecinden geçmektedir. Bu yaklaşım, zararlı uygulamaların kullanıcılara ulaşmasını zorlaştırmaktadır.
  - **Sandbox (Korumalı Alan):** Her uygulama, diğer uygulamalardan ve sistemden izole edilmiş kendi "sandbox" ortamında çalışmaktadır. Bu, bir uygulamanın güvenliğinin aşılması durumunda diğer uygulamalara ve hassas verilere zarar vermesini engellemektedir.
  - **İzin Modeli:** Uygulamalar, kameraya, mikrofona veya konumlara erişim gibi hassas verilere erişim için kullanıcıdan açık izin istemektedir.

- **Android Güvenlik Modeli:**
  - **Açık Ekosistem:** Android, uygulamaların Google Play Store'un yanı sıra üçüncü parti mağazalardan da yüklenebilmesine olanak tanımaktadır. Bu esneklik, geliştiriciler için daha fazla özgürlük sağlarken, kullanıcılar için potansiyel riskleri de artırmaktadır.
  - **Güvenlik Katmanları:** Android, sandbox modelini kullanmakta ve her uygulamayı kendi kullanıcı kimliğiyle çalıştırmaktadır. Ayrıca, Google Play Protect gibi hizmetler, kötü amaçlı yazılımlara karşı ek bir koruma sağlamaktadır.
  - **BYOD Politikaları:** Android for Work ve Samsung Knox gibi platformlar, kurumsal uygulamaları ve verileri kişisel verilerden ayırmak için iş profilleri oluşturma yeteneği sunmaktadır.

3.4.7 Sanallaştırma Güvenliği: Hypervisor ve VM İzolasyonu

Sanallaştırma, bir fiziksel sunucu üzerinde birden fazla sanal makinenin (VM) çalıştırılmasına olanak tanıyarak kaynak kullanımını optimize etmektedir. Bu mimarinin güvenliği, VM'ler arasında izolasyonu sağlayan ve fiziksel donanıma erişimi kontrol eden temel bileşen olan **Hypervisor**'a bağlıdır.

Hypervisor Türleri ve Güvenlik:

- **Tip 1 (Bare-metal) Hypervisor:** Fiziksel donanım üzerinde doğrudan çalışmaktadır (örneğin, VMware ESXi, Microsoft Hyper-V). Bu tür hypervisor'lar, bir aracı işletim sistemi katmanına ihtiyaç duymadığı için yüksek düzeyde güvenlik ve performans sağlamaktadır.
- **Tip 2 (Hosted) Hypervisor:** Bir ana işletim sistemi üzerinde bir uygulama olarak çalışmaktadır (örneğin, VMware Workstation, VirtualBox). Kurulumu daha kolay olmasına rağmen, güvenliği ana işletim sisteminin güvenliğine bağlıdır, bu da onu saldırılara karşı daha hassas hale getirebilmektedir.

VM İzolasyon Teknikleri:

- **Donanım Soyutlama:** Hypervisor, her VM'ye sanal CPU, bellek ve depolama gibi sanal donanım kaynakları sağlamaktadır. Bu soyutlama, VM'lerin birbirlerinin kaynaklarına doğrudan erişimini engellemektedir.
- **Bellek ve Ağ Segmentasyonu:** Her VM'ye kendi bellek segmenti ve sanal ağ arayüzü tahsis edilmektedir. Bu segmentasyon, bir VM'nin belleğindeki verilere veya ağ trafiğine diğer VM'lerin erişmesini önlemektedir, bu da yanal hareket gibi saldırı türlerini zorlaştırmaktadır.
- **CPU Planlama:** Hypervisor, her VM için CPU zamanını planlayarak hiçbir VM'nin CPU kaynaklarını tekeline almasını engellemektedir.

Hyper-V Sertleştirme Adımları

Bir Hyper-V ortamının güvenliğini sağlamak için aşağıdaki adımlar izlenmelidir:

1. **Ana Bilgisayarı (Host) Minimal Tutma:** Hyper-V ana bilgisayarı, yalnızca sanallaştırma için kullanılmalı ve dosya sunucusu veya etki alanı denetleyicisi gibi ek roller yüklenmemelidir.
2. **Güvenlik Duvarı Konfigürasyonu:** Yalnızca gerekli portlar açılmalı ve yönetim portlarına (örneğin, RDP) yalnızca kısıtlı ağlardan (VPN gibi) erişime izin verilmelidir.
3. **Anti-malware Kurulumu:** Hyper-V ana bilgisayarında anti-malware yazılımı çalıştırılmalı ve VM dosyaları (VHD/VHDX) tarama istisnalarına eklenmelidir.
4. **En Az Yetki İlkesi:** Hyper-V yönetimine erişim, yalnızca gerekli kullanıcılarla sınırlandırılmalıdır.
5. **VM Güvenliği:** Yeni oluşturulan VM'ler, üretime geçmeden önce tıpkı fiziksel bir sunucu gibi sertleştirilmelidir. Güvenli Önyükleme (Secure Boot) etkinleştirilmeli ve güncel yamalar uygulanmalıdır.

3.5 Sistem Konfigürasyon Yönetimi ve Uyum

Sistem konfigürasyon yönetimi, bir kuruluşun ağındaki tüm sistemlerin ve cihazların yapılandırmalarını standartlaştırma, izleme ve yönetme sürecidir. Bu, güvenlik politikalarının tutarlı bir şekilde uygulanmasını sağlar, yanlış yapılandırmalardan kaynaklanan güvenlik açıklarını azaltır ve operasyonel verimliliği artırır. Uyum (compliance) ise, bir kuruluşun belirli endüstri standartlarına, yasal düzenlemelere ve en iyi uygulamalara uygunluğunu sağlama sürecidir.

3.5.1 Güvenlik Temel Konfigürasyon Standartları (CIS Benchmarks)

CIS Benchmarks, Center for Internet Security (CIS) tarafından geliştirilen ve işletim sistemleri, uygulamalar, ağ cihazları ve bulut altyapıları için konsensüs tabanlı güvenlik konfigürasyon önerileridir. Bu standartlar, bir organizasyonun güvenlik duruşunu geliştirmesine, zafiyetleri azaltmasına ve yasal düzenlemelerle (GDPR, HIPAA gibi) uyumunu sağlamasına yardımcı olmaktadır.

Level 1 vs. Level 2

CIS Benchmarks, organizasyonların ihtiyaçlarına göre farklı seviyelerde uygulanabilmektedir.

- **Level 1 (Temel):** Hizmet ve işlevsellik üzerinde minimal etkiye sahip temel güvenlik gereksinimlerini içermektedir. Sınırlı siber güvenlik kaynaklarına sahip küçük ve orta ölçekli işletmeler için uygundur.
- **Level 2 (Kapsamlı):** Daha kapsamlı ve sıkı güvenlik gereksinimleri sunmaktadır. Uygulaması daha fazla test ve operasyonel değişiklik gerektirebilmektedir. Hassas verileri işleyen, kritik altyapıya sahip veya yüksek siber risk altındaki organizasyonlar (finans, sağlık, kamu) için tasarlanmıştır.

Bir organizasyonun hangi seviyeyi uygulayacağına karar verirken, kendi risk profilini, iş gereksinimlerini ve yasal uyumluluk zorunluluklarını dikkate alması gerekmektedir.

3.5.2 Grup İlkesi Yönetimi ve Güvenlik Şablonları

Grup İlkesi Yönetimi (GPO), Microsoft Active Directory (AD) ortamında, Windows makineleri için merkezi konfigürasyon ve güvenlik politikalarının uygulanmasını sağlayan güçlü bir araçtır. GPO'lar, yöneticilerin tüm alan (domain) veya belirli organizasyonel birim (OU) içindeki bilgisayarlara ve kullanıcılara tek bir merkezden ayarlar uygulamasına olanak tanımaktadır.

Temel Uygulama Alanları:

- **Parola Politikaları:** GPO, parola karmaşıklığı, uzunluğu ve geçerlilik süresi gibi ayarları tüm kullanıcılar için zorunlu kılmaktadır.
- **Erişim Kısıtlamaları:** Yönetim paneli, komut satırı veya belirli uygulamalara erişimi kısıtlamak için kullanılabilmektedir.
- **Güvenlik Şablonları:** Önceden yapılandırılmış güvenlik ayarları setlerini içeren şablonlar, tutarlı ve standartlaştırılmış güvenlik politikalarının uygulanmasını kolaylaştırmaktadır. Örneğin, bir güvenlik şablonu, bir Windows sistemini CIS Benchmarks'e göre sertleştirmek için tüm gerekli ayarları içerebilmektedir.

3.5.3 Konfigürasyon Yönetim Araçları (Ansible, Puppet, Chef)

Konfigürasyon yönetim araçları, altyapıyı "kod olarak" (Infrastructure as Code - IaC) yöneterek, güvenlik politikalarının ve konfigürasyonların otomatik, tekrarlanabilir ve tutarlı bir şekilde uygulanmasını sağlamaktadır. Bu otomasyon, güvenlik ekiplerinin iş yükünü azaltmakta ve manuel hataların neden olduğu güvenlik açıklarını gidermektedir.

- **Ansible: Ajansız (agentless)** bir mimariye sahiptir, yani yönetilen makinelerde bir ajan yazılımı gerektirmez. Python ve YAML tabanlı "playbook" adı verilen dosyaları kullanarak SSH veya WinRM üzerinden uzaktan komutları çalıştırmaktadır. Kurulumu kolaydır ve küçük ile orta ölçekli ortamlar için idealdir.

- **Puppet: Ajan tabanlı (agent-based)** bir mimariye sahiptir. Yönetilen her makinede bir ajan (Puppet Agent) çalışmaktadır ve bu ajan, merkezi bir sunucudan (Puppet Master) konfigürasyonları düzenli aralıklarla çekmektedir.
- **Chef:** Puppet gibi **ajan tabanlı** bir mimari kullanmaktadır. Konfigürasyonlar, Ruby diliyle yazılmış ”cookbook” ve ”recipe” adı verilen dosyalarda tanımlanmaktadır.

3.5.4 Konfigürasyon Yönetim Araçları Karşılaştırması

Kriterler	Ansible	Puppet	Chef
Mimari	Ajansız (Agentless).	Ajan tabanlı (Agent-based).	Ajan tabanlı (Agent-based).
İletişim Modeli	İtme (Push) modeli. Komutları hedef makinelerde çalıştırır.	Çekme (Pull) modeli. Ajanlar sunucudan konfigürasyonları çeker.	Çekme (Pull) modeli. Ajanlar sunucudan konfigürasyonları çeker.
Dil	YAML tabanlı ”Playbook”lar.	Puppet DSL.	Ruby tabanlı ”Cookbook”lar.
Öğrenme Zorluğu	Diğerlerine göre daha kolaydır.	Öğrenme eğrisi daha diktir.	Ruby bilgisi gerektirdiğinden zor olabilir.
İdeal Kullanım	Hızlı dağıtımlar, küçük ortamlar ve ağ cihazı yönetimi.	Karmaşık, büyük kurumsal ortamlar için tutarlılık ve otomasyon.	Özelleştirme ve esneklik gerektiren DevOps ortamları.

3.5.5 Sürekli Uyum İzleme ve Drift Tespiti

Bir sistemin konfigürasyonu, manuel değişiklikler, yamalar veya diğer faktörler nedeniyle zamanla başlangıçtaki güvenli durumundan sapabilmektedir. Bu duruma **konfigürasyon kayması (drift)** denilmektedir ve güvenlik açıkları, istikrarsızlık ve uyumluluk ihlallerine yol açabilmektedir.

Drift Tespiti:

Drift tespiti, bir sistemin gerçek durumunu, kodda (IaC) tanımlanan ”istenilen durumla” sürekli olarak karşılaştıran bir mekanizmadır. Bu, manuel değişikliklerin sistemin güvenliğinden sapıp sapmadığını belirlemeyi sağlamaktadır. GitOps gibi metodolojiler, tüm altyapı konfigürasyonlarını Git versiyon kontrol sisteminde saklayarak ve canlı ortamı bu kodla sürekli senkronize ederek drift'i önlemektedir.

Araçlar ve Teknikler:

- **IaC Araçları:** Terraform, OpenTofu veya AWS CloudFormation gibi altyapı kodu araçları, `terraform plan` gibi komutlarla canlı ortamdaki değişiklikleri tespit edebilmektedir.
- **Konfigürasyon Yönetimi:** Puppet ve Chef InSpec gibi araçlar, düzenli taramalarla sistemlerin belirlenen güvenlik politikalarına (örneğin, CIS Benchmarks) uygunluğunu sürekli denetleyebilmektedir.
- **Otomatik İyileştirme:** Drift tespit edildiğinde, ArgoCD veya Chef Automate gibi çözümler, sistemi otomatik olarak istenilen duruma geri döndürerek (remediation) tutarlılığı sağlamaktadır.

3.5.6 Yama Yönetimi Otomasyonu ve Test Prosedürleri

Yama yönetimi (patch management), işletim sistemlerinde ve uygulamalarda bulunan güvenlik açıklarını düzeltmek için güncellemeleri yönetme sürecidir. Bu süreç, siber güvenlik stratejisinin temel bir bileşenidir çünkü bilinen zafiyetler, saldırganlar için en yaygın giriş noktalarından birini oluşturmaktadır.

Yama yönetimi süreci aşağıdaki adımları içermektedir:

1. **Varlık Yönetimi:** Güncelleme gerektiren tüm donanım ve yazılımların envanteri çıkarılmalıdır.

2. **Zafiyet Değerlendirmesi:** Düzenli zafiyet taramalarıyla, sistemlerdeki mevcut güvenlik açıkları tespit edilmelidir.

3. **Yama İzleme:** Yazılım üreticilerinden gelen yeni yamalar ve güvenlik bültenleri sürekli olarak takip edilmelidir.

4. **Test Etme:** En kritik adım, yamaları üretim ortamına uygulamadan önce bir test ortamında denemektir. Bu ortam, canlı sistemlerin konfigürasyonunu ve kullanılan üçüncü parti uygulamaları taklit etmelidir. Test, yamanın beklenmedik yan etkilere veya uygulama kesintilerine neden olup olmadığını belirlemek için yapılmaktadır.

5. **Otomasyon:** Yama yönetimini otomatikleştiren çözümler, süreci hızlandırmakta ve manuel müdahale ihtiyacını azaltmaktadır. Otomatik dağıtım araçları, yamaların doğru zamanda ve doğru sistemlere uygulanmasını sağlamaktadır.

6. **Doğrulama:** Yama uygulandıktan sonra, zafiyetin gerçekten giderildiğinden emin olmak için ek taramalar yapılmalıdır.

3.6 Mobile Device Management (MDM) ve BYOD Güvenliği

Mobil Cihaz Yönetimi (MDM), bir kuruluşun çalışanlarına ait veya şirket tarafından sağlanan mobil cihazları (akıllı telefonlar, tabletler) merkezi olarak yönetmesini, güvenliğini sağlamasını ve izlemesini sağlayan bir yazılım çözümüdür. Kendi Cihazını Getir (BYOD) politikaları, çalışanların kişisel cihazlarını iş amaçlı kullanmalarına izin verir. Bu, esneklik ve maliyet tasarrufu sağlarken, aynı zamanda önemli güvenlik riskleri de yaratır.

MDM çözümleri, cihaz kaydı, uzaktan silme, şifre politikaları, uygulama yönetimi ve güvenlik güncellemeleri gibi özellikler sunarak, mobil cihazların güvenli bir şekilde yönetilmesini sağlamaktadır. Ayrıca, BYOD politikalarıyla birlikte kullanıldığında, çalışanların kişisel verilerini korurken kurumsal verilerin güvenliğini de garanti altına almaktadır.

3.6.1 Enterprise Mobility Management (EMM) Çözümleri

Kurumsal Mobilite Yönetimi (EMM), organizasyonun mobil cihazlarını, uygulamalarını ve verilerini yönetmek ve güvence altına almak için kullanılan kapsamlı bir çerçevedir. EMM, aşağıdaki bileşenleri bir araya getirmektedir:

- **Mobil Cihaz Yönetimi (MDM):** Cihazın tamamını (envanter, konfigürasyon, güvenlik politikaları) yönetmeye odaklanır.
- **Mobil Uygulama Yönetimi (MAM):** Yalnızca kurumsal uygulamaları ve verileri yönetmeye odaklanır.
- **Mobil İçerik Yönetimi (MCM):** Kurumsal verilere güvenli bir şekilde erişim ve paylaşım sağlar.

EMM, organizasyonlara mobil varlıklarını tek bir platformdan yönetme, hassas verileri gelişmiş şifreleme ve veri kaybı önleme (DLP) önlemleriyle koruma yeteneği sunmaktadır.

3.6.2 Mobil Uygulama Yönetimi (MAM) Stratejileri

MAM, özellikle BYOD ortamlarında çalışan gizliliğini korumak için tasarlanmış bir yaklaşımdır. Cihazın tamamını kontrol etmeden, sadece kurumsal uygulamaları ve bu uygulamalardaki verileri korumaya odaklanmaktadır.

Temel MAM Özellikleri:

- **Uygulama Koruma Politikaları:** Hassas kurumsal verilerin kişisel uygulamalara (örneğin, OneDrive'dan WhatsApp'a) kopyalanmasını veya yapıştırılmasını engellemektedir.
- **Seçici Veri Silme:** Cihaz çalındığında veya bir çalışan işten ayrıldığında, cihazdaki kişisel verileri silmeden yalnızca kurumsal verilerin ve uygulamaların uzaktan silinmesini sağlamaktadır.

- **Erişim Kontrolü:** Kurumsal uygulamalara erişim için bir PIN veya biyometrik doğrulama gereksinimi gibi politikalar uygulanabilmektedir.

Bu stratejiler, çalışanların kişisel cihazlarında kurumsal verilerle güvenli bir şekilde çalışmasına olanak tanımaktadır.

3.6.3 MDM vs. MAM Karşılaştırması

Kriterler	**Mobile Device Management (MDM)**	**Mobile Application Management (MAM)**
Kontrol Seviyesi	Cihazın tamamını yönetir.	Yalnızca belirli uygulamaları ve verilerini yönetir.
Yönetim Odağı	Cihazın kendisi, konfigürasyonu, ağ ayarları, güvenlik politikaları.	Uygulamalar, uygulama verileri ve veri hareket kuralları.
Kullanım Alanı	Şirkete ait cihazlar (Corporate-owned devices).	Çalışana ait cihazlar (BYOD) ve özel uygulamalar.
BYOD Uyumluluğu	Genellikle kullanıcı gizliliği endişesi nedeniyle sınırlı.	Kullanıcının kişisel verilerini koruduğu için BYOD dostudur.
Veri Silme	Tüm cihazı fabrika ayarlarına döndüren tam silme.	Yalnızca kurumsal verileri silen seçici silme.

3.6.4 Konteynerleştirme ve İş Profili Yönetimi

BYOD güvenliği için kritik bir strateji de konteynerleştirme teknolojisidir. Konteynerleştirme, bir cihazda kişisel verileri ve kurumsal verileri birbirinden izole edilmiş sanal alanlarda ayırmaktadır.

İş Profili Yönetimi:

Android Enterprise gibi platformlar, cihaz üzerinde bir ”iş profili” oluşturmaktadır. Bu profil, kişisel uygulamalardan ve verilerden tamamen izole edilmiştir. Tüm kurumsal uygulamalar ve veriler bu profilin içinde yer alır ve kuruluş tarafından yönetilmektedir. Bu yaklaşım, bir yandan çalışan mahremiyetini korurken, diğer yandan hassas kurumsal verilerin güvenliğini garanti altına almaktadır. Bu izolasyon, veri sızıntısını ve zararlı yazılımların kurumsal alana bulaşmasını engellemektedir.

3.6.5 Mobil Tehdit Savunması (MTD) Entegrasyonu

Mobil Tehdit Savunması (MTD), mobil cihazlara yönelik oltalama (phishing) saldırıları, kötü amaçlı yazılımlar ve sıfır-gün tehditleri gibi dinamik siber tehditlere karşı gerçek zamanlı koruma sağlayan bir güvenlik stratejisidir. MTD çözümleri, mobil cihazların, ağ bağlantılarının ve yüklü uygulamaların güvenlik seviyesini sürekli olarak analiz etmektedir.

MTD'nin en önemli özelliklerinden biri, MDM çözümleriyle olan entegrasyonudur. MTD, bir cihazın jailbreak yapıldığını veya şüpheli bir ağa bağlı olduğunu tespit ettiğinde, MDM'ye bu durumu bildirmektedir. MDM, bu bilgilere dayanarak cihazın kurumsal kaynaklara erişimini engelleyebilmekte veya önceden tanımlanmış diğer yanıt eylemlerini tetikleyebilmektedir. Bu entegrasyon, mobil cihazlar için hem yönetim (MDM) hem de savunma (MTD) yeteneklerini birleştiren katmanlı bir güvenlik modeli oluşturmaktadır.

3.6.6 Uzaktan Silme (Remote Wipe) ve Veri Sızıntısı Önleme

Uzaktan silme, bir mobil cihaz çalındığında, kaybolduğunda veya bir çalışanın işten ayrılması durumunda, cihazdaki hassas verilerin uzaktan silinmesini sağlayan kritik bir güvenlik özelliğidir. Bu özellik, veri sızıntısını (DLP - Data

Leakage Prevention) önlemek için son savunma hattını temsil etmektedir.

Kullanım Senaryoları:

- **Cihaz Hırsızlığı veya Kaybı:** Bir cihazın yanlış ellere düşmesi durumunda, kurumsal verilerin yetkisiz erişime karşı korunmasını sağlamaktadır.
- **Çalışan Ayrılışı:** Şirketten ayrılan bir çalışanın cihazında kalan kurumsal verilerin silinmesi için kullanılmaktadır.
- **İç Tehditler:** Kötü niyetli bir çalışanın şirkete ait verileri ele geçirmesi durumunda, verilerin uzaktan silinmesini sağlamaktadır.

Seçici (Selective) vs. Tam (Full) Silme:

- **Tam Silme (Full Wipe):** Cihazdaki tüm veriyi kalıcı olarak silerek cihazı fabrika ayarlarına döndürmektedir.
- **Seçici Silme (Selective Wipe):** BYOD ortamları için idealdir. Yalnızca kurumsal verileri ve uygulamaları silmekte, kullanıcının kişisel fotoğraflarını, mesajlarını veya diğer verilerini korumaktadır.

Uzaktan silme eylemleri, MDM çözümleri aracılığıyla tek bir yönetim konsolundan başlatılabilmektedir. Otomatik olarak tetiklenebilen kurallar (örneğin, belirli bir süre boyunca cihazın bağlanmaması veya birden fazla başarısız parola denemesi) ile süreçler daha da otomatikleştirilebilmektedir. Ancak, bu işlemin başarılı olması için cihazın açık olması ve internete bağlı olması gerekmektedir.

Bölüm 4

UYGULAMA GÜVENLİĞİ VE DEVSECOPS

Giriş

Uygulama güvenliği ve DevSecOps, modern yazılım geliştirme süreçlerinde güvenliğin entegrasyonunu sağlayan kritik yaklaşımlardır. Bu bölümde güvenli yazılım geliştirme, kod analizi teknikleri ve DevSecOps uygulamalarını ele alacağız.

4.1 Güvenli Yazılım Geliştirme Yaşam Döngüsü (SSDLC)

Güvenli Yazılım Geliştirme Yaşam Döngüsü (SSDLC), yazılım geliştirme sürecinin her aşamasına güvenlik kontrollerinin ve uygulamalarının entegre edilmesini sağlayan bir metodolojidir. Geleneksel yazılım geliştirme modelleri, güvenliği genellikle son aşamada, yani test veya dağıtım aşamasında ele alırken, SSDLC bu yaklaşımın yetersiz olduğunu savunur. Güvenlik, geliştirme sürecinin başlangıcından itibaren bir öncelik olarak kabul edilmelidir. Bu proaktif yaklaşım, güvenlik açıklarının erken tespit edilmesini ve düzeltilmesini sağlayarak, hem maliyeti düşürür hem de genel yazılım kalitesini artırır.

4.1.1 Güvenliğin Tasarımda Olması (Security by Design) ve Sola Kaydırma (Shift-Left) Güvenlik Yaklaşımları

Güvenliğin tasarımda olması prensibi, güvenlik hususlarının bir sistemin temel mimarisine en başından itibaren dahil edilmesini öngörür. Bu yaklaşımın temel amacı, daha sonra tüm sistemi yeniden tasarlamayı gerektirecek ciddi tasarım kusurlarının oluşmasını önlemektir. Güvenlik, geliştirme sürecinin bir eklentisi veya son adımı olarak değil, projenin en temel yapı taşlarından biri olarak ele alınır. Bu felsefeyi destekleyen anahtar metodoloji ise ”Sola Kaydırma” (Shift-Left) güvenlik yaklaşımıdır. Sola kaydırma, güvenlik pratiklerinin geleneksel olarak geliştirme sürecinin son aşamalarına bırakılması yerine, planlama, tasarım ve kodlama gibi en erken safhalara taşınmasını ifade eder. Bu yaklaşımın temel amacı, güvenlik açıklarını üretime girmeden önce tespit edip düzeltmektir, çünkü bu, düzeltme maliyetini ve çabasını önemli ölçüde azaltır. Sola kaydırma, sadece güvenlik araçlarının geliştirme boru hattına eklenmesiyle sınırlı değildir. Bu yaklaşımın başarısı, aynı zamanda derin bir kültürel dönüşümü gerektirir. Geleneksel ”kodu duvardan atma” (toss it over the wall) modelinde, geliştiriciler kodu yazıp güvenlik ekibine teslim eder, güvenlik testleri ise ayrı bir silo içinde gerçekleştirilir. Bu durum, güvenlik testlerinin geliştirme hızını yavaşlatması ve son dakika sorunlarına yol açmasıyla sonuçlanabilir. Sola kaydırma ise geliştiricilerin güvenliği kendi sorumlulukları olarak benimsemesini, güvenlik uzmanlarının da bu süreci kolaylaştırıcı bir rol üstlenmesini gerektirir. Geliştiricilerin iş akışına entegre olan, soyut ve hızlı geri bildirim sağlayan araçlar ve politikalar, bu yaklaşımın benimsenmesini hızlandırır. Bu sayede, geliştiricilerin güvenlik farkındalığı artar ve daha güvenli kod yazma alışkanlıkları kazanılır, bu da genel olarak daha güvenli yazılımların ortaya çıkmasına yol açan bir döngü oluşturur. Bu felsefenin etkileri, yalnızca uygulama katmanıyla sınırlı değildir. Bulut yerlisi (cloud-native) mimarilerin yaygınlaşmasıyla birlikte, altyapının da kod olarak (IaC) yönetilmesi mümkün hale gelmiştir. Bu durum, altyapı güvenliğinin de sola kaydırılabileceği yeni bir alan yaratmıştır. IaC taraması, hatalı yapılandırılmış ağ portları veya sabit kodlanmış kimlik bilgileri gibi riskleri, altyapı canlı bir or-

tamda saldırı yüzeyi oluşturmadan önce tespit etmeyi sağlar. Bu tür proaktif ve önleyici güvenlik kontrolleri, sistemin genel güvenlik duruşunu güçlendirir.

4.1.2 Tehdit Modelleme Metodolojileri

Tehdit modelleme, bir sistemdeki olası tehditleri, zafiyetleri ve açıklıkları sistematik olarak belirleme, listeleme ve önceliklendirme sürecidir. Bu süreç, yazılımın daha tasarlanma aşamasındayken proaktif olarak güvenlik önlemlerinin alınmasını sağlar. Farklı metodolojiler, tehditleri farklı açılardan ele alarak kuruluşların ihtiyaçlarına özel çözümler sunar.

- **STRIDE Metodolojisi:** Microsoft tarafından geliştirilen STRIDE, tehditleri altı temel kategoriye ayırarak bir çerçeve sunar:
  - **S**poofing (Kimliğe Bürünme): Kimlik doğrulama kontrollerini hedef alır.
  - **T**ampering (Kurcalama): Veri bütünlüğünü tehlikeye atar.
  - **R**epudiation (Red): İzlenebilirlik ve loglama süreçlerini engellemeyi amaçlar.
  - **I**nformation Disclosure (Bilgi Açığa Çıkarma): Gizli bilgilerin ifşa edilmesini içerir.
  - **D**enial of Service (Hizmet Reddi): Erişilebilirliği engeller.
  - **E**levation of Privilege (Yetki Yükseltme): Yetkilendirme kontrollerini aşmayı amaçlar.

  STRIDE, genellikle geliştirme sürecinin tasarım aşamasında kullanılır ve sistem mimarisini analiz ederek potansiyel tehdit kategorilerini belirlemeye yardımcı olur.

- **DREAD Metodolojisi:** DREAD, bir tehdidin ciddiyetini değerlendirmek için kullanılan nicel bir modeldir. Genellikle STRIDE ile birlikte kullanılır; STRIDE tehditleri belirlerken, DREAD bunların önceliklendirilmesine yardımcı olur. Her bir tehdit, beş kritere göre 1'den 10'a kadar bir puanla derecelendirilir:
  - **D**amage (Hasar): Saldırının potansiyel etkisi.
  - **R**eproducibility (Tekrarlanabilirlik): Saldırının ne kadar kolay tekrarlanabildiği.
  - **E**xploitability (İstismar Edilebilirlik): Zafiyetin ne kadar kolay istismar edilebildiği.
  - **A**ffected Users (Etkilenen Kullanıcılar): Etkilenen kullanıcı sayısı.
  - **D**iscoverability (Keşfedilebilirlik): Zafiyetin ne kadar kolay keşfedilebildiği.

  Bu puanların ortalaması alınarak her bir tehdide bir önem derecesi atanır ve böylece düzeltme stratejileri önceliklendirilir.

- **PASTA Metodolojisi (Process for Attack Simulation and Threat Analysis):** PASTA, yedi adımlı, saldırgan odaklı ve risk merkezli kapsamlı bir metodolojidir. Bu yaklaşım, tehditleri iş hedefleriyle ilişkilendirerek olgun güvenlik programlarına sahip büyük kuruluşlar için idealdir.

Tehdit modelleme çerçeveleri birbirini dışlayan yaklaşımlar değildir, aksine birbirini tamamlayan araçlar olarak kullanılabilir. STRIDE gibi bir model, potansiyel tehditlerin varlığını sistematik bir şekilde ortaya çıkarırken, DREAD bu tehditlerin ne kadar acil bir risk oluşturduğunu belirlemek için kullanılır. Bu sinerjik kullanım, sınırlı kaynakların en kritik güvenlik sorunlarına yönlendirilmesini sağlar. Ayrıca, PASTA'nın iş süreçlerini ve teknik tehditleri bir araya getirme yaklaşımı, tehdit modellemenin sadece teknik bir egzersiz değil, aynı zamanda stratejik bir iş faaliyeti olduğunu gösterir. **Uygulamalı Senaryo:** Bir e-ticaret uygulamasının kullanıcı profili güncelleme bileşenine yönelik tehdit modellemesi şu adımlarla gerçekleştirilebilir:

- Veri Akış Şeması (DFD) oluşturularak, kullanıcının profiline ilişkin verilerin nasıl hareket ettiği görselleştirilir.
- STRIDE kullanılarak, her bir veri akışı ve işlem için olası tehditler listelenir. Örneğin, `HTTP POST` isteğiyle gönderilen kullanıcı verileri için "Kurcalama" tehdidi incelenir.
- Tespit edilen her tehdit, DREAD kriterlerine göre puanlanarak en yüksek riske sahip olanlar önceliklendirilir. Bu sayede, güvenlik ekipleri en acil sorunlara odaklanabilir.

4.1.3 Güvenli Kodlama Standartları ve En İyi Uygulamalar

Güvenli kodlama, zafiyetlerin kaynak kod seviyesinde oluşmasını engellemek için benimsenen bir dizi disiplinli yaklaşımdır. Uygulamaların güvenliğini temelden sağlamlaştırmak için belirli prensiplere uyulması gerekir.

- **Varsayılan Olarak Reddetme (Default Deny):** Erişim kararlarının, özellikle yetkilendirme ve erişim kontrollerinde, varsayılan olarak reddetme üzerine inşa edilmesi esastır. Bu, yalnızca açıkça izin verilen koşullar altında erişime izin verilmesini sağlar.
- **En Az Yetkilendirme (Principle of Least Privilege):** Her süreç veya kullanıcı, görevini tamamlamak için gerekli olan en az yetki setiyle çalışmalıdır. Bu yaklaşım, bir saldırganın sistemde yetki yükseltme yapma olasılığını azaltır.
- **Girdi Doğrulama:** Tüm güvenilmeyen veri kaynaklarından (kullanıcı girdisi, API yanıtları, çevre değişkenleri) gelen verilerin titizlikle doğrulanması, zafiyetlerin büyük bir kısmını önleyebilir.
- **Merkezi Güvenlik Kontrolleri:** Kimlik doğrulama, yetkilendirme ve girdi doğrulama gibi güvenlik işlevlerinin, tüm uygulama genelinde tek ve merkezi bir rutin aracılığıyla yapılması tavsiye edilir. Bu, güvenlik politikalarının tutarlı bir şekilde uygulanmasını sağlar.
- **Savunmada Derinlik (Defense in Depth):** Bir savunma katmanının başarısız olması durumunda diğerlerinin koruma sağlaması için çoklu savunma stratejilerinin uygulanması gerekir. Bu prensip, güvenli programlama tekniklerinin güvenli çalışma zamanı ortamlarıyla birleştirilmesi gibi yöntemlerle desteklenebilir.

4.1.4 Güvenlik Kod İncelemesi ve Statik Analiz (SAST) Entegrasyonu

Statik Uygulama Güvenlik Testi (SAST), uygulamanın kaynak kodunu, bayt kodunu veya ikili kodunu çalıştırmadan analiz eden bir ”beyaz kutu” test metodolojisidir. SAST, geliştirme yaşam döngüsünün en erken aşamalarında zafiyetleri (örn. SQL enjeksiyonları, arabellek taşmaları) tespit etme yeteneğine sahiptir. Bir SAST aracı, kodu ayrıştırarak soyut bir sözdizimi ağacı (AST) oluşturur. Bu ağaç, kodun yapısını temsil eder ve aracın veri akışını izlemesine olanak tanır. Analiz, OWASP Top 10 ve CWE/SANS Top 25 gibi bilinen güvenlik standartlarına ve kurallarına göre yapılır. SAST, entegre geliştirme ortamlarına (IDE), versiyon kontrol sistemlerine ve CI/CD hatlarına entegre edilebilir. Bu sayede geliştiriciler, kodlarını kaydederken veya birleştirme isteği (pull request) oluştururken anında geri bildirim alabilirler. SAST'ın en büyük zorluklarından biri, yanlış pozitif (false positives) uyarı sayısının yüksek olabilmesidir. Ayrıca, SAST araçları kodda temsil edilmeyen, örneğin sunucu yapılandırması gibi sorunları tespit edemez.

4.1.5 CI/CD Pipeline'larında Güvenlik Testi Entegrasyonu

Modern yazılım geliştirme süreçlerinde, güvenlik testleri CI/CD (Sürekli Entegrasyon/Sürekli Dağıtım) boru hatlarına entegre edilerek otomatize edilir ve sürekli hale getirilir.

- **SAST:** SAST, CI/CD boru hattına `pre-commit` hook'ları veya build aşaması gibi en erken noktalarda entegre edilir. Örneğin, bir geliştirici kodunu göndermeden önce yerel bir tarama yaparak basit hataları hızlıca düzeltebilir.
- **DAST (Dynamic Application Security Testing):** DAST, çalışan bir uygulama üzerinde ”siyah kutu” testi yaparak gerçek bir saldırıyı simüle eder. Bu, SAST'ın gözden kaçırabileceği iş mantığı hataları, kimlik doğrulama bypass'ları veya yapılandırma zafiyetleri gibi çalışma zamanı (runtime) sorunlarını ortaya çıkarır. DAST taramaları genellikle boru hattının sonlarına doğru, staging veya QA ortamında otomatik olarak çalıştırılır.
- **IAST (Interactive Application Security Testing):** IAST, SAST ve DAST'ın hibrit birleşimidir. Uygulama çalışırken içeriden analiz yaparak, zafiyetin tam olarak hangi kod satırında olduğunu saptar ve düşük yanlış pozitif oranıyla sonuç üretir. IAST, CI/CD hatlarına sorunsuz entegre olabilen tek dinamik test tekniği olarak öne çıkar ve geliştiricilere hızlı ve bağlamsal geri bildirim sağlar.

Tablo 4.1.5: Uygulama Güvenlik Testi (AST) Metodolojileri Karşılaştırması

Metodoloji	Tanım	SDLC Konumu	Yöntem	Kapsadığı Zafiyetler	Avantajları	Zorlukları
SAST	Statik Uygulama Güvenlik Testi	Plan, Kod, Build	Beyaz Kutu, Kod Analizi	SQLi, XSS, Buffer Overflow, Hard-coded Secrets	Erken ve sürekli geri bildirim, geniş kod kapsama	Yüksek yanlış pozitif, çalışma zamanı hatalarını bulamaz
DAST	Dinamik Uygulama Güvenlik Testi	Test, Dağıtım	Siyah Kutu, Çalışan Uygulama Simülasyonu	SQLi, XSS, Kimlik Doğrulama Hataları, İş Mantığı Hataları	Gerçek dünya saldırılarını simüle eder, dil bağımsızdır	Kaynak kodu bilmez, kapsam sınırlıdır, yanlış pozitif olabilir
IAST	Etkileşimli Uygulama Güvenlik Testi	Test, QA	Hibrit (Beyaz + Siyah Kutu)	SQLi, XSS, Veri Akışı Zafiyetleri	Düşük yanlış pozitif, zafiyetin kod satırını belirler, CI/CD'ye entegre olur	Karmaşık kurulum, performans yükü yaratabilir
RASP	Çalışma Zamanı Uygulama Kendi Kendini Koruma	Çalışma (Runtime)	Uygulama İçi İzleme	Bilinen ve sıfırgün saldırıları (SQLi, XSS)	Gerçek zamanlı koruma, çevre savunmasını aşan saldırıları engeller	Performansı etkileyebilir, hedef cihazda çalışması gerekir
SCA	Yazılım	Plan, Kod, Build	Bağımlılık Taraması	Üçüncü	Açık	Sadece

4.2 Web Uygulama Güvenliği ve OWASP Çerçevesi

Web uygulama güvenliği, modern siber güvenlik stratejilerinin temel taşlarından biridir. Open Web Application Security Project (OWASP), web uygulamalarının güvenliğini artırmak için standartlar, araçlar ve en iyi uygulamalar sunan, kar amacı gütmeyen bir kuruluştur. OWASP'ın kaynak havuzu şunları içerir:

- **OWASP Top 10:** Web uygulamalarına yönelik en kritik on güvenlik riskini sıralayan ve düzenli olarak güncellenen bir liste. Bu liste, geliştiricilerin ve güvenlik profesyonellerinin en yaygın saldırı vektörlerine karşı savunma stratejileri oluşturmasına yardımcı olur.
- **OWASP Testing Guide:** Kapsamlı bir web uygulama güvenlik testi rehberi. Test süreçlerini, metodolojileri ve araçları detaylıca açıklar.
- **OWASP Code Review Guide:** Güvenli kod inceleme pratikleri için yol gösterici bir kaynak. Geliştiricilere ve güvenlik uzmanlarına yönelik rehberlik sağlar.
- **OWASP Software Assurance Maturity Model (SAMM):** Organizasyonların yazılım güvenliği programlarını değerlendirmeleri ve geliştirmeleri için bir çerçeve sunar.
- **OWASP Security Knowledge Framework (SKF):** Güvenli yazılım geliştirme için bir eğitim platformu ve kılavuz.
- **OWASP DevSecOps Maturity Model:** DevSecOps olgunluğunu değerlendirmek ve geliştirmek için bir çerçeve sunar.
- **OWASP Mobile Security Testing Guide (MASTG):** Mobil uygulama güvenlik testi için kapsamlı bir rehber.
- **OWASP Web Security Testing Guide (WSTG):** Web uygulaması güvenlik testi için metodolojiler ve örnekler sunar.
- **OWASP API Security Top 10:** API güvenliğindeki en kritik riskleri tanımlar.

OWASP, düzenli olarak güncellenen, gönüllü topluluğun katkılarıyla gelişen ve açık kaynak prensiplerini benimseyen bir projedir. Kuruluşların güvenlik olgunluğunu artırmak için birçok farklı araç ve çerçeve sunar.

4.2.1 OWASP Top 10 Güvenlik Açıkları ve Azaltma Stratejileri

OWASP Top 10, web uygulamaları için en kritik on riskin konsensüs listesidir. Bu liste, güvenlik uzmanlarının ve geliştiricilerin en önemli zafiyetlere odaklanmasına yardımcı olur.

- **A01:2021 Bozuk Erişim Kontrolü (Broken Access Control):** 2021 listesinde birinci sıraya yükselen bu zafiyet, bir saldırganın yetkisi olmayan kullanıcı hesaplarına veya işlevlere erişimini sağlar. Örneğin, bir URL'deki kullanıcı kimliğini temsil eden birincil anahtarın değiştirilmesiyle, bir saldırgan başka bir kullanıcının hesabını görüntüleyebilir veya düzenleyebilir. Bu tür zafiyetlerin giderilmesi için IAST araçları kullanılabilir ve mimari tasarımda güven sınırları oluşturulması gerekebilir.
- **A02:2021 Kriptografik Başarısızlıklar (Cryptographic Failures):** Daha önce "Hassas Veri Açığa Çıkması" olarak bilinen bu risk, parolalar, kimlik numaraları gibi hassas verilerin depolanması veya iletilmesi sırasında yetersiz şekilde korunmasıyla ilgilidir. Zayıf şifreleme algoritmaları veya sabit kodlanmış kriptografik anahtarlar gibi kök nedenlere odaklanır. SAST ve SCA araçları, zayıf şifreleme gücü ve riskli algoritmaları tarayarak bu riskleri azaltmaya yardımcı olurken, IAST sürekli izleme sağlar.
- **A03:2021 Enjeksiyon (Injection):** Listede üçüncü sırada yer alan enjeksiyon, güvensiz verilerin komut veya sorgu olarak yorumlanması sonucu meydana gelir. En yaygın türleri SQL enjeksiyonu ve artık bu kategoriye dahil edilen XSS'tir.

4.2.2 Girdi Doğrulama, Çıktı Kodlama ve Parametreli Sorgular

Bu üç teknik, enjeksiyon saldırılarına karşı birincil savunma mekanizmalarıdır.

- **Girdi Doğrulama (Input Validation):** Güvenilmeyen kaynaklardan gelen verinin, beklenen format, tip ve uzunluk gibi önceden tanımlanmış kriterlere uyduğunu doğrular. En güvenli yaklaşım, yalnızca kabul edilebilir karakter veya formatları tanımlayan bir ”beyaz liste” (allowlist) kullanmaktır, ”kara liste” (denylist) ise eksik olabilir ve yeni saldırı vektörlerine karşı savunmasız kalabilir. Sunucu tarafında uygulanması zorunludur, çünkü istemci tarafı doğrulama kolayca atlatılabilir.
- **Çıktı Kodlama (Output Encoding):** Bir uygulamanın, kullanıcıdan gelen veriyi tarayıcı veya başka bir yorumlayıcıya göndermeden önce potansiyel olarak zararlı karakterleri güvenli bir formata dönüştürmesidir. Bu, karakterlerin kod olarak değil, salt metin olarak görüntülenmesini sağlar ve XSS saldırılarını önlemede kritik bir rol oynar.
- **Parametreli Sorgular (Parameterized Queries):** SQL enjeksiyonuna karşı en etkili savunma yöntemidir. Bu teknik, SQL kod yapısını kullanıcıdan gelen veriden ayırır, böylece girdi yalnızca bir veri değeri olarak işlenir ve kötü amaçlı komutların veritabanı tarafından yürütülmesini engeller.

4.2.3 Kimlik Doğrulama ve Oturum Yönetimi Güvenliği

Kimlik doğrulama, kullanıcıyı doğrulamayı, oturum yönetimi ise bu doğrulama sonrası kullanıcı oturumunu güvenli bir şekilde sürdürmeyi amaçlar.

- **Oturum Güvenlik Açıkları:** İlgili riskler arasında bir saldırganın geçerli bir oturum belirtecini ele geçirdiği **Oturum Korsanlığı (Session Hijacking)** ve bir kullanıcıyı önceden belirlenmiş bir oturum kimliği kullanmaya zorladığı **Oturum Sabitleme (Session Fixation)** yer alır.
- **En İyi Uygulamalar:** Oturum kimlikleri, yeterli rastgeleliğe (en az 64 bit entropi) sahip bir CSPRNG (Kriptografik Olarak Güvenli Sözde Rastgele Sayı Üreteci) ile oluşturulmalıdır. Oturum çerezlerinin `HttpOnly` bayrağı ile korunması, JavaScript'in çereze erişimini engeller ve XSS saldırılarına karşı bir savunma katmanı sağlar. Oturumların, hem kullanıcı çıkış yaptığında hem de bir süre işlem yapılmadığında güvenli bir şekilde sonlandırılması gereklidir.

4.2.4 Cross-Site Scripting (XSS) ve Cross-Site Request Forgery (CSRF)

XSS ve CSRF, web uygulamalarının kullanıcılarına duyduğu ”güven”i istismar eden iki farklı ancak birbiriyle ilişkili saldırı türüdür.

- **XSS (Siteler Arası Komut Dosyası Çalıştırma):** Saldırgan, bir web uygulamasına kötü amaçlı bir komut dosyası enjekte ederek bu betiğin diğer kullanıcıların tarayıcılarında çalışmasını sağlar. Bu, uygulamanın kullanıcı girdisini yeterince filtrelememesinden kaynaklanır. Azaltım stratejileri, girdi doğrulaması, çıktı kodlaması ve DOMPurify gibi kütüphanelerle HTML sanitizasyonunu içerir.
- **CSRF (Siteler Arası İstek Sahteciliği):** Bu saldırıda, bir saldırgan bir kullanıcıyı, rızası olmadan bir eylem gerçekleştirmesi için sahte bir istek göndermesi için kandırır. Saldırganın hedefi, uygulamanın kullanıcının oturum kimliğine ve dolayısıyla isteğin meşruiyetine olan güvenini istismar etmektir. En yaygın savunma, her istek için benzersiz ve öngörülemez bir **CSRF belirteci** (token) kullanmaktır.

XSS ve CSRF, farklı güvenlik zafiyetleri olmasına rağmen, birbirlerinin savunmasını etkileyebilir. Bir saldırgan XSS ile başarılı olursa, `HttpOnly` bayrağı olmayan oturum çerezlerini ele geçirerek oturum korsanlığına zemin hazırlayabilir. Ancak, CSRF belirteçleri gibi bir kontrol, XSS saldırılarına karşı ikincil bir savunma katmanı sağlayabilir. Bir saldırgan XSS ile kötü amaçlı bir betik enjekte etse bile, bu betik, istek için geçerli olan benzersiz CSRF belirtecine erişemez ve bu da saldırının engellenmesine yardımcı olur. Bu durum, güvenlik kontrollerinin birbirini desteklemesi gerektiğini vurgulayan ”savunmada derinlik” prensibinin bir yansımasıdır.

4.2.5 İçerik Güvenlik Politikası (CSP) ve Güvenlik Başlıkları

İçerik Güvenlik Politikası (CSP), bir web uygulamasının güvenlik modeline ikinci bir savunma katmanı ekleyen bir güvenlik başlığıdır. Sunucu tarafından gönderilen bu HTTP başlığı, tarayıcıya, betikler, stiller, resimler gibi hangi kaynaklardan içerik yükleyebileceğini bildirir.

CSP, XSS saldırılarına karşı birden fazla şekilde koruma sağlar:

- **Satır İçi Betik Kısıtlaması:** `inline` (satır içi) <script> etiketlerinin çalışmasını engeller.
- **Uzak Kaynak Kısıtlaması:** Rastgele sunuculardan betik yüklenmesini engeller.
- **Güvenli Olmayan Fonksiyonların Engellenmesi:** `eval()` gibi güvensiz JavaScript fonksiyonlarının yürütülmesini kısıtlar.
- **Form Kısıtlaması:** HTML formlarının yalnızca belirlenen güvenilir hedeflere veri göndermesini sağlar, bu da kimlik avı formlarının enjekte edilmesini önler.

CSP, ”yalnızca raporlama” moduyla test edilebilir veya doğrudan etkinleştirilerek zorunlu hale getirilebilir. Güvenlik başlıklarının doğru bir şekilde yapılandırılması, uygulamanın genel güvenlik duruşunu önemli ölçüde artırır.

4.3 API Güvenliği ve Microservices Mimarisi

API'ler (Application Programming Interfaces), modern yazılım mimarilerinin temel bir bileşenidir ve farklı uygulamaların birbirleriyle iletişim kurmasını sağlar. Mikroservis mimarileri, büyük ve monolitik uygulamaları daha küçük, bağımsız hizmetlere ayırarak geliştirme ve dağıtım süreçlerini kolaylaştırır. Ancak, bu mimari aynı zamanda yeni güvenlik zorlukları da ortaya çıkarır. Her bir mikroservis ve API, potansiyel bir saldırı yüzeyi oluşturur ve bu nedenle güvenli bir şekilde tasarlanmalı ve yönetilmelidir.

4.3.1 RESTful API Güvenliği En İyi Uygulamaları

RESTful API'ler, web uygulamalarıyla aynı seviyede veya daha fazla güvenlik riskine sahiptir. Bu nedenle, tasarım aşamasından itibaren katı güvenlik prensiplerinin uygulanması gereklidir.

- **Her Zaman TLS Şifrelemesi Kullanın:** Tüm API iletişimleri, uçtan uca veri şifrelemesi sağlamak için TLS (Transport Layer Security) ile korunmalıdır. Bu, ağ trafiği içindeki parolalar, API anahtarları veya tokenlar gibi hassas bilgilerin gizliliğini sağlar.
- **Güçlü Kimlik Doğrulama ve Yetkilendirme Modeli:** API anahtarları ve güvenlik tokenları gibi yöntemlerle ölçeklenebilir bir modelin uygulanması esastır. Bu süreç, özellikle OAuth 2.0 ve OpenID Connect gibi standartlarla entegre bir şekilde yürütüldüğünde, daha güvenli ve yönetilebilir hale gelir.
- **Hassas Bilgileri URL'de Tutmayın:** Bir RESTful API tasarım hatası, kullanıcı kimlik bilgileri veya tokenlar gibi hassas verilerin URL'lere eklenmesidir. Bu, TLS kullanılsa dahi, verilerin sunucu günlüklerinde veya ağ cihazlarında kolayca keşfedilmesine yol açabilir.
- **İstek ve Yanıtları Dar Tanımlayın:** Saldırganların API'leri kötü niyetli veya amaç dışı kullanmaya çalışacağını varsaymak önemlidir. Bu nedenle, gelen parametrelerin formatı, uzunluğu ve tipi gibi nitelikleri titizlikle doğrulanmalı, API yanıtları ise yalnızca açıkça izin verilen içerikle sınırlı tutulmalıdır.
- **Sürekli API Keşfi:** ”Gölge” (shadow) API'leri (standart süreçler dışında geliştirilenler) ve ”zombi” (zombie) API'leri (unutulmuş eski altyapılardaki riskler) gibi varlıkları belirlemek için sürekli bir envanter yönetimi ve keşif yeteneği uygulanmalıdır.

4.3.2 OAuth 2.0, OpenID Connect ve JWT Token Güvenliği

Bu üç teknoloji, API güvenliği için modern kimlik doğrulama ve yetkilendirme standartlarını oluşturur.

- **OAuth 2.0:** Bir yetkilendirme çerçevesidir. Kullanıcının parolalarını paylaşmadan, bir uygulamaya kendi adına belirli kaynaklara erişim izni vermesini sağlar.
- **OpenID Connect (OIDC):** OAuth 2.0 üzerine inşa edilmiş, kimlik doğrulama katmanı ekleyen bir protokoldür. Bir kullanıcının kimliğini doğrulamak ve temel profil bilgilerini almak için kullanılır.
- **JWT (JSON Web Token):** JSON tabanlı, sıkıştırılmış ve güvenli bir belirteç formatıdır. OIDC ve OAuth 2.0'da kimlik ve erişim belirteçlerini iletmek için kullanılır.
- **JWT Güvenliği:**
  - **Doğrulama:** Gelen her JWT, her istekte imzası, süresi (`exp`) ve yayıncısı (`iss`) gibi iddialarının (`claims`) doğrulanması zorunludur. Bu, tokenın sahte veya kurcalanmış olmadığını garanti eder.
  - **Güvenli Saklama:** JWT'ler, Çapraz Site Komut Dosyası Çalıştırma (XSS) saldırılarına karşı `HttpOnly` ve `Secure` bayraklarına sahip çerezlerde saklanmalıdır. `localStorage` veya `sessionStorage` gibi tarayıcı depolama alanları, JavaScript tarafından erişilebilir oldukları için XSS'e karşı savunmasızdır.
  - **Kısa Yaşam Süresi:** JWT'lerin kısa ömürlü olması, token çalınması durumunda saldırganın etki süresini sınırlar. Bu tokenlar, daha uzun ömürlü ancak daha güvenli bir şekilde saklanan yenileme belirteçleriyle (refresh tokens) birlikte kullanılmalıdır.

4.3.3 API Gateway Güvenlik Özellikleri ve Hız Sınırlandırma (Rate Limiting)

Mikroservis mimarisinde, bir API Gateway, hizmetler için tek bir giriş noktası görevi görür. Bu merkezi konum, kimlik doğrulama, yetkilendirme, izleme ve özellikle hız sınırlama gibi güvenlik kontrollerinin tutarlı bir şekilde uygulanmasını sağlar.

- **Hız Sınırlandırma (Rate Limiting):** Bir istemcinin belirli bir zaman diliminde API'ye gönderebileceği istek sayısını kontrol eden bir mekanizmadır. Bu, hizmet reddi (DoS) ve kaba kuvvet saldırılarını önlemenin yanı sıra, sistemin aşırı yüklenmesini engeller ve adil kaynak dağılımını sağlar.
- **Algoritmalar:** En yaygın kullanılan hız sınırlama algoritmaları arasında **Sabit Pencere Sayacı (Fixed Window Counter)**, **Kaydırılan Günlük (Sliding Log)** ve ani trafik artışlarını yönetmek için ideal olan **Token Kovası (Token Bucket)** yer alır.

4.3.4 GraphQL Güvenlik Hususları

GraphQL, esnekliği sayesinde geliştiriciler arasında popülerlik kazanmıştır, ancak bu esneklik kendine özgü güvenlik tehditlerini de beraberinde getirir.

- **DoS Saldırıları:** GraphQL'in iç içe geçmiş sorgulara (nested queries) izin vermesi, bir saldırganın tek bir istekte çok fazla kaynak talep ederek hizmet reddi (DoS) saldırısı gerçekleştirmesine olanak tanır.
- **Azaltım:** Sorgu derinliği ve maliyet analizi sınırlamaları uygulanmalıdır. Bu kontroller, bir sorgunun karmaşıklığına göre sunucunun tüketebileceği maksimum kaynak miktarını sınırlar.
- **Yetkilendirme:** GraphQL API'leri, talep edilen veriler üzerinde yetkilendirme kontrollerini titizlikle uygulamalıdır. Yetkilendirme kontrollerinin hem düğümler (`nodes`) hem de kenarlar (`edges`) üzerinde uygulanması, Broken Object Level Authorization (BOLA) gibi zafiyetleri önler.
- **Güvenli Yapılandırma:** Üretim ortamlarında, API'nin yapısını keşfetmeyi sağlayan **Introspection** sorguları devre dışı bırakılmalı veya yalnızca yetkili kullanıcılara açık olmalıdır. Ayrıca, hata mesajlarında gereksiz iç sistem bilgilerinin (örn. yığın izleri) açığa çıkması engellenmelidir.

4.3.5 Microservices İletişim Güvenliği (mTLS, Service Mesh)

Mikroservisler arası iletişim, genellikle bir ağ üzerinden gerçekleştiği için ortadaki adam (MITM) ve kimliğe bürünme saldırılarına açıktır.

- **mTLS (Mutual TLS):** Geleneksel TLS'nin aksine, mTLS'de hem istemci hem de sunucu, iletişim başlamadan önce birbirlerinin kimliklerini doğrular. Bu, sıfır güven (zero-trust) mimarilerinin temelini oluşturan, karşılıklı kimlik doğrulama ve şifreleme sağlar. Uygulama, OpenSSL gibi araçlarla sertifika ve anahtar çiftleri oluşturularak adım adım uygulanabilir.

- **Service Mesh:** Mikroservisler arası iletişimin güvenliğini, izlenebilirliğini ve yönetimini sağlayan bir altyapı katmanıdır. Service Mesh, mTLS'i hizmetler için otomatik olarak yönetir, bu da her bir hizmetin kendi güvenlik mantığını uygulamasına gerek kalmamasını sağlar.

Service Mesh, mTLS'in tek başına yeterli olmadığı yetkilendirme sorununu da ele alır. mTLS bir hizmetin kimliğini doğrular, ancak o hizmetin belirli bir eylemi yapmaya yetkili olup olmadığını garantilemez. Service Mesh, yetkilendirme kurallarını merkezi olarak tanımlama ve uygulama yeteneği sunarak, sıfır güven mimarisinin "asla güvenme, her zaman doğrula" prensibini en iyi şekilde uygulamayı mümkün kılar. Bu, güvenlik politikasını ağ katmanından daha granüler olan iş yükü (pod) seviyesine taşıyarak daha sağlam bir koruma sağlar.

4.4 Uygulama Güvenlik Testi (AST) Metodolojileri

Uygulama Güvenlik Testi (AST), yazılım geliştirme yaşam döngüsü boyunca güvenlik açıklarını tespit etmek ve düzeltmek için kullanılan bir dizi metodolojiyi ifade eder. AST, statik, dinamik ve interaktif test yaklaşımlarını içerir.

4.4.1 Statik Uygulama Güvenlik Testi (SAST) Araçları

SAST, kaynak kodun, bayt kodun veya ikili kodun çalıştırılmadan analizini yapar. Bu "beyaz kutu" test metodolojisi, SQL enjeksiyonları, arabellek taşmaları ve XSS gibi güvenlik açıklarını, yazılım geliştirme yaşam döngüsünün çok erken aşamalarında, hatta kod derlenmeden önce bile tespit edebilir. SAST araçları, kodun tamamını tarayarak potansiyel güvenlik zafiyetlerini belirler. Bu araçlar, geliştirme boru hatlarına, IDE'lere ve versiyon kontrol sistemlerine entegre edilebilir.

4.4.2 Dinamik Uygulama Güvenlik Testi (DAST) Otomasyonu

DAST, bir uygulamayı çalıştığı sırada test ederek, gerçek dünya saldırılarını simüle eden "siyah kutu" bir yaklaşımdır. Bu yöntem, kaynak koduna erişim olmaksızın, uygulamanın dış arayüzleri aracılığıyla zafiyetleri tespit etmeye odaklanır. DAST, SAST'ın gözden kaçırabileceği kimlik doğrulama bypass'ları, zayıf parolalar veya iş mantığı hataları gibi çalışma zamanı sorunlarını ortaya çıkarır. CI/CD boru hatlarına entegre edilerek, kod derlemesi veya her kod gönderiminde otomatik taramalar gerçekleştirilebilir.

4.4.3 Etkileşimli Uygulama Güvenlik Testi (IAST) Faydaları

IAST, bir uygulamanın içinde çalışarak SAST ve DAST'ın güçlü yönlerini birleştiren hibrit bir metodolojidir. IAST, uygulamanın çalışma zamanı davranışını izleyerek zafiyetin tam olarak hangi kod satırında olduğunu saptar ve bu sayede düzeltme sürecini geliştiriciler için çok daha hızlı hale getirir. DAST'ın aksine, IAST düşük yanlış pozitif oranıyla sonuç üretir ve CI/CD hatlarına sorunsuz bir şekilde entegre olabilen tek dinamik test tekniğidir. Bu sayede, güvenlik kontrolleri geliştiricinin iş akışını bozmadan sürekli olarak uygulanabilir.

4.4.4 Çalışma Zamanı Uygulama Kendi Kendini Koruma (RASP) Uygulaması

RASP, uygulamanın içine yerleştirilen ve canlı ortamda saldırıları gerçek zamanlı olarak algılayıp engelleyen bir güvenlik teknolojisidir. Bir RASP aracı, bir ağ güvenlik duvarı (WAF) gibi ağ tabanlı çözümlerden farklı olarak, uygulamanın davranışını ve içinde gerçekleşen olayları analiz eder. Bu sayede, saldırgan çevre savunmasını aştıktan sonra bile uygulamayı koruyabilir ve SQL enjeksiyonu veya XSS gibi saldırıları, uygulamaya ulaşmadan önce engelleyebilir. RASP, uygulamanın kodunda herhangi bir değişiklik yapılmasını gerektirmez, bu da mevcut uygulamaların kolayca korunmasını sağlar.

4.4.5 Yazılım Bileşen Analizi (SCA) ve Açık Kaynak Güvenliği

Yazılım Bileşen Analizi (SCA), uygulamalarda kullanılan üçüncü taraf ve açık kaynak bileşenleri tarayarak bilinen güvenlik zafiyetlerini ve lisans uyumluluğu sorunlarını bulur. Açık kaynak bileşenleri, bilinen zafiyetler içerebileceğinden, bu zafiyetlerin üretime dağıtılmadan önce belirlenmesi ve yönetilmesi hayati önem taşır.

SCA'nın en iyi uygulamalarından biri, her derleme (build) için bir Yazılım Malzeme Listesi (SBOM) oluşturmaktır. SBOM, projedeki tüm yazılım bileşenlerinin detaylı bir envanterini sunar ve sıfır gün (zero-day) zafiyetleri gibi durumlarda etkilenen bileşenlerin hızlıca tespit edilmesini sağlar. SCA'yı CI/CD boru hattına entegre etmek, politikalara aykırı veya risk içeren birleştirme isteklerinin (pull requests) ana kod tabanına birleşmesini otomatik olarak engellemeye yardımcı olur. Bu, geliştirme sürecini yavaşlatmadan güvenliği sürekli kılar.

4.5 DevSecOps Entegrasyonu ve CI/CD Pipeline Güvenliği

DevSecOps, geliştirme (Development), güvenlik (Security) ve operasyon (Operations) ekiplerini bir araya getirerek, güvenlik kontrollerini yazılım geliştirme yaşam döngüsünün her aşamasına entegre eden bir yaklaşımdır. Bu model, güvenliği bir "sonradan eklenen" bir özellik olarak değil, sürecin ayrılmaz bir parçası olarak ele alır. CI/CD (Continuous Integration/Continuous Deployment) pipeline'ları, yazılımın hızlı ve otomatik bir şekilde oluşturulmasını, test edilmesini ve dağıtılmasını sağlar. Bu pipeline'lara güvenlik kontrollerinin entegre edilmesi, güvenlik açıklarının erken tespit edilmesine ve düzeltilmesine olanak tanır.

4.5.1 Güvenlik Araç Zinciri Entegrasyonu ve Otomasyonu

Güvenlik otomasyonu, modern yazılım geliştirmede ölçeklenebilirlik, esneklik ve uyumluluk sağlamanın temelidir. Bu, güvenlik tarama araçlarının (SAST, SCA) geliştirici iş akışına ve `pre-commit` hook'larına entegre edilmesiyle başlar. Otomatik güvenlik araçları, her kod gönderiminde veya birleştirme isteğinde çalıştırılarak geliştiricilere anında geri bildirim sağlar.

Bu araçların doğru bir şekilde orkestrasyonu, tehditlerin otomatik olarak algılanmasını ve yanıt verilmesini sağlar. Güvenlik olayları, izleme araçları tarafından tespit edildiğinde, otomatik düzeltme iş akışlarını tetikleyebilir.

4.5.2 Konteyner Güvenliği Taraması ve İmge Zafiyet Yönetimi

Konteyner imajları, zafiyet içeren üçüncü taraf kütüphaneleri ve paketleri içerebilir. Bu zafiyetlerin üretime dağıtılmadan önce belirlenmesi zorunludur. Konteyner güvenliği taraması, bir konteyner imajındaki bilinen zafiyetleri, hatalı yapılandırmaları ve diğer güvenlik sorunlarını tanımlayan bir süreçtir.

Otomatik konteyner taraması, CI/CD boru hattına entegre edilerek, her derleme veya dağıtım aşamasında imajların sürekli olarak taranmasını sağlar. Bu taramaların sonuçları, zafiyetin açıklaması, ciddiyeti ve düzeltme adımları gibi ayrıntılı bilgiler içerir. En iyi uygulama, yüksek riskli zafiyetler içeren imajların dağıtımını otomasyonla engellemektir. Bu, dağıtım kapıları (security gates) aracılığıyla uygulanabilir.

4.5.3 Kod Olarak Altyapı (IaC) Güvenlik Taraması

Kod Olarak Altyapı (IaC), (örn. Terraform, CloudFormation), altyapıyı kod olarak yönetmeyi sağlar. Bu kodlar, açık ağ portları veya sabit kodlanmış kimlik bilgileri gibi hatalı yapılandırmalar içerebilir. IaC güvenlik taraması, bu tür

hataları altyapı dağıtılmadan önce tespit eden bir güvenlik kontrolüdür. Bu süreç, IaC dosyalarını analiz ederek önceden tanımlanmış güvenlik politikaları ve kurallarına aykırı olan riskli ayarları işaretler. Böylece, altyapı dağıtımı sırasında oluşabilecek güvenlik zafiyetleri proaktif olarak engellenir.

4.5.4 DevOps'ta Güvenlik Kapıları ve Kalite Kontrolü

Güvenlik kapısı (Security Gate), bir yazılımın bir sonraki aşamaya geçmesi için karşılaması gereken bir dizi güvenlik gereksinimidir. Bu kapılar, bir derlemenin (build) devam etmesi için belirli bir eşiğin altında güvenlik sorunu barındırması gibi koşulları belirleyebilir.

Otomatik güvenlik taramaları bir "geçti/kaldı" kriteri olarak yapılandırılarak, belirlenen eşiğin altındaki güvenlik sorunları çözülmeden boru hattının ilerlemesi engellenir. Bu, teknik borcun birikmesini önler ve güvenliğin sonradan eklenen bir düşünce değil, sürecin doğal bir parçası olmasını sağlar. Güvenlik kapıları, ürün kalitesi ve güvenliğin ayrılmaz bir şekilde bağlantılı olduğunu vurgular.

4.5.5 Sola Kaydırma (Shift-Left) Güvenlik Kültürü ve Geliştirici Eğitimi

DevSecOps'un başarısı, güvenlik kapıları ve otomasyon araçlarından daha fazlasına bağlıdır; geliştiricilerin güvenliği kendi sorumlulukları olarak benimsemesini sağlayan bir kültür değişimini gerektirir.

Güvenlik ekipleriyle yakın iş birliği ve sürekli geliştirici eğitimi, bu kültürün temelidir. Geliştiricilerin güvenli kodlama pratikleri konusunda eğitilmesi ve güvenlik araçlarının sunduğu geri bildirimleri anlamaları, güvenlik açıklarının daha kod yazılırken engellenmesine yardımcı olur. Bu yaklaşım, güvenlik sorunlarının hızlı ve etkili bir şekilde çözülmesini sağlar ve geliştirme sürecini daha verimli hale getirir.

4.6 Endüstriyel Kontrol Sistemleri (ICS) için ATT&CK Çerçevesi

MITRE ATT&CK for ICS (Industrial Control Systems), endüstriyel kontrol sistemlerine yönelik saldırı taktiklerini ve tekniklerini sistematik olarak kategorize eden bir bilgi bankasıdır. Bu çerçeve, kritik altyapı operatörlerinin ve güvenlik ekiplerinin, ICS ortamlarına yönelik tehditleri daha iyi anlamalarını ve bunlara karşı savunma stratejileri geliştirmelerini sağlar.

4.6.1 ICS Taktikleri ve Teknikleri

ATT&CK for ICS, şu ana taktik kategorilerini içerir:

- **İlk Erişim (Initial Access):** Saldırganların ICS ağına ilk giriş yöntemleri.
  - Mühendislik İş İstasyonlarının Ele Geçirilmesi
  - Dış Uzaktan Hizmetler
  - İnternet Erişilebilir Cihaz
  - Yeniden Satıcı/Tedarik Zinciri Ele Geçirme
- **Keşif (Discovery):** ICS ortamını ve sistemlerini öğrenme teknikleri.
  - Kontrol Cihazı Keşfi
  - I/O Modül Keşfi
  - Seri Bağlantı Keşfi
  - Program Organizasyon Birimi Keşfi
- **Kalıcılık (Persistence):** Sistemde uzun süreli erişim sağlama yöntemleri.
  - Modül Firmware'i
  - Sistem Firmware'i

  - Proje Dosya Enfeksiyonu

- **Yan Hareket (Lateral Movement):** ICS ağı içinde hareket etme teknikleri.
  - Programlama Yazılımı Kullanarak Erişim
  - Uzak Hizmetler Üzerinden Hareket
  - Veri Toplama Protokollerini Kötüye Kullanma

- **Etki (Impact):** Sistemin işleyişini bozma veya kesintiye uğratma yöntemleri.
  - Kontrol Mantığının Değiştirilmesi
  - Parametre Değişiklikleri
  - Hizmet Kesintisi
  - Güvenlik Enstrümantasyon Sisteminin Atlatılması

4.6.2 ICS Savunma Stratejileri

ATT&CK for ICS çerçevesi, her teknik için karşı önlemler ve algılama yöntemleri önerir:

- **Ağ Segmentasyonu:** ICS ağının kurumsal ağdan ve internetten izole edilmesi.
- **Erişim Kontrolü:** Sıkı erişim kontrol politikaları ve çok faktörlü kimlik doğrulama.
- **Güvenlik İzleme:** ICS-spesifik anormallik tespiti ve olay izleme.
- **Güvenlik Duvarları:** Endüstriyel protokolleri anlayan özelleştirilmiş güvenlik duvarları.
- **Varlık Yönetimi:** Tüm ICS bileşenlerinin detaylı envanteri.

4.6.3 Risk Değerlendirme ve Azaltma

ICS güvenliği için risk değerlendirmesi, şu adımları içerir:

1. **Varlık Tanımlama:** Kritik ICS bileşenlerinin belirlenmesi.
2. **Tehdit Modelleme:** ATT&CK for ICS kullanarak olası saldırı vektörlerinin belirlenmesi.
3. **Zafiyet Analizi:** Sistemdeki güvenlik açıklarının tespiti.
4. **Risk Değerlendirme:** Tehditlerin olasılık ve etki analizinin yapılması.
5. **Kontrol Seçimi:** Uygun güvenlik kontrollerinin belirlenmesi ve uygulanması.

4.7 Mobil Uygulama Güvenliği

Mobil uygulama güvenliği, kullanıcı verilerini korumak ve yetkisiz erişimi önlemek için kritik öneme sahiptir. Mobil uygulamalar, hem cihazda hem de sunucu tarafında çeşitli güvenlik riskleriyle karşı karşıyadır. Bu riskler, veri sızıntıları, kötü amaçlı yazılım bulaşmaları ve kimlik avı saldırılarını içerir.

4.7.1 iOS Uygulama Güvenliği: Kod İmzalaması, App Transport Security (ATS)

- **Kod İmzalaması (Code Signing):** Bir iOS uygulamasının, Apple tarafından yetkilendirilmiş bir geliştirici tarafından imzalandığını doğrular. Bu imza, uygulamanın dağıtıldıktan sonra kurcalanmadığını ve değiştirilmediğini garanti eder.
- **App Transport Security (ATS):** iOS 9 ve sonraki sürümler için varsayılan olarak etkin olan bir güvenlik özelliğidir. ATS, bir uygulamanın sunucuyla kurduğu tüm ağ bağlantılarının, minimum güvenlik gereksinimlerini (örn. TLS 1.2 veya daha yeni bir sürüm, güçlü anahtar boyutları) karşılayan güvenli bir TLS protokolüyle şifrelenmesini zorunlu kılar.

4.7.2 Android Uygulama Güvenliği: ProGuard, Sertifika Sabitleme (Certificate Pinning)

- **ProGuard:** Kod gizleme (obfuscation) ve optimizasyon aracıdır. Uygulamanın kodunu okunması ve tersine mühendislikle analiz edilmesi zor bir hale getirerek, fikri mülkiyetin korunmasına yardımcı olur. `build.gradle` dosyasında kolayca etkinleştirilebilir.
- **Sertifika Sabitleme (Certificate Pinning):** Bir uygulamanın yalnızca belirli, önceden tanımlanmış sunucu sertifikalarına veya ortak anahtarlara güvenmesini sağlayan bir tekniktir. Bu, ortadaki adam (MITM) saldırılarına karşı güçlü bir savunma mekanizmasıdır. Ancak bu teknik, sunucu tarafındaki sertifika rotasyonu gibi durumlar için alternatif sabitleme (`pinset`) mekanizmalarıyla dikkatli bir şekilde yönetilmelidir.

4.7.3 Mobil Uygulama Sızma Testi Metodolojileri

Mobil uygulama sızma testi, uygulamadaki zafiyetleri bulmak için gerçek dünya saldırılarını simüle eder.

- **Statik Analiz:** Uygulama çalıştırılmadan kaynak kodu veya ikili dosyaları incelenir. Bu yöntem, sabit kodlanmış kimlik bilgileri ve güvensiz kodlama pratikleri gibi sorunları tespit eder.
- **Dinamik Analiz:** Uygulama çalışırken, ağ trafiği, bellek kullanımı ve çalışma zamanı davranışı izlenir. Bu, çalışma zamanı zafiyetlerini (örn. güvensiz veri depolama, zafiyetli iletişim) ortaya çıkarır.
- **En İyi Uygulamalar:** Mobil uygulama sızma testleri, OWASP Mobile Application Security Testing Guide (MASTG) gibi çerçeveler kullanılarak sistematik bir şekilde yürütülmelidir.

4.7.4 Binary Koruma ve Anti-tampering Teknikleri

- **İkili Koruma (Binary Protection):** Uygulama ikililerini tersine mühendislikten ve izinsiz değişikliklerden korumayı amaçlar.
- **Anti-tampering:** Uygulamanın çalışırken veya statik haldeyken kurcalanmasını (değiştirilmesini) engellemek için tasarlanmış tekniklerdir.
- **Teknikler:** Kod gizleme (obfuscation), dosya bütünlüğü kontrolü, hata ayıklama (debugger) algılama, jailbreak/root algılama gibi çeşitli yöntemler kullanılır.
- **Savunma:** Statik (obfuscation, encryption) ve dinamik (RASP) önlemleri birleştiren çok katmanlı bir yaklaşım benimsenmelidir. RASP, çalışma zamanında kurcalama girişimlerini algılayıp yanıt vererek dinamik saldırılara karşı koruma sağlar.

4.7.5 Mobil Backend API Güvenliği Hususları

Mobil uygulamalar, kullanıcı verilerini yönetmek ve iş mantığını yürütmek için arka uç API'leriyle iletişim kurar. Bu API'lerin güvenliği, en az mobil uygulama kadar kritik öneme sahiptir. Mobil backend API güvenliği, bu raporun Bölüm 4.3'te ele alınan genel API güvenliği ilkelerine dayanır. Bu, hem platformlar arası güvenlik tutarlılığını hem de savunmada derinlik prensibini pekiştirir. API'ler için TLS şifrelemesi, güçlü kimlik doğrulama, hız sınırlama ve API ağ geçidi kullanımı gibi uygulamalar, mobil ekosistemin bütünlüğünü korumada anahtar rol oynar.

Bölüm 5

BULUT VE İŞ YÜKÜ GÜVENLİĞİ

Giriş

Bulut güvenliği, modern IT altyapısının temelini oluşturan bulut bilişim teknolojilerinin güvenli bir şekilde kullanılmasını sağlayan kritik bir alandır. Bu bölümde bulut güvenlik mimarileri, konteyner güvenliği ve iş yükü korunması konularını inceleyeceğiz.

5.1 Bulut Güvenlik Mimarisi ve Paylaşılan Sorumluluk Modeli

Bulut bilişim, kuruluşların altyapı yönetimi yükünü önemli ölçüde azaltırken, güvenlik yaklaşımında köklü bir değişikliği zorunlu kılmıştır. Geleneksel güvenlik modelleri, genellikle fiziksel veri merkezinin etrafına inşa edilen bir ”çevre”yi korumaya odaklanırken, bulut ortamları dinamik ve dağınık yapısıyla yeni bir mimari ve sorumluluk çerçevesi gerektirir. Bu çerçeve, bulutun doğası gereği esneklik ve paylaşılan yetki alanları sunar ve bu nedenle güvenlik, yalnızca tek bir tarafın görevi olmaktan çıkmıştır.

5.1.1 Bulut Güvenlik Mimarisi İlkeleri ve Referans Mimariler

Bir bulut güvenlik mimarisi, bir kuruluşun buluta geçişle birlikte ortaya çıkan benzersiz güvenlik zorluklarından korunmasını sağlayan bir çerçevedir. Bu çerçeve, yalnızca teknolojik bileşenleri değil, aynı zamanda bu bileşenlerin nasıl tasarlanıp yönetileceğini belirleyen süreçleri ve ilkeleri de kapsar. Bulut güvenlik mimarisinin temel ilkeleri, riski yönetirken iş operasyonlarını kolaylaştırma dengesini gözetir.

Bu tasarımın temelinde yer alan ilkelerden biri otomasyondur. Bulut ortamlarının dinamik doğası ve hızlı ölçeklenebilme kapasitesi, güvenlik önlemlerinin manuel olarak uygulanmasını verimsiz ve hataya açık hale getirir. Güvenlik kontrollerinin ve dağıtımlarının otomasyonu, altyapı genişledikçe güvenlik önlemlerini ölçeklendirmek için temel bir gerekliliktir. Bu sayede güvenlik ekipleri tekrarlayan görevlerden kurtularak daha stratejik konulara odaklanabilir. Bir diğer kritik ilke ise derinlemesine savunma (Defense-in-Depth) yaklaşımıdır. Bu mimari prensip, tek bir güvenlik kontrolünün arızalanması durumunda sistemi koruyacak katmanlı kontrollerin uygulanmasını içerir. Bu, güvenlikte tek bir hata noktasını önlemeyi amaçlar. Son olarak, sıfır güven (Zero Trust) yaklaşımı, bir bulut güvenlik mimarisinin temel taşıdır. Bu model, ağ içindeki veya dışındaki hiçbir kullanıcıya ya da kaynağa peşin olarak güvenmez; bunun yerine, kimlik ve bağlama dayalı sürekli risk değerlendirmesi ve doğrulama gerektirir.

Bu ilkeler, AWS Güvenlik Referans Mimarisi (SRA) gibi somut kılavuzlarda uygulanmaktadır. AWS SRA, AWS Organizations kullanılarak çoklu hesap yapılarının nasıl güvenli hale getirileceğini gösterir. Bu, organizasyon genelinde güvenlik hizmetlerinin tutarlı bir şekilde nasıl uygulanacağına dair pratik bir yol haritası sunar. Ancak, bir referans mimarisi, her iş yükünün veya ortamın benzersiz tehdit maruziyetine göre tüm güvenlik hizmetlerini uygulamak zorunda olmadığını da belirtir. Bu, mimarinin katı kurallar bütünü yerine, riske dayalı ve duruma göre uyarlanabilir bir yaklaşım sergilemesi gerektiğini vurgular.

5.1.2 Paylaşılan Sorumluluk Modeli: IaaS, PaaS, SaaS Güvenlik Sorumlulukları

Bulut güvenliğinin en temel ve en sık yanlış anlaşılan kavramlarından biri Paylaşılan Sorumluluk Modeli'dir. Bu model, bulut sağlayıcısı (CSP) ve müşteri (CSC) arasında güvenlik görevlerinin net bir şekilde ayrıştırılmasını tanımlar. Temel ilke, sağlayıcının **"bulutun güvenliğinden"** (fiziksel altyapı, donanım, ağ) sorumlu olması, müşterinin ise **"bulttaki güvenlikten"** (veri, uygulamalar, yapılandırmalar) sorumlu olmasıdır. Bu ayrım, her bir bulut hizmet modeline (IaaS, PaaS, SaaS) göre farklılık gösterir ve güvenlik görevlerinin dağılımını belirler. Bu modelin doğru anlaşılması, potansiyel güvenlik risklerinin ele alınması, güvenlik duruşunun güçlendirilmesi ve ihlallere yol açabilecek yanlış anlamaların önüne geçilmesi için hayati önem taşır.

Hizmet modellerine göre sorumluluklar şu şekilde detaylandırılabilir:

- **IaaS (Hizmet Olarak Altyapı):** AWS EC2 gibi IaaS hizmetlerinde müşteri, güvenlik görevlerinin çoğundan sorumludur. Bulut sağlayıcısı fiziksel altyapıyı yönetirken, müşteri işletim sistemi güncellemeleri, yama yönetimi, güvenlik grubu ve güvenlik duvarı yapılandırmaları, veri şifreleme ve IAM politikaları gibi görevleri üstlenmelidir. Bu modelde, müşteri kendi sanal veri merkezini yönetir.
- **PaaS (Hizmet Olarak Platform):** Microsoft Azure App Services gibi PaaS hizmetlerinde, bulut sağlayıcısı işletim sistemi, çalışma ortamı (runtime) ve ara katman yazılımları gibi bileşenlerin güvenliğini de sağlar. Müşterinin sorumluluğu, esas olarak kendi uygulamalarının ve bu uygulamalar içindeki verilerinin güvenliğine odaklanır.
- **SaaS (Hizmet Olarak Yazılım):** Microsoft 365 veya Google Workspace gibi SaaS hizmetlerinde, sağlayıcı yazılım, altyapı ve temel sistemler dahil olmak üzere neredeyse tüm güvenlik yönlerini yönetir. Müşteri, öncelikli olarak kimlik ve erişim yönetimini (IAM) ve verilerinin korunmasını sağlamalıdır.

Bu modelin en önemli çıkarımı, bulut sağlayıcısının kendi yükümlülüklerini yerine getirse bile, müşterinin tarafındaki yanlış yapılandırmaların veya zayıf erişim kontrollerinin ihlallere neden olabileceği gerçeğidir. Paylaşılan Sorumluluk Modeli, güvenlik boşluklarının önlenmesi, düzenleyici uyumluluğun sağlanması ve yasal yükümlülüğün azaltılması için kritik bir çerçevedir.

Tablo 5.1: Paylaşılan Sorumluluk Modeli: Hizmet Modellerine Göre Sorumluluk Dağılımı

Güvenlik Görevi	**IaaS (Örn: AWS EC2)**	**PaaS (Örn: Azure App Services)**	**SaaS (Örn: Microsoft 365)**
Fiziksel Altyapı ve Tesis Güvenliği	Bulut Sağlayıcısı	Bulut Sağlayıcısı	Bulut Sağlayıcısı
Donanım, Yazılım ve Ağ	Bulut Sağlayıcısı	Bulut Sağlayıcısı	Bulut Sağlayıcısı
İşletim Sistemi ve Sanallaştırma Katmanı	Müşteri	Bulut Sağlayıcısı	Bulut Sağlayıcısı
Uygulamalar ve Ara Katman Yazılımları	Müşteri	Müşteri	Bulut Sağlayıcısı
Kimlik ve Erişim Yönetimi (IAM)	Müşteri	Müşteri	Müşteri
Veri ve Şifreleme	Müşteri	Müşteri	Müşteri
Yapılandırma Yönetimi	Müşteri	Müşteri	Müşteri

5.1.3 Çoklu-Bulut ve Hibrit Bulut Güvenlik Stratejileri

Modern işletmeler, adaptasyon yetenekleri ve maliyet etkinliği gibi nedenlerle çoklu bulut ve hibrit bulut ortamlarını benimsemektedir. Ancak, bu stratejiler güvenlik yönetimini karmaşıklaştırır ve görünürlük eksikliği gibi zorlukları beraberinde getirir. Çoklu bulut güvenliği, birden fazla bulut ortamında tutarlı görünürlük, politika ve yönetişim sağlamak için tek bir yönetim noktasını kullanan bir yaklaşımdır.

Etkili bir çoklu ve hibrit bulut güvenlik stratejisi oluşturmak için izlenmesi gereken adımlar şunlardır:

1. **Paylaşılan Sorumluluk Modelini Anlamak:** Her bir bulut sağlayıcısının farklı sorumluluk seviyelerine sahip olabileceğini kavramak önemlidir. Bu, güvenlik boşluklarını önlemek için ek önlemlerin nerede uygulanması gerektiğini belirlemeye yardımcı olur.

2. **Kapsamlı Bir Güvenlik Politikası Uygulamak:** Veri merkezleri ve tüm bulut ortamları arasında uçtan uca, tutarlı bir güvenlik politikası uygulanmalıdır. Bu politikalar, şifreleme, erişim kontrolü ve uygulama katmanı güvenliğini içermeli ve güncel en iyi uygulamaları yansıtacak şekilde düzenli olarak gözden geçirilmelidir.

3. **Merkezi Görünürlük ve Kontrolü Sürdürmek:** Etkili güvenlik yönetimi için merkezi izleme ve yönetim araçları hayati öneme sahiptir. Bu araçlar, tüm bulut ortamlarına gerçek zamanlı görünürlük sağlayarak güvenlik olaylarına hızlı yanıt verilmesini mümkün kılar. Çoklu bulut ortamlarının dağınık yapısı, merkezi bir bulut güvenlik aracı olmadan yönetişim ve görünürlük zorlukları yaratır.

4. **En Az Ayrıcalık Prensibini Uygulamak:** Kimlik ve erişim yönetimini (IAM) merkezi bir stratejiyle yönetmek, her bir bulut hizmet sağlayıcısının farklı kimlik doğrulama ve yetkilendirme protokollerini kullanması sorununu hafifletir. Bu, tüm altyapı genelinde tutarlı bir şekilde en az ayrıcalık erişimi politikalarının otomatik olarak uygulanmasını sağlar.

5.1.4 Bulut Güvenlik Çerçeveleri: Cloud Security Alliance (CSA) ve NIST Uygulamaları

Bulut güvenliğindeki standartlaşma ve yönetişim, CSA ve NIST gibi kuruluşların sağladığı çerçeveler aracılığıyla sağlanır. Bu çerçeveler, kuruluşların bulut güvenliğini sistematik bir yaklaşımla ele almasını, uyumluluk gereksinimlerini karşılamasını ve risk temelli bir güvenlik duruşu oluşturmasını destekler.

- **Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM):** CSA CCM, bulut bilişim için tasarlanmış bir siber güvenlik kontrol çerçevesidir. 17 alanda 197 kontrol hedefinden oluşur ve bulut tedarik zincirindeki hangi güvenlik kontrollerinin hangi aktör tarafından uygulanması gerektiği konusunda rehberlik sunar. CCM, kuruluşların ISO, NIST ve PCI DSS gibi çoklu güvenlik standartlarına uyum sağlamasına yardımcı olarak süreci kolaylaştırır. CCM'in farklı standartlara eşleşmesi, sektörde güvenlik ve uyumluluk için tek bir dil oluşturma çabasını ortaya koyar. Bu, bulut güvenliğinin artık sadece bir teknoloji sorunu olmaktan çıkıp, uyumluluk ve yönetişim odaklı bir iş stratejisine dönüştüğünün bir göstergesidir.

- **NIST Cloud Security Guidelines:** Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), Federal Bilgi Güvenliği Yönetimi Yasası (FISMA) gibi federal yasalar kapsamında bilgi güvenliğine ilişkin standartlar ve yönergeler geliştirir. NIST Özel Yayın (SP) 800-53, ABD federal hükümet kurumları ve kritik altyapılar için gerekli güvenlik ve gizlilik kontrollerini belirler. Bu çerçeve, kuruluşların risk değerlendirmesi yapmasına, uygun güvenlik kontrollerini seçip uygulamasını sağlamasına ve güvenlik duruşunu sürekli olarak izlemesine olanak tanıyan sistematik bir yaklaşım sunar. NIST'in kılavuzları, bulut ortamlarında güvenlik tehditlerini sistemik olarak yönetmek ve düzenleyici gerekliliklere uyumu sağlamak için pratik adımlar sağlar.

Bulut güvenliğinde referans mimarilerinin ve belirli ürünlerin (örneğin Google Cloud) bu gibi çerçevelere (NIST) uygunluğunun vurgulanması, bu standartların artık yalnızca birer kılavuz değil, aynı zamanda pratik uygulamaların güvenilirliğini ve pazarlama değerini artıran birer unsur haline geldiğini gösterir.

5.2 Bulut Altyapı Güvenliği ve Yönetişim

Bulut altyapısının dinamik ve programlanabilir doğası, güvenliğin de benzer şekilde kodlanabilir ve otomatikleştirilebilir olmasını gerektirmektedir. Bu bölüm, bulut ortamlarının yapılandırmasını ve korunmasını yönetmek için kullanılan modern araç ve yöntemleri ele almaktadır. Odak noktası, insan hatasını azaltan ve tutarlılığı artıran otomasyon ve politika tabanlı yaklaşımlardır.

5.2.1 Kod Olarak Altyapı (IaC) Güvenlik En İyi Uygulamaları

Kod Olarak Altyapı (IaC), altyapının insan tarafından okunabilir kod dosyalarıyla tanımlanmasını ve yönetilmesini sağlar. Bu yaklaşım, güvenlik açısından, yapılandırma hatalarının daha dağıtım aşamasına gelmeden önce "sola kaydırılarak" (shift-left) tespit edilmesini mümkün kılar. IaC güvenliği, geleneksel manuel süreçlerdeki tutarsızlıkları ve hataları ortadan kaldırarak daha güvenli ve tutarlı bir altyapı oluşturur.

Güvenli bir IaC stratejisi için en iyi uygulamalar şunlardır:

1. **Gizli Verilerin Koddan Ayrılması:** API anahtarları, parolalar veya jetonlar gibi gizli veriler asla doğrudan IaC dosyalarına gömülmemelidir. Bunun yerine, AWS Secrets Manager, Azure Key Vault veya HashiCorp Vault gibi özel gizlilik yönetimi araçları kullanılmalıdır. Bu araçlar, hassas verilerin koddan ayrı ve güvenli bir şekilde saklanmasını sağlar.
2. **Sürüm Kontrolü ve Akran Denetimi Uygulaması:** Tüm altyapı kodları bir sürüm kontrol deposunda (örneğin Git) saklanmalı ve değişiklikler pull requestler aracılığıyla akran denetiminden geçirilmelidir. Bu, yanlış yapılandırmaların erken aşamada yakalanmasını ve birden fazla geliştiricinin güvenliği gözden geçirmesini sağlar.
3. **CI/CD Ortamında Güvenlik Taramaları:** Geliştirme ve dağıtım süreçlerinin otomatikleştirildiği CI/CD hatlarına güvenlik tarama araçları entegre edilmelidir. `Checkov`, `tfsec` veya `cfn-lint` gibi statik analiz araçları, dağıtımdan önce güvenlik risklerini otomatik olarak tespit eder.

Pratik Senaryo: `Checkov` ile Yanlış Yapılandırılmış Terraform Kodunun Analizi

Bu senaryo, herkese açık erişime izin veren yanlış yapılandırılmış bir AWS S3 kovasını (bucket) tanımlayan bir Terraform dosyasının güvenlik denetimini göstermektedir.

Adım 1: Güvenlik Açığı İçeren Terraform Dosyasını Oluşturma (`main.tf`)

Bu dosya, S3 kovası için genel erişim engelleme ayarlarını `false` olarak belirleyerek bir güvenlik açığı oluşturur.

```
resource "aws_s3_bucket" "example" {
  bucket = "my-insecure-bucket-12345"
}

resource "aws_s3_bucket_public_access_block" "example" {
  bucket = aws_s3_bucket.example.id
  block_public_acls = false
  block_public_policy = false
  ignore_public_acls = false
  restrict_public_buckets = false
}
```

Adım 2: `Checkov` ile Taramayı Çalıştırma

Dosyayı kaydettikten sonra, terminalden aşağıdaki komut çalıştırılır:

```
checkov -f main.tf
```

Adım 3: Komut Satırı Çıktısının Analizi `Checkov` taraması, aşağıdaki gibi bir çıktı üreterek güvenlik açıklarını net bir şekilde gösterir:

```
    Passed checks: 2, Failed checks: 4, Skipped checks: 0
Check: CKV_AWS_53: "Ensure S3 bucket has block public ACLS enabled" FAILED for resource: aws_s3_bucke
Check: CKV_AWS_54: "Ensure S3 bucket has block public policy enabled" FAILED...
```

Çıktı, `block_public_acls` ve `block_public_policy` gibi kritik güvenlik ayarlarının beklendiği gibi yapılandırılmadığını belirtir. **Adım 4: Güvenlik Açıklarını Giderme** Geliştirici, `main.tf` dosyasını `block_public_acls = true` ve `block_public_policy = true` olarak düzelterek güvenlik politikasını uygular. Bu basit düzeltme, kodun güvenli bir varsayılan duruma (secure-by-default) dönmesini sağlar. Bu pratik senaryo, IaC tarama araçlarının dağıtımdan önce güvenlik açıklarını nasıl yakaladığına dair somut bir örnek sunar.

5.2.2 Bulut Güvenlik Duruşu Yönetimi (CSPM) Araçları

Bulut Güvenlik Duruşu Yönetimi (CSPM), bulut ortamlarındaki yanlış yapılandırmaları ve uyumluluk ihlallerini sürekli olarak izlemek ve düzeltmek için kullanılan otomasyon tabanlı bir yaklaşımdır. CSPM araçları, güvenlik denetimini otomatikleştirmeleri sayesinde, bulut ortamlarının hızlı ölçeklenmesine ayak uydurabilir.

CSPM'nin temel işlevleri şunlardır:

1. **API Entegrasyonu:** CSPM çözümleri, aracı (agent) gerektirmezler. Bunun yerine, bulut sağlayıcılarının API'lerine (AWS, Azure, GCP) bağlanarak envanter, yapılandırmalar ve denetim kayıtları gibi verilere erişirler. Bu aracısız yaklaşım, dağıtımı kolaylaştırır.
2. **Yanlış Yapılandırma Tespiti:** CSPM araçları, bulut kaynaklarının yapılandırmalarını CIS Benchmarks veya PCI DSS gibi endüstri standartlarına ve düzenleyici çerçevelere göre denetleyen önceden tanımlanmış politikalarla gelir. Örneğin, herkese açık erişimi olan bir S3 kovası veya herkese açık bir Kubernetes uç noktası, bir yanlış yapılandırma olarak otomatik olarak raporlanır.
3. **Risk Bağlamsallaştırması ve Önceliklendirme:** Gelişmiş CSPM çözümleri, riskleri bağlamsallaştırmak için tekil miskonfigürasyonların ötesine geçer. Zafiyetler, aşırı izinler ve aktif tehditler gibi ek bulguları bir araya getirerek olası saldırı yollarını ortaya çıkarır. Bu, güvenlik ekiplerinin binlerce uyarı içinde en kritik tehditlere odaklanmasını sağlar.

CSPM araçları, pasif bir denetim mekanizması olmaktan çıkıp, siber riskleri aktif olarak önceliklendiren ve saldırı yollarını haritalayan akıllı platformlara dönüşmüştür. Bu evrim, güvenlik operasyonlarında reaktiflikten proaktifliğe doğru bir değişimi temsil etmektedir.

5.2.3 Bulut İş Yükü Koruma Platformu (CWPP) Çözümleri

Bulut İş Yükü Koruma Platformu (CWPP), sanal makineler, konteynerler ve sunucusuz fonksiyonlar gibi bulut iş yüklerini siber tehditlere karşı korumayı amaçlayan bir çözümdür. CWPP'ler genellikle bir CWPP/CSPM/CRD platformu olan daha geniş bir CNAPP'nin (Bulut Yerel Uygulama Koruma Platformu) bir alt kümesidir. CWPP'nin yükselişi, geleneksel güvenlik çözümlerinin modern bulut ortamlarının dinamik ve heterojen yapısını korumada yetersiz kaldığının bir kanıtıdır.

CWPP'nin temel yetenekleri, çok çeşitli bulut iş yükleri için kapsamlı bir koruma sağlar:

- **Zafiyet Yönetimi:** CWPP, iş yüklerindeki zafiyetleri sürekli olarak değerlendirir ve CVE (Common Vulnerabilities and Exposures) veritabanları gibi kaynakları kullanarak risklerini önceliklendirir.
- **Çalışma Zamanı Koruması (Runtime Protection):** CWPP'ler, iş yükleri çalışırken onları korur. Bu, davranışsal analiz, makine öğrenimi ve imza tabanlı algılama gibi yöntemleri kullanarak gerçek zamanlı tehdit tespiti yapmayı içerir.
- **Yapılandırma Yönetimi:** CWPP'ler, CIS Benchmarks gibi endüstri standartlarına uygunluğu denetler ve güvenli yapılandırmaların uygulanmasına yardımcı olur.
- **CI/CD Entegrasyonu:** CWPP'ler, güvenlik geri bildirimini yazılım geliştirme yaşam döngüsünün (SDLC) erken aşamalarına entegre ederek "sola kaydırma" (shift-left) güvenlik uygulamalarını kolaylaştırır.

CWPP çözümleri, iş yüklerinin çok çeşitlilik göstermesi (VM'ler, konteynerler, sunucusuz) ve çoklu bulut ortamlarının karmaşıklığı nedeniyle tekil bir platformun tümünü korumasını gerekli kılmıştır.

5.2.4 Bulut Erişim Güvenlik Aracısı (CASB) Uygulaması

Bulut Erişim Güvenlik Aracısı (CASB), bulut hizmetleri (özellikle SaaS) ile kullanıcılar arasında bir güvenlik politika uygulama noktası görevi görür. CASB'ler, kuruluşlara bulut hizmetleri üzerindeki kontrolü geri verir ve görünürlük eksikliği, veri kaybı ve kötü amaçlı yazılım gibi riskleri ele alır.

Bir CASB'nin temel fonksiyonları şunlardır:

- **Uygulama Görünürlüğü ve Kontrolü:** Çalışanların hangi bulut uygulamalarını kullandığını keşfeder ve onaylanmamış (Shadow IT) uygulamalara erişimi engeller. Bu, riskleri yöneterek üretkenliği güvenli bir şekilde sürdürmeye yardımcı olur.
- **Veri Kaybı Önleme (DLP):** Hassas verilerin bulut uygulamalarına yüklenmesini veya bu uygulamalardan sızdırılmasını engeller. Bu, hem depolanan (at-rest) hem de aktarılan (in-transit) veriler için politikalar belirlemeyi içerir.
- **Bulut Kötü Amaçlı Yazılım Koruması:** Onaylanmış bulut dosya depolama uygulamalarındaki kötü amaçlı yazılımları algılar ve kaldırır, enfekte olmuş dosyaların indirilmesini ve paylaşılmasını önler.

CASB, güvenlik hizmeti sınırı (SSE) mimarisinin temel bir parçasıdır ve güvenli erişim hizmeti sınırı (SASE) mimarisine doğru bir evrimde kritik rol oynar. Bu araçlar, modern, hibrit çalışma ortamlarında uzaktan çalışanlar için kurumsal ağ dışındaki bulut uygulamalarını güvenli hale getirmenin anahtarıdır.

5.2.5 Çoklu Bulut Güvenlik Yönetimi ve Yönetişimi

Çoklu bulut ortamları, farklı kimlik yönetim sistemlerini, API'leri ve güvenlik hizmetlerini bir araya getirir. Etkili bir yönetişim için merkezi bir bakış açısı şarttır. Her bir bulut sağlayıcısının kendine özgü bir güvenlik araç ve hizmet setine sahip olması, çoklu bulut ortamlarında tutarlılık sağlamayı zorlaştırır. Bu zorluk, CSPM, CWPP ve CASB gibi platformların ortaya çıkmasına neden olmuştur. Bu platformlar, farklı bulut ortamlarının üzerine bir "soyutlama katmanı" inşa ederek tek bir kontrol noktasından yönetişimi mümkün kılar. Bu, bulut güvenliğindeki en önemli mimari trendlerden biridir. Güvenlik, artık her bir bulut ortamında ayrı ayrı yönetilmesi gereken bir mesele değil, tüm ekosistemde tutarlılık gerektiren bir disiplin olarak ele alınmaktadır.

5.3 Konteyner ve Kubernetes Güvenliği

Konteynerler ve Kubernetes, modern, bulut yerel (cloud-native) uygulamaların omurgasını oluşturmaktadır. Bu teknolojiler, hızlı geliştirme ve dağıtım döngüleri sağlarken, geleneksel güvenlik modellerinin karşılamadığı benzersiz güvenlik zorluklarını da beraberinde getirmektedir. Bu bölüm, konteyner ve Kubernetes yaşam döngüsünün her aşamasında güvenliğin nasıl sağlanacağını detaylandırmaktadır.

5.3.1 Konteyner ve Sanallaştırma: Güvenlik Farklılıkları

Sanal makinelerin (VM) aksine, konteynerler ana işletim sisteminin (host OS) çekirdeğini (kernel) paylaşır. Bu, bir konteynerdeki zafiyetin tüm ana sisteme veya diğer konteynerlere yayılması (container escape) riskini artırır. Konteynerler efemer (geçici) ve dinamik bir yapıya sahiptir; saniyeler içinde başlatılıp durdurulabilirler. Bu geçici yapı, statik güvenlik önlemlerinin yetersiz kalmasına neden olur ve sürekli izleme gereksinimini artırır.

5.3.2 Konteyner İmaj Güvenliği ve Zafiyet Taraması

Bir konteyner imajı, uygulama kodunu, çalışma zamanı ortamını ve bağımlılıkları içeren, salt okunur bir şablondur. İmaj güvenliği, uygulamanın çalışmaya başlamadan önce, yani geliştirme ve inşa aşamalarında güvenlik açıklarının taranmasını (shift-left) içerir.

En İyi Uygulamalar ve Adımlar:

1. **Güvenilir Taban İmajı Kullanımı:** İmajlar, yalnızca Alpine veya Red Hat UBI gibi minimum boyutlu ve güvenilir taban imajlarından oluşturulmalıdır. Bu, potansiyel zafiyetleri ve saldırı yüzeyini azaltır.
2. **Otomatik Tarama Entegrasyonu:** İmajlar, geliştirme döngüsünün (SDLC) erken aşamalarında, CI/CD hattına entegre edilmiş tarama araçlarıyla sürekli taranmalıdır. `Trivy`, `Grype` veya `Anchore` gibi araçlar, imajı bilinen zafiyetler (CVE) veritabanlarına göre analiz eder.
3. **İmajların İmzalanması:** İmajların bütünlüğünü ve kimliğini doğrulamak için imza mekanizmaları kullanılmalıdır. Bu, kötü niyetli veya kurcalanmış imajların dağıtımını engeller.

5.3.3 Konteyner Çalışma Zamanı (Runtime) Güvenliği ve Davranışsal İzleme

Çalışma zamanı (runtime) güvenliği, konteynerler aktif olarak çalışırken onları korumaya odaklanır. Statik güvenlik, imajları incelerken, çalışma zamanı güvenliği gerçek zamanlı olarak şüpheli davranışları izler ve tespit eder. Bu yaklaşım, derleme veya dağıtım aşamalarından kaçan tehditlere karşı kritik bir savunma katmanı sunar.

Davranışsal İzleme Mekanizması:

- **Dinamik Taban Çizgisi:** Makine öğrenimi algoritmaları, normal uygulama davranışını (örneğin CPU kullanımı, ağ trafiği, dosya erişim kalıpları) öğrenir ve bu taban çizgisinden sapmaları (anomalileri) tespit eder. Bu yaklaşım, bilinmeyen veya sıfırıncı gün (zero-day) saldırılarını yakalamada özellikle etkilidir.
- **Tespit Edilen Tehdit Örnekleri:** Kötü amaçlı kod yürütme, ayrıcalık yükseltme saldırıları ve yapılandırma kayması (configuration drift) gibi tehditler bu yöntemle yakalanabilir. Örneğin, bir web sunucusu konteynerinin normalde bir kabuk (shell) başlatması veya hassas sistem dosyalarına erişmesi beklenmez. Davranışsal izleme, bu tür anormal aktiviteleri anında tespit ederek bir saldırı girişimi hakkında uyarı verir.
- **Önem:** Konteynerlerin ana sistem çekirdeğini paylaşması nedeniyle, bir ayrıcalık yükseltme saldırısı tüm ana sistemi tehlikeye atabilir. Bu nedenle, ayrıcalık değişikliklerini ve şüpheli sistem çağrılarını izlemek hayati önem taşır.

5.3.4 Kubernetes Güvenliği: RBAC, Ağ Politikaları, Pod Güvenliği

Kubernetes, konteyner orkestrasyonunun temel aracıdır. Güvenliği, küme katmanı, uygulama katmanı ve ağ katmanında çok yönlü bir yaklaşım gerektirir.

- **Rol Tabanlı Erişim Kontrolü (RBAC):**
  - **Teori:** RBAC, Kubernetes API'si ve küme kaynaklarına kimin (kullanıcılar, hizmet hesapları) erişebileceğini ve hangi eylemleri gerçekleştirebileceğini kontrol eder. En az ayrıcalık (Least Privilege) prensibini uygulamak için en temel mekanizmadır.
  - **Uygulama (YAML Örneği):**
    - Bir `Role` objesi, bir `namespace` içindeki izinleri tanımlar. Aşağıdaki örnek, `default` `namespace`'indeki `pods` kaynaklarına `get`, `watch` ve `list` eylemlerini okuma yetkisi verir:

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

    - Bir `RoleBinding` objesi, tanımlanan `Role`'ü bir kullanıcıya veya hizmet hesabına atar. Aşağıdaki örnek, `pod-reader` rolünü `ci-bot` adlı bir `ServiceAccount`'a bağlar:

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: ServiceAccount
  name: ci-bot
  namespace: default
```

```
roleRef:
  kind: Role
  name: pod-reader
  apiGroup: rbac.authorization.k8s.io
```

- **Ağ Politikaları (Network Policies):**
  - **Teori:** Varsayılan olarak, Kubernetes'te podlar arasında sınırsız iletişim vardır. Ağ politikaları, podlar arası iletişimi kısıtlayarak saldırı yüzeyini azaltmaya yardımcı olur. Bu, hizmetleri mikro segmentlere ayırarak yanal hareket (lateral movement) riskini azaltır.
  - **Uygulama (YAML Örneği):**
    - Aşağıdaki örnek, yalnızca `access: true` etiketi olan podların, `app: nginx` etiketi olan bir servisle iletişim kurmasına izin veren bir ağ politikası tanımlar:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: access-nginx
spec:
  podSelector:
    matchLabels:
      app: nginx
  ingress:
  - from:
    - podSelector:
        matchLabels:
          access: "true"
```

- **Pod Güvenlik Standartları (PSS):** Pod Güvenlik Politikaları'nın (PSP) yerini alan PSS, podların güvenlik seviyelerini (ayrıcalıklı, temel, kısıtlı) tanımlayan önceden belirlenmiş bir yaklaşımdır. Bu, yöneticilerin podların güvenlik yapılandırmalarını daha basit ve tutarlı bir şekilde uygulamalarını sağlar.

5.3.5 Servis Mesh Güvenliği (Istio, Linkerd) Uygulamaları

Mikro hizmet mimarilerinde hizmetler arası iletişimi yönetmek için bir servis ağı (service mesh) kullanılır. Güvenlik açısından, bir servis ağı, sıfır güven (zero-trust) ilkelerini uygulama ve trafik güvenliğini yönetme için ideal bir platformdur.

- **Temel Güvenlik Özellikleri:**
  - **Karşılıklı TLS (mTLS):** Istio ve Linkerd, hizmetler arası tüm iletişimi otomatik olarak mTLS ile şifreler, bu da uygulama kodunda herhangi bir değişiklik yapmadan veri trafiğinin güvenliğini sağlar.
  - **Politika Yönetimi:** Servis ağları, hizmetler arasında hangi iletişimin (örneğin, "X servisi sadece Y servisiyle konuşabilir") izin verildiğini tanımlayan politikalar oluşturulmasına olanak tanır.
- **Istio vs. Linkerd Karşılaştırması:**
  - **Istio:** Google, IBM ve Lyft tarafından geliştirilmiş olup, daha zengin bir özellik setine ve daha geniş bir ekosisteme sahiptir.
  - **Linkerd:** Daha basit, daha hafif ve performansa odaklanmıştır. Ancak, bazı gelişmiş güvenlik ve trafik yönetimi özelliklerinden yoksundur ve yalnızca Kubernetes'i desteklemesi gibi sınırlamaları vardır.

Servis ağları, ağ güvenliğini bir kodlama probleminden bir altyapı sorununa dönüştürür. Bu, güvenliğin bir yan görev olmaktan çıkıp, altyapı katmanında otomatik olarak uygulandığı, "kodla güvenlik" (security-as-as-code) trendinin bir parçasıdır.

5.4 Sunucusuz Güvenlik ve Hizmet Olarak Fonksiyon (FaaS)

Sunucusuz mimariler, altyapı yönetimini tamamen bulut sağlayıcısına bırakarak geliştirici çevikliğini artırır. Ancak, bu yeni model geleneksel güvenlik kontrollerinin öngöremediği benzersiz zorlukları beraberinde getirir.

5.4.1 Sunucusuz Mimarideki Benzersiz Güvenlik Zorlukları

- **Genişleyen Saldırı Yüzeyi:** Sunucusuz bir uygulama, düzinelerce veya yüzlerce küçük fonksiyondan oluşabilir. Her bir fonksiyon, HTTP API'leri, veri depoları veya IoT cihazları gibi farklı olay kaynaklarından girdi alabilir. Bu, saldırı yüzeyini geleneksel uygulamalara göre önemli ölçüde artırır.
- **Aşırı Ayrıcalıklı Fonksiyonlar:** Fonksiyonlar arası etkileşimin karmaşıklığı nedeniyle, geliştiriciler bazen zaman baskısıyla fonksiyonlara ihtiyaç duyduklarından daha fazla izin verebilir. Bu durum, bir saldırganın tek bir zafiyeti kullanarak tüm sisteme yayılmasına olanak tanıyan bir güvenlik açığı oluşturur.
- **Olay Verisi Enjeksiyonu:** Kullanıcıdan gelen verilerin (örneğin, bir HTTP isteği gövdesi veya dosya adı) doğru şekilde doğrulanmadığı durumlarda, bir saldırgan kötü amaçlı kod enjekte edebilir. Sunucusuz mimaride, bu durum "olay enjeksiyonu" olarak adlandırılır ve uygulamanın beklenmeyen şekilde davranmasına neden olabilir.
- **"Denial of Wallet" (DoW) Saldırıları:** Sunucusuz modelde tüketim bazlı ödeme yapıldığından, bir saldırgan aşırı fonksiyon çağrımı yaparak veya fonksiyonları beklenenden daha uzun süre çalışmaya zorlayarak kuruluşun maliyetlerini artırabilir. Bu, DoS saldırılarının sunucusuz ortama özgü bir varyantıdır.

5.4.2 Fonksiyon Düzeyi Güvenlik ve Kod Enjeksiyonu Önleme

Fonksiyon düzeyindeki güvenlik, her bir fonksiyonu bağımsız bir güvenlik sınırı olarak ele alır. Kod enjeksiyonu, kullanıcı girdisinin güvenli bir şekilde işlenmemesinden kaynaklanan yaygın bir tehdittir.

Pratik Yönergeler ve Örnekler:

1. **Giriş Doğrulama (Input Validation) ve Temizleme (Sanitization):** Tüm kullanıcı girdisi (form alanları, HTTP başlıkları, çerezler vb.) güvenilir kabul edilmemeli ve kullanılmadan önce doğrulanmalıdır.
   - **Senaryo:** AWS Lambda'da bir Node.js fonksiyonu ile gelen JSON verisini doğrulama.
   - **Adım 1:** Gelen veriyi (event body) doğrulamak için `jsonschema` gibi bir kütüphane kullanılabilir.
   - **Adım 2:** Giriş verisinin beklenen yapıda olduğunu garanti eden bir JSON şeması oluşturun. Şema, gerekli alanları, veri tiplerini ve ek özelliklere izin verilip verilmeyeceğini tanımlar.

     ```
     const schema = {
       "id": "/Order",
       "type": "object",
       "properties": {
         "order_id": {"type": "string"},
         "amount": {"type": "number"},
         "item": {"type": "string"}
       },
       "required": ["order_id", "amount", "item"],
       "additionalProperties": false
     };
     ```

   - **Adım 3:** Gelen veriyi bu şemaya göre doğrulayın. Eğer doğrulama başarısız olursa, fonksiyonun yürütülmesini durdurun ve anlamlı bir hata mesajı döndürün. Örneğin, `amount` alanı sayı yerine bir metin olarak gelirse, `jsonschema` bir doğrulama hatası fırlatacaktır.
2. **Parametrelendirilmiş Sorgular:** SQL enjeksiyonunu önlemek için, kullanıcı girdilerini doğrudan SQL sorgularına eklemek yerine parametreli sorgular (prepared statements) kullanılmalıdır. Bu yöntem, girdiyi veri olarak değerlendirir ve çalıştırılabilir kod olarak yorumlamaz.

5.4.3 Olay Odaklı (Event-driven) Güvenlik ve Tetikleyici Doğrulaması

Sunucusuz mimariler, bir veritabanında değişiklik yapılması veya bir S3 kovasına dosya yüklenmesi gibi olaylar tarafından tetiklenir. Güvenlik, bu olayların ve tetikleyicilerin güvenli bir şekilde doğrulanmasını gerektirir. Olay yönlendiricileri (event routers), olayları yayınlayabilecek ve onlara abone olabilecek kullanıcı ve kaynakları kısıtlamak için rol tabanlı erişim kontrolü (RBAC) politikaları uygulanabilir. Bu, güvenlik kontrollerinin geleneksel ağ topolojilerinden ziyade, veri akışına ve olay zincirlerine dayalı hale geldiğini göstermektedir. Bu mimari, bileşenler arasındaki bağımlılığı azaltarak uygulamaların daha esnek olmasını sağlar.

5.4.4 Sunucusuz Uygulama İzleme ve Loglama

Sunucusuz fonksiyonlar efemer (kısa ömürlü) ve durumsuzdur. Bu, geleneksel izleme ve loglamayı zorlaştırır, çünkü bir fonksiyonun yaşam süresi milisaniyeler olabilir ve bağlamı korumak zorlaşır.

İzleme ve Loglama için Pratik Yaklaşımlar:

- **Merkezi Loglama:** Tüm fonksiyonların logları, bulut sağlayıcının (örneğin AWS CloudWatch, Google Cloud Logging) veya üçüncü parti bir izleme aracının merkezi bir loglama hizmetine gönderilmelidir.
- **Korelasyon Kimlikleri:** Asenkron çağrılan fonksiyonlar arasında aynı isteğe ait log girdilerini birleştirmek için korelasyon kimlikleri kullanılmalıdır. Bu, bir uygulamanın uçtan uca davranışını izlemeyi mümkün kılar.
- **Metrik İzleme:** Sunucusuz izleme çözümleri, fonksiyon sağlığı ve bellek kullanımı gibi metrikleri izleyerek, uygulama darboğazlarını ve maliyet sorunlarını belirlemeye yardımcı olur.

Sunucusuz ortamlar, ”yüksek düzeyde gözlemlenebilirlik” (deep observability) ihtiyacını artırmıştır. Geleneksel log toplama yetersizdir; bir işlem akışının başlangıcından sonuna kadar tüm mikro hizmetleri ve olayları bağlamsal olarak izleyebilen araçlar zorunluluk haline gelmiştir. Bu, sunucusuz güvenliğin maliyet yönetimi ve performans optimizasyonuyla doğrudan ilişkili olduğu bir alandır.

5.4.5 Sunucusuz Mimaride Üçüncü Parti Entegrasyon Güvenliği

Sunucusuz uygulamalar genellikle üçüncü parti hizmetlere (örneğin, API'ler, veri tabanları, ödeme ağ geçitleri) entegre olur. Bu entegrasyonlar, potansiyel güvenlik riskleri taşır. Güvenliği sağlamak için aşağıdaki önlemler alınmalıdır:

- **Sıkı Erişim Kontrolleri:** Üçüncü parti hizmetlere erişim için en az ayrıcalık ilkesi uygulanmalı, yalnızca gerekli izinler verilmelidir.
- **Gizli Veri Yönetimi:** API anahtarları gibi gizli veriler, kodun içine gömülmek yerine AWS Secrets Manager veya Azure Key Vault gibi özel gizlilik yönetimi hizmetlerinde saklanmalıdır.
- **Sürekli İzleme ve Loglama:** Üçüncü parti entegrasyonlarındaki anormal davranışlar, sürekli izleme ve loglama mekanizmalarıyla takip edilmelidir.

5.5 Bulut Kimlik ve Erişim Yönetimi (CIAM)

Kimlik ve erişim yönetimi (CIAM), bulut güvenliğinin temel direklerinden biridir. Bu bölüm, buluta özgü IAM modellerini, federasyon kavramını ve ayrıcalıklı erişim yönetimi gibi gelişmiş konuları ele alacaktır.

5.5.1 Buluta Özgü Kimlik Sağlayıcıları (AWS IAM, Azure AD, GCP IAM) Karşılaştırması

Üç büyük bulut sağlayıcısı da benzer hedeflere (kimin neye ve hangi koşullarda erişebileceğini kontrol etme) sahip olsa da, her birinin IAM çerçevesi kendi platform mimarisini ve felsefesini yansıtan benzersiz bir yapıya sahiptir. Bu farklılıkları anlamak, çoklu bulut ortamlarında güvenli ve yönetilebilir erişim kontrolleri tasarlamak için kritik öneme sahiptir.

Tablo 5.2: Büyük Bulut Sağlayıcılarında Kimlik ve Erişim Yönetimi (IAM) Karşılaştırması

Özellik	**AWS IAM**	**Azure IAM (Microsoft Entra ID)**	**GCP IAM**
Mimari Modeli	Düz ancak esnek bir yapı; politikalar doğrudan kullanıcı, grup veya rollere eklenir. Hassas ve ayrıntılı izinler sunar.	Geleneksel kurumsal yapılara benzer, dört katmanlı hiyerarşik model (Yönetim Grubu > Abonelik > Kaynak Grubu > Kaynak).	Basit hiyerarşik kaynak ağacı (Kuruluş > Klasör > Proje > Kaynak). İzinler bu ağaçta aşağı doğru akar.
Rol ve İlke Yönetimi	İzinleri tanımlayan JSON tabanlı politikalar (Policy) kullanılır. Politika Simülatörü ve Erişim Analiz Aracı gibi araçlarla yapılandırma hataları önlenir.	Rol Tabanlı Erişim Kontrolü (RBAC) ve yerleşik roller (örneğin Katkıda Bulunan, Okuyucu) kullanılır. Yetkilendirme için Koşullu Erişim politikaları uygulanabilir.	Üyelere (kullanıcılar, gruplar) bir kaynak üzerinde rol atayan ”bağlantılar” (bindings) kullanılır. ”IAM Koşulları” (Conditions) ile bağlama dayalı (zaman, IP vb.) izinler belirlenebilir.
Geçici Erişim Mekanizmaları	Geçici kimlik bilgileri için rollerin ”üstlenilmesi” (assume role) mekanizması kullanılır. Bu, uzun süreli kimlik bilgilerine olan ihtiyacı azaltır.	Privileged Identity Management (PIM) hizmeti ile ayrıcalıklı erişim için just-in-time (JIT) erişim ve onay iş akışları sağlanır.	Belirli bir kapsamdaki izinlere sahip hizmet hesapları ve kısa ömürlü belirteçler (token) kullanılır.
Örnek Senaryo	Bir EC2 örneğine bir S3 kovasına erişmesi için belirli bir rol atanması.	Bir kullanıcının yalnızca belirli bir abonelik altındaki kaynak gruplarına kaynak dağıtma izninin verilmesi.	Bir projedeki tüm kullanıcıların, projenin altındaki tüm kaynaklara yalnızca okuma iznine sahip olması.

Bu üç platformun her birinin kendine özgü bir mimari ve yönetim modeline sahip olması, çoklu bulut ortamlarında tutarlı bir kimlik politikası oluşturmanın en büyük zorluklarından biridir. Bu durum, birleşik bir kimlik federasyonu stratejisini zorunlu kılmaktadır.

5.5.2 Federasyon ve Tek Oturum Açma (SSO) Kavramları ve Uygulaması

- **Tanım:** Tek Oturum Açma (SSO), kullanıcıların tek bir kimlik bilgisi seti ile aynı etki alanı (domain) içindeki birden fazla uygulamaya erişmesini sağlar. Federasyon ise bu kavramı genişleterek, farklı etki alanları veya kuruluşlar (örneğin bir çalışan portalı ve üçüncü parti bir SaaS uygulaması) arasında bir güven ilişkisi kurar.

- **Nasıl Çalışır:** Bir kullanıcı, kimlik sağlayıcısına (IdP) bir kez oturum açtığında, IdP kullanıcıyı doğrular ve bir güvenlik onayı (assertion) gönderir. Hizmet sağlayıcısı (SP), bu onayı doğrulayarak kullanıcıya tekrar oturum açmasına gerek kalmadan erişim izni verir. Bu süreç, SAML, OAuth ve OpenID Connect gibi standart protokollerle mümkün olur.

- **Faydaları:** Federasyonun ana değeri, farklı kuruluşlar arasında iş birliğini basitleştirmesi ve IT ekiplerinin harici

hizmetler için manuel hesap yönetimi yükünü azaltmasıdır. Bu, özellikle çoklu bulut ve SaaS uygulamalarının yaygınlaştığı bulut çağında tedarik zinciri güvenliği için kritik bir adımdır.

5.5.3 Bulut Ortamlarında Ayrıcalıklı Erişim Yönetimi (PAM)

Ayrıcalıklı Erişim Yönetimi (PAM), kuruluşları siber tehditlere karşı korumak için kritik kaynaklara (root hesaplar, yöneticiler, veritabanları vb.) ayrıcalıklı erişimi izleyen, tespit eden ve engelleyen bir kimlik güvenliği çözümüdür.

- **Modern PAM Yaklaşımları:**
  - **En Az Ayrıcalık (Least Privilege):** Bir kullanıcıya veya sürece, görevini yerine getirmesi için kesinlikle gerekli olan en az ayrıcalık verilmelidir.
  - **Sıfır Duran Ayrıcalık (Zero Standing Privileges) ve Just-in-Time (JIT) Erişimi:** Kullanıcılara ayrıcalıklar kalıcı olarak verilmez. Bunun yerine, erişim yalnızca belirli bir süre (örneğin 1 saat) için, bir onay süreci veya gerekçe belirtme sonrasında dinamik olarak atanır. Azure'ın PIM (Privileged Identity Management) hizmeti, bu yaklaşımı uygulamak için bir örnek teşkil eder. Kullanıcılar, rolleri etkinleştirmek için istekte bulunur ve onaylandıktan sonra belirlenen süre boyunca bu role sahip olurlar. Süre dolduğunda, ayrıcalık otomatik olarak iptal edilir.
- **Pratik Yönergeler:**
  - **Çok Faktörlü Kimlik Doğrulama (MFA):** Tüm ayrıcalıklı hesaplar için MFA zorunlu hale getirilmelidir. Bu, kimlik bilgilerinin çalınması durumunda bile ek bir güvenlik katmanı sağlar.
  - **Otomasyon:** Erişim ve yetkilendirme süreçlerini otomatikleştirmek, insan hatası riskini azaltır ve verimliliği artırır.
  - **Sürekli İzleme ve Denetim:** Ayrıcalıklı hesap faaliyetleri sürekli olarak izlenmeli ve anormallikleri tespit etmek için denetlenmelidir.

Bulut ortamlarında PAM, geleneksel on-premise çözümlerden daha güvenli olabilir. Otomatik parola döndürme, MFA ve JIT erişim mekanizmaları, insan hatası riskini azaltır ve saldırganların çalınan kimlik bilgilerini kullanmasını engeller.

5.5.4 Kimlik Yönetişimi ve Otomatik Sağlama (Provisioning)

- **Tanım:** Kimlik yönetişimi, kimlik ve erişim haklarının, politikalar ve iş kurallarıyla uyumlu olmasını sağlar. Otomatik sağlama, yeni kullanıcılara veya rol değişikliklerine göre erişim haklarının otomatik olarak atanması veya kaldırılması sürecidir.
- **Faydaları:**
  - **Gelişmiş Güvenlik:** Manuel süreçlerdeki hataları ve gözden kaçan güvenlik ayarlarını ortadan kaldırır.
  - **Hızlı Eşleme ve İşten Çıkarma:** Yeni çalışanlara ihtiyaç duydukları erişimi ilk günden itibaren sağlar, aynı zamanda işten ayrılanların erişimini otomatik olarak kaldırarak "sahipsiz hesap" (orphaned accounts) riskini en aza indirir.
  - **Etkili Uyumluluk ve Denetim:** Sağlama faaliyetlerine ilişkin detaylı denetim kayıtları (audit logs) oluşturarak, düzenleyici gerekliliklere uyumu kolaylaştırır.

Otomatik sağlama, PAM ve SSO gibi teknik kontrollerin etkili bir şekilde çalışması için gerekli operasyonel altyapıyı sağlar. Kimlik ve erişim yönetiminin güvenliğini ve uyumluluğunu en üst düzeye çıkarmak için teknik kontroller ile operasyonel süreçlerin entegrasyonu şarttır. Bu, "güvenliğin bir ürün olmadığı, bir süreç olduğu" yönündeki temel ilkeyi destekler.

5.6 Bulut Veri Koruma ve Şifreleme

Veri, bulut bilişimin en değerli varlığıdır ve modern bulut güvenliğinin temelini oluşturur. Bu bölüm, verilerin yaşam döngülerinin farklı aşamalarında (durumda, aktarımda ve kullanımda) korunma stratejilerini ve anahtar yönetimi uygulamalarını ele almaktadır.

5.6.1 Bulut Veri Sınıflandırma ve Etiketleme Stratejileri

- **Teori:** Veri sınıflandırması, verilerin hassasiyetine, gizliliğine ve iş üzerindeki etkisine göre kategorize edilmesidir. Bu, hangi verilerin daha yüksek düzeyde korunması gerektiğini belirleyen temel bir yönetişim adımıdır.
- **Faydaları:** Doğru sınıflandırma, erişim kontrolünü daha etkin yönetmeye, şifreleme politikalarını zorunlu kılmaya ve düzenleyici uyumluluğu sağlamaya yardımcı olur.
- **Pratik Adımlar:**
  1. **Hedeflerin Tanımlanması:** Sınıflandırmanın amacını belirleyin (uyumluluk, veri güvenliği vb.).
  2. **Paydaşların Katılımı:** Sadece IT değil, aynı zamanda uyumluluk, hukuk ve iş birimi yöneticilerini de sürece dahil edin.
  3. **Çerçevenin Geliştirilmesi:** Veri kategorilerini (örneğin ”Gizli,””Hassas,””KamuyaAçık”) ve onlara atanan güvenlik kontrollerini içeren bir çerçeve oluşturun.
  4. **Otomasyonun Kullanımı:** Veri sınıflandırma ve etiketleme süreçlerini otomatik hale getirmek için teknolojiden yararlanın.

Veri sınıflandırması, teknolojik bir kontrol olmaktan çok, güvenlik stratejilerini yönlendiren bir yönetişim aracıdır. Doğru bir sınıflandırma, en az ayrıcalık, şifreleme ve veri kaybı önleme (DLP) gibi diğer güvenlik mekanizmalarının etkin bir şekilde uygulanabilmesi için temel ön koşuldur.

5.6.2 Bulut Şifrelemesi: Durumda, Aktarımda, Kullanımda

Veri şifrelemesi, bir veri güvenliği stratejisinin en önemli bileşenidir. Verinin, yaşam döngüsünün her aşamasında şifrelenmesi gerekir.

- **Durumda Şifreleme (At-rest):** Veri depolama ortamında (sabit disk, veritabanı, yedekleme) hareketsizken şifrelenmesidir. Bulut sağlayıcıları genellikle bu tür verileri varsayılan olarak şifreler.
- **Aktarımda Şifreleme (In-transit):** Veri, bir konumdan diğerine taşınırken şifrelenmesidir. Bu genellikle TLS (Transport Layer Security) veya mTLS gibi protokollerle sağlanır. Bu, verinin ağ üzerinde ele geçirilse bile okunamamasını sağlar.
- **Kullanımda Şifreleme (In-use):** Veri, işlem yapılırken, bellekte veya işlemci önbelleğinde şifrelenir. ”Gizli Bilişim” (Confidential Computing) gibi yeni teknolojiler bu alana odaklanmaktadır.

Bulut sağlayıcıları varsayılan şifreleme sunsa da, bir uzman için bunun her zaman yeterli olmadığını anlamak önemlidir. Özellikle düzenleyici uyumluluk gereksinimleri (örneğin HIPAA, PCI DSS), müşteri tarafında ek şifreleme katmanlarının uygulanmasını zorunlu kılabilir.

5.6.3 Bulut Anahtar Yönetim Hizmeti (KMS) Uygulaması

KMS, şifreleme anahtarlarının yaşam döngüsünü (oluşturma, rotasyon, devre dışı bırakma, yok etme) merkezi olarak yönetmeyi sağlayan bir bulut hizmetidir. KMS, Paylaşılan Sorumluluk Modeli'nin doğrudan bir uzantısıdır. Sağlayıcı KMS hizmetini sunarken, anahtarların politikalarını ve yönetimini kontrol etmek müşterinin sorumluluğundadır.

Anahtar Yönetim Yaklaşımları:

- **Bulut Tarafından Yönetilen Anahtarlar:** Bulut sağlayıcısı tarafından oluşturulan ve yönetilen anahtarlar, kolaylık ve düşük yönetim yükü sağlar.
- **Müşteri Tarafından Yönetilen Anahtarlar (CMEK):** Müşterinin kendi anahtarlarını yönetmesine olanak tanır. Bu, anahtar yaşam döngüsü üzerinde tam kontrol ve sorumluluk sağlar.
- **Harici Anahtar Yöneticileri (EKM):** Kuruluş, anahtarları bulut ortamı dışında yönetebilir. Bu, en yüksek düzeyde kontrol ve güven gerektiren durumlarda kullanılır.

Pratik Yönergeler ve Örnekler (GCP KMS):

- **Adım 1: Keyring (Anahtar Halkası) Oluşturma:** Anahtarlar, mantıksal bir koleksiyon olan ”keyring” içinde gruplandırılır.

```
gcloud kms keyrings create "my-keyring" --location "global"
```

- **Adım 2: Şifreleme Anahtarı Oluşturma:** Şifreleme amacına yönelik bir anahtar oluşturulur.

```
gcloud kms keys create "my-key" --location "global" --keyring "my-keyring" --purpose
    "encryption"
gcloud kms keys create "my-key" \
    --location "global" \
    --keyring "my-keyring" \
    --purpose "encryption"
```

- **Adım 4: Dosyayı Şifreleme Çözme:** Şifrelenmiş dosya aynı anahtar kullanılarak geri çözülür.

```
gcloud kms decrypt \
    --ciphertext-file data.txt.enc \
    --plaintext-file data-decrypted.txt \
    --location "global" \
    --keyring "my-keyring" \
    --key "my-key"
```

5.6.4 Veritabanı Şifrelemesi ve Şeffaf Veri Şifrelemesi (TDE)

- **Teori:** Şeffaf Veri Şifrelemesi (TDE), veritabanındaki hassas verileri, veritabanı dosyaları çalınsa bile okunamaz hale getirmek için şifreler. En önemlisi, veriler yetkili kullanıcı veya uygulama için ”şeffaf” bir şekilde şifrelenip çözülür; yani uygulama kodunda herhangi bir değişiklik yapılmasına gerek kalmaz.
- **Uygulama:** TDE, bulut sağlayıcısının varsayılan depolama şifrelemesine ek bir katman olarak kullanılır ve özellikle PCI DSS gibi düzenlemelere uyum sağlamak için gereklidir. Örneğin, Google Cloud SQL for SQL Server, yerleşik TDE desteği sunar ve veritabanı birincil anahtarından oluşturulan bir sertifika ile verileri şifreler.

5.6.5 Yedekleme Şifrelemesi ve Olağanüstü Durum Kurtarma Güvenliği

- **Teori:** Yedekleme, genellikle güvenlik kontrollerinin gözden kaçtığı bir aşamadır. Yedeklemeler de en az canlı veriler kadar hassastır ve saldırganlar için cazip bir hedeftir.
- **Pratik Yönergeler:** Yedekleme verileri, bulutta depolanırken her zaman şifrelenmelidir. Felaket kurtarma (DR) planları, şifrelenmiş verilerin geri yüklenmesi için gerekli anahtar yönetimi süreçlerini içermelidir. Veritabanı gibi point-in-time recovery (PITR) etkinleştirilmiş bir hizmette TDE sertifikası döndürüldüğünde, yeni bir yedek oluşturulması, sertifika kaybı durumunda geri yükleme riskini azaltmaya yardımcı olur.

Bulut ortamlarında uçtan uca veri koruması, yalnızca aktif sistemlerin değil, aynı zamanda yedekleme ve kurtarma mekanizmalarının da güvenliğini sağlamayı gerektirir. Bu, güvenlik planlamasının bir yaşam döngüsü yaklaşımıyla ele alınması gerektiğini göstermektedir.

Bölüm 6

DONANIM VE FİZİKSEL GÜVENLİK

Giriş

Donanım ve fiziksel güvenlik, siber güvenliğin temel katmanlarından birini oluşturur. Bu bölümde donanım temelli güvenlik teknolojileri, fiziksel erişim kontrolleri ve güvenlik sistemleri konularını detaylı olarak inceleyeceğiz.

```
$ sudo sbsign --key MOK.priv --cert MOK.pem \
    "/boot/vmlinuz-$(uname -r)" \
    --output "/boot/vmlinuz-$(uname -r)-signed"
```

Güvenliğin en temel ve en kritik katmanını, yani donanım güvenliğini ve fiziksel koruma sistemlerini derinlemesine incelemektedir. Yazılım ve ağ güvenliği, sağlam bir donanım temeli olmadan yetersiz kalır. Bu rapor, siber güvenlik uzmanları için tasarlanmış olup, sistemin bütünlüğünü ve gizliliğini donanım seviyesinde nasıl koruyacaklarını anlamaları için kapsamlı bir rehber sunmaktadır.

6.1 Donanım Güvenliği Temelleri ve Güven Kökü (Hardware Security Fundamentals and Root of Trust)

Donanım güvenliği, bir sistemin tüm güvenli operasyonlarının dayandığı sarsılmaz bir temel görevi görür. Bu katman, yazılım tabanlı saldırıların ötesine geçerek, bir cihazın fiziksel bütünlüğünü ve en kritik verilerinin güvenliğini sağlamak için özel olarak tasarlanmış bileşenleri içerir. Bu bileşenler, bir güven zincirinin başlangıç noktası olarak işlev görerek sistemin en alt katmanından itibaren güvenilir bir şekilde başlatılmasını garanti eder.

6.1.1 Donanım Güvenlik Modülü (HSM) Mimarisi

Bir Donanım Güvenlik Modülü (HSM), kriptografik anahtarlar gibi dijital sırları saklamak, yönetmek ve şifreleme/şifre çözme gibi kritik kriptografik işlevleri güvenli bir ortamda gerçekleştirmek için tasarlanmış fiziksel bir donanımdır. Bu modüller, genellikle bir eklenti kartı veya bir bilgisayar sunucusuna veya ağa doğrudan bağlanan harici bir cihaz şeklinde gelir. Temel mimarisi, bus probing veya fiziksel kurcalama gibi saldırıları önlemek amacıyla bir veya daha fazla güvenli kriptoişlemci çipi içerir.

Bir HSM'in en önemli güvenlik mekanizmalarından biri, kurcalanmaya karşı sunduğu koruma seviyesidir. Bu, üç ana kategoride incelenir: kurcalanma kanıtı (tamper evidence), kurcalanma direnci (tamper resistance) ve kurcalanmaya duyarlı tepki (tamper responsiveness). Kurcalanmaya duyarlı HSM'ler, fiziksel bir müdahale girişimi tespit ettiklerinde, içlerindeki anahtarları silme veya cihazı kullanılamaz hale getirme gibi yıkıcı eylemler gerçekleştirebilir. Bu seviyedeki koruma, Common Criteria (EAL4+) ve FIPS 140 gibi uluslararası güvenlik standartları ve sertifikasyonlar aracılığıyla doğrulanır. HSM'in bu fiziksel ve mantıksal koruma mekanizmaları, onu yazılım temelli saldırılara karşı neredeyse bağışık kılar.

Bu mimarinin en önemli faydası, bir uygulamanın hassas özel anahtarlarını bir web sunucusunun belleğinde açıkta bırakma riskini ortadan kaldırmasıdır. Kriptografik işlemler, verilerin saldırganlardan korunduğu HSM ortamı içinde

gerçekleşir. Bu durum, HSM'nin sadece bir depolama alanı değil, aynı zamanda güvenli bir işlem birimi olarak işlev görmesini sağlar. Bir uygulamanın, HSM'nin iç işleyişine doğrudan erişimi olmadan anahtarları kullanabilmesi, anahtar hırsızlığı ve oltalama saldırıları gibi yaygın tehditlere karşı önemli bir savunma katmanı oluşturur.

HSM'ler, yüksek değerli anahtarları koruma ihtiyacı duyulan birçok alanda kullanılır. Öne çıkan kullanım alanları şunlardır:

- **Açık Anahtar Altyapısı (PKI):** Kök sertifika yetkilisi (CA) anahtarları gibi en kritik anahtarları koruyarak tüm güven zincirinin bütünlüğünü garanti eder.
- **Ödeme Sistemleri:** Kredi ve ödeme kartı bilgilerini korumak için kullanılır ve Payment Card Industry Data Security Standards (PCI DSS) gibi endüstri standartlarına uyumu kolaylaştırır.
- **SSL Bağlantıları:** Asimetrik anahtar operasyonları için yüksek performans sunarak sunucu işlemcisi üzerindeki yükü hafifletir.
- **Blok Zinciri:** Özel anahtarların güvenliğini sağlayarak blok zinciri süreçlerinin bütünlüğünü korumada kritik bir rol oynar.

HSM'lerin bir sunucuya entegrasyonu genellikle PKCS#11 gibi standart arayüzler aracılığıyla gerçekleşir. Bu, uygulamaların donanıma özgü kodlar yazmadan HSM'in işlevselliğini kullanmasına olanak tanır.

6.1.2 Güvenilir Platform Modülü (TPM) ve Ölçülmüş Önyükleme (Measured Boot)

Güvenilir Platform Modülü (TPM), bir bilgisayarın donanım ve yazılım bütünlüğünü doğrulamak ve korumak için tasarlanmış, anakarta yerleşik bir çiptir. TPM'in en temel işlevi, bir sistemin önyükleme süreci boyunca çeşitli bileşenlerin (firmware, bootloader, işletim sistemi çekirdeği) ölçümlerini, yani kriptografik hash'lerini, güvenli ve kurcalanmaya dayanıklı Platform Konfigürasyon Kayıtları (PCR) olarak bilinen yazmaçlarda saklamaktır.

Ölçülmüş önyükleme (measured boot) süreci, bu ölçümlerin nasıl alındığını tanımlar. Sistem her bir bileşeni çalıştırmadan önce, onun hash değerini alır. Bu hash değeri, o anki PCR değeriyle birleştirilerek yeni bir hash oluşturulur ve PCR'ye kaydedilir. Bu sürece "hash zincirlemesi" (hash-chaining) denir. Bu zincirleme mekanizması, önyükleme sürecinin herhangi bir noktasında yapılan en küçük bir değişikliğin bile (örneğin bir rootkit'in enjeksiyonu) zincirin sonraki tüm aşamalarındaki PCR değerlerini tamamen değiştirmesini sağlar. Bu, bir saldırganın kötü amaçlı kodu gizlice enjekte etmesini neredeyse imkansız hale getirir, çünkü sistemin güvenli durumu anında bozulacaktır.

Bu PCR değerleri, uzaktan doğrulama (remote attestation) için kritik bir rol oynar. Bir cihaz, mevcut "sağlıklı" durumunu kanıtlamak için, TPM tarafından imzalanmış PCR değerlerini içeren bir kanıtı (quote) uzaktaki bir Sağlık Doğrulama Hizmeti'ne (Health Attestation Service) gönderebilir. Hizmet, bu kanıtı önceden tanımlanmış bir "güvenli durum" politikasıyla karşılaştırır. Eşleşme sağlanırsa, cihazın donanım ve yazılım bütünlüğünün bozulmadığı, yani sistemin güvenli bir şekilde başlatıldığı kanıtlanmış olur.

Bir sistemde TPM ve HSM'nin rolleri birbirini tamamlayıcıdır. HSM, hassas anahtarlar için bir "güvenlik kasası" işlevi görürken, TPM bir platformun bütünlüğünü doğrulayan bir "güvenilir denetçi" görevi görür. Bir sistem, önyükleme sırasındaki bütünlüğünü bir TPM aracılığıyla kanıtlayarak, bir HSM'de depolanan kriptografik anahtarlara erişim için güvenilir bir ortam haline geldiğini kanıtlayabilir. Bu, anahtar yönetimi ve platform bütünlüğü arasında bir "güven köprüsü" kurarak modern donanım tabanlı güvenlik mimarilerinin temelini oluşturur.

6.1.3 Donanım Güven Kökü (Hardware Root of Trust) ve Güvenli Önyükleme (Secure Boot) Süreci

Donanım Güven Kökü (RoT), bir sistemin tüm güvenli operasyonlarının dayandığı temeldir. Donanım tabanlı bir RoT, yazılım tabanlı saldırılara karşı bağışık olduğu için en güvenli uygulamayı temsil eder ve kriptografik anahtarları içererek yazılım için bir güven temeli oluşturur.

Bu temel, bir "güven zinciri" (chain of trust) aracılığıyla tüm sisteme yayılır. Güven zinciri, her bir donanım ve yazılım bileşeninin, bir önceki bileşen tarafından dijital olarak imzalanmış ve doğrulanmış olmasını sağlayan bir yapıdır. Bu süreç, güvenin donanıma yazılmış ilk ve kurcalanamaz kod (genellikle bir ROM'daki Birincil Önyükleme

Yükleyicisi) ile başlamasını sağlar. Bu, tüm zincirin ”güvenilir çapası”dır. Zincir, esneklik sağlamak amacıyla tasarlanmıştır, çünkü donanımın her bir yazılım parçasını kendisi doğrulaması yerine, güveni bir sonraki katmana (örneğin, önyükleme yükleyicisine) devreder.

Güvenli Önyükleme (Secure Boot) süreci, bu güven zincirinin pratik bir uygulamasıdır. Bu, donanım kökünden başlayarak, önyükleme sürecinde yalnızca güvenilir ve dijital olarak imzalanmış yazılımların (UEFI firmware, bootloader, kernel) yüklenmesini sağlayan bir güvenlik standardıdır. Bu mekanizma, önyükleme sırasında firmware rootkit'leri ve bootkit'leri gibi kötü amaçlı yazılımların sisteme sızmasını engeller.

Bir Linux sisteminde UEFI Güvenli Önyükleme uygulamasının adımları şunlardır:

1. **Gerekli Araçların Kurulumu:**

```
$ sudo apt install openssl sbsigntool efitools mokutil shim-signed grub-efi-amd64-signed
```

2. **Özel Anahtar ve Sertifika Oluşturma:**

```
$ sudo openssl req --new -x509 -newkey rsa:2048 -keyout MOK.priv -outform DER -out
    MOK.der -days 36500 -subj "/CN=ECI/"
```

3. **Çekirdeği İmzalama:**

```
$ sudo sbsign --key MOK.priv --cert MOK.pem "/boot/vmlinuz-$(uname -r)" --output
    "/boot/vmlinuz-$(uname -r)-signed"
```

4. **Sertifikayı Kaydetme (Enrollment):**

```
$ sudo mokutil --import MOK.der
```

5. **Durum Doğrulaması:**

```
$ mokutil --sb-state
```

Bu adımlar, sistemin önyükleme sürecinin yalnızca yetkili yazılımları yüklemesini garanti eder.

6.1.4 Çip Seviyesi Güvenlik Özellikleri ve Güvenli Enklavlar (Secure Enclaves)

Çip seviyesi güvenlik, bir sistemin saldırı yüzeyini en aza indirmek için özel olarak tasarlanmış donanım bileşenlerini içerir. Bu bileşenlerden biri olan Güvenli Enklav (Secure Enclave), ana işlemciden izole edilmiş özel, güvenli bir alt sistemdir. Amacı, uygulama işlemcisinin çekirdeği tehlikeye atılsa bile hassas kullanıcı verilerini güvende tutmaktır.

Güvenli Enklav'ın mimarisi, kendi özel işlemcisi, şifrelenmiş belleği, güvenli önyüklemesi ve kurcalamaya dayanıklı bir yapısı sayesinde yüksek düzeyde izolasyon sağlar. Bir örnek olarak Apple'ın Secure Enclave'i, ana işlemcinin şifresiz belleği normal bir şekilde okuyup yazmasını sağlarken, dışarıdan herhangi bir gözlemcinin sadece şifrelenmiş ve doğrulanmış bellek içeriğini görmesini garantiler. Bu, verilerin aktarımı sırasında bile gizliliği korur.

Bu özel donanım, biyometrik verilerin (Face ID, Touch ID), şifrelerin, kriptografik anahtarların ve mobil ödemelerle ilgili kritik işlemlerin güvenli bir şekilde işlenmesi ve saklanması için kullanılır. Güvenli Enklavlar, güven kökü mimarisinin bir uzantısı olarak kabul edilebilir. Güvenli Önyükleme tüm platformun bütünlüğünü sağlarken, Güvenli Enklavlar bu güvenilir platform içinde bile daha yüksek bir izolasyon ve gizlilik sunar. Bu, saldırı yüzeyini yalnızca önyükleme sürecinde değil, tüm sistemin yaşam döngüsü boyunca küçültür ve ana işlemci çekirdeğinin güvenliği ihlal edilse bile kritik verilerin güvende kalmasını sağlar.

6.1.5 Tedarik Zinciri Donanım Güvenliği ve Sahte Ürün Tespiti

Tedarik zinciri güvenliği, bir cihazın ve bileşenlerinin orijinal olduğunu, üretim ve dağıtım aşamalarında kurcalanmadığını doğrulamayı amaçlayan kritik bir alandır. Güvenilirliği kanıtlanmamış teknolojilerin kullanılması, organizasyonlar için gizli güvenlik açıklarına ve risklere yol açabilir.

Donanım bütünlüğünü doğrulamak için, cihaz üreticileri her cihaza, cihazın kimliğine güvenli bir şekilde bağlı bir "eser" (artifact) yerleştirebilir. Bu eser, cihazın attributes'larını (özelliklerini) içerebilir ve orijinalliğini teyit etmek için kullanılabilir. Müşteri, bu eserin kaynağını ve orijinalliğini doğrulayarak cihazın gerçekliğini teyit edebilir. Benzer bir süreç, cihazlar operasyonel kullanımdayken periyodik bütünlük doğrulaması için de uygulanabilir. Bu, sahte veya değiştirilmiş donanımın tespit edilmesine yardımcı olur ve güvenli bir tedarik zincirinin sürdürülmesi için kritik bir adımdır.

6.2 Donanım Yazılımı (Firmware) Güvenliği ve Düşük Seviyeli Sistem Koruması

Donanım yazılımı (firmware), bir donanım aygıtının temel işlevlerini kontrol eden ve işletim sistemi ile donanım arasında bir köprü görevi gören yazılımdır. Firmware, genellikle bir aygıtın kalıcı belleğinde saklanır ve işletim sistemi başlatılmadan önce çalışır. Bu nedenle, firmware seviyesindeki bir güvenlik açığı, işletim sistemi seviyesindeki güvenlik kontrollerini atlatabilir ve tespit edilmesi zor, kalıcı bir tehdit oluşturabilir.

6.2.1 UEFI/BIOS Güvenliği ve Güvenli Önyükleme Uygulaması

UEFI (Unified Extensible Firmware Interface) Güvenli Önyükleme, bir cihazın yalnızca güvenilir ve dijital olarak imzalanmış yazılımları kullanarak önyükleme yapmasını sağlayan bir güvenlik standardıdır. Bu, saldırganların sistem yazılımını değiştirerek yetkisiz veya kötü amaçlı kod çalıştırmasını engeller.

Bir Linux sisteminde bu standardın uygulanması, kendi sertifikanızı oluşturmayı ve önyükleme bileşenlerini bu sertifikayla imzalamayı içerir. Aşağıdaki adımlar, bu sürecin bir örneğini sunar:

1. **Gerekli Araçların Kurulumu:**

```
$ sudo apt install openssl sbsigntool efitools mokutil shim-signed grub-efi-amd64-signed
```

   Bu komut, güvenli önyükleme için gerekli olan tüm araçları kurar.

2. **Özel Anahtar ve Sertifika Çifti Oluşturma:**

```
$ sudo openssl req --new -x509 -newkey rsa:2048 -keyout MOK.priv -outform DER -out
    MOK.der -days 36500 -subj "/CN=ECI/"
```

   Bu komut, çekirdeği imzalamak için kullanılacak yeni bir anahtar ve sertifika çifti oluşturur.

3. **Çekirdeği İmzalama:**

```
$ sudo sbsign --key MOK.priv --cert MOK.pem "/boot/vmlinuz-$(uname -r)" --output
    "/boot/vmlinuz-$(uname -r)-signed"
```

   `sbsign` komutu, çekirdeğinizi yeni oluşturulan sertifikayla imzalar.

4. **Sertifikayı Makine Sahibi Anahtarı (MOK) Olarak Kaydetme:**

```
$ sudo mokutil --import MOK.der
```

   Bu komut, sertifikanızı bir MOK olarak kaydetmek üzere sıraya alır. Sistem yeniden başlatıldığında, UEFI key manager ekranında bu sertifikayı kaydetmeyi onaylamanız gerekir.

5. **Durum Doğrulaması:**

```
$ mokutil --sb-state
```

Yeniden başlatmanın ardından, bu komut Güvenli Önyüklemenin etkinleştirildiğini onaylar.

6.2.2 Firmware Attestation ve Bütünlük Doğrulaması

Firmware attestation, bir cihazın donanım ve firmware'inin bütünlüğünü bir üçüncü tarafa uzaktan kanıtlama sürecidir. Bu süreç, cihazın güvenli bir şekilde başlatıldığından ve yetkisiz değişikliklere uğramadığından emin olmak için proaktif bir yaklaşım sunar.

Bu süreç, bir Trusted Platform Module (TPM) ile gerçekleştirilir. Cihaz önyükleme yaparken, UEFI Güvenli Önyükleme ve diğer bileşenlerin ölçümleri alınarak TPM'in Platform Konfigürasyon Kayıtları'na (PCR'ler) kaydedilir. Cihazın sağlığı, bu ölçümlerin bir Attestation Identity Key (AIK) ile imzalanarak uzaktaki bir Sağlık Doğrulama Hizmeti'ne gönderilmesiyle doğrulanır. Hizmet, ölçümleri önceden tanımlanmış bir "güvenli durum" politikasıyla karşılaştırır. Doğrulama başarılı olursa, MDM (Mobile Device Management) çözümü gibi bir hizmet, cihazın kurumsal kaynaklara erişmesine izin verir. Bu, bir saldırının hasara yol açmadan önce durdurulmasına olanak tanır, çünkü bütünlüğü bozulmuş bir cihaz ağa erişim elde edemez.

6.2.3 Bootloader Güvenliği ve Güven Zinciri (Chain of Trust)

Bootloader, işletim sistemini başlatan ilk yazılımdır ve bu nedenle tüm sistem güvenliği için hayati bir hedef noktasıdır. Güven zinciri, bootloader'ın güven kökü (RoT) tarafından doğrulanmasıyla başlar ve işletim sistemi çekirdeğine ve diğer çevre birimi firmware'lerine kadar devam eder. Bu süreçte, her bir aşama bir sonraki aşamanın bütünlüğünü kriptografik olarak doğrular. İlk aşama, çipin değişmez belleğinde (ROM) bulunan Birincil Önyükleme Yükleyicisi (PBL) tarafından gerçekleştirilir. PBL, işletim sistemi çekirdeği ve diğer çevre birimi firmware'lerini yükleyen ikincil bootloader'ın dijital imzasını doğrular.

Bir ARM mimarili gömülü sistemde güvenli bootloader'ın işleyişi aşağıdaki adımları içerebilir:

```
static void BootJump( uint32_t *Address ) {
    // Tüm interrupt'ları ve fault handler'ları kapatır
    // Kökten başlayarak işlemciyi güvenli bir duruma döndürür
    NVIC->ICER[ 0 ] = 0xFFFFFFFF;
    //...
    // Bootloader'daki SysTick'i ve diğer fault
    // handler'ları devre dışı bırakır
    SCB->SHCSR & = ~(
        SCB_SHCSR_USGFAULTENA_Msk |
        SCB_SHCSR_BUSFAULTENA_Msk |
        SCB_SHCSR_MEMFAULTENA_Msk
    );

    // Kullanıcı uygulamasının vektör tablosu
    // adresini SCB->VTOR yazmacına yükler
    SCB->VTOR = ( uint32_t )Address;

    // Uygulamanın stack pointer ve reset vector adreslerini yükler
    BootJumpASM( Address[ 0 ], Address[ 1 ] );
}
```

Bu kod, güvenli bir geçiş yaparak bootloader'dan ana uygulamaya geçişi sağlar. Güven zinciri, parçalanmış güven ortamlarında (örneğin, güvenilir yürütme ortamı TEE ve zengin yürütme ortamı REE) kritik bir rol oynar. Örneğin, birincil bootloader (PBL), TEE ve REE için bootloader'ları ayrı ayrı yükleyebilir. Bu, TEE'nin güvenliğini REE'den bağımsız olarak korur.

6.2.4 Gömülü Sistem Firmware Güvenliği

Gömülü sistemler, uzun hizmet ömürleri (20 yıldan fazla), sınırlı kaynakları ve güncellenme zorlukları nedeniyle benzersiz güvenlik riskleri taşır. Güvenliğin sonradan eklenen bir özellik değil, tasarım aşamasından itibaren entegre edilmesi esastır.

Güvenli bir yazılım geliştirme yaşam döngüsü (SDL), tehdit modelleme, güvenlik gereksinimlerinin tanımlanması ve tüm geliştirme aşamalarına güvenlik testlerinin entegre edilmesi gibi adımları içerir. Bu süreçte, fuzz testing gibi dinamik program analizleri, uygulamanın beklenmedik girdilere nasıl tepki verdiğini test ederek güvenlik açıklarını belirlemeye yardımcı olabilir.

Firmware güncellemeleri, bir sistemin yaşam döngüsü boyunca güvenli kalabilmesi için hayati öneme sahiptir. Monolitik işletim sistemlerinde (Linux gibi) güncelleme yapmak zorken, mikro çekirdekli sistemlerde (QNX gibi) tek bir hizmeti yeniden başlatmak yeterli olabilir, bu da güncelleme sürecini kolaylaştırır.

6.2.5 Firmware Güncelleme Güvenliği ve Kablosuz (OTA) Güncellemeler

Kablosuz (Over-the-Air - OTA) güncellemeler, bir işletim sistemi veya firmware'in kablosuz ağ üzerinden güncellenmesidir. Bu, güncellemelerin büyük ölçekte ve düşük maliyetle dağıtılmasını sağlar. Ancak, OTA güncellemelerinin güvenliği, kötü amaçlı firmware enjeksiyonlarını önlemek için kritik öneme sahiptir.

Güvenli bir OTA sistemi, güncelleme paketinin bütünlüğünü ve orijinalliğini sağlamak için kriptografik imzalar ve hash'ler kullanmalıdır. Süreç şu adımları içerir:

1. **Anahtar Çifti Oluşturma:** Dağıtım sistemi üzerinde bir genel/özel anahtar çifti oluşturulur.
2. **Genel Anahtarı Cihaza Yükleme:** Genel anahtar, cihazın ilk firmware sürümüne yerleştirilir. Bu, cihazın güncellemelerin orijinalliğini doğrulaması için bir güven temeli oluşturur.
3. **Güncelleme Paketinin İmzalanması:** Yeni firmware paketi, özel anahtar kullanılarak imzalanır ve bir hash (HMAC) ile bütünlüğü sağlanır.
4. **Cihazda Doğrulama:** Cihaz, güncelleme paketini aldığında, hash'i kontrol ederek bütünlüğü ve imzayı kontrol ederek orijinalliği doğrular.
5. **Atomik Değiştirme:** Güncellemenin bütünlüğü onaylandıktan sonra, mevcut firmware'in yeni firmware ile atomik (bölünemez) olarak değiştirilmesi sağlanır. Bu, genellikle iki ayrı depolama bölümü kullanılarak gerçekleştirilir. Bir bölüm mevcut firmware'i barındırırken, diğer bölüm yeni güncellemeyi alır. Güncelleme başarıyla kurulduğunda, bootloader yeni bölüme geçiş yapar.

Güvenli OTA, sadece bir yazılım güncelleme mekanizması değil, aynı zamanda cihazın yaşam döngüsü güvenliğinin merkezinde yer alan bir bileşendir. Bir cihaz operasyonel ömrü boyunca maruz kaldığı tehditler değiştiği için, güvenli OTA, bu değişen tehdit ortamına karşı savunma kabiliyetini sürekli olarak korumayı sağlar.

6.3 Fiziksel Güvenlik Kontrolleri ve Erişim Yönetimi

Fiziksel güvenlik, siber güvenlik stratejisinin temel bir bileşenidir ve dijital varlıkları yetkisiz fiziksel erişime, hırsızlığa veya hasara karşı korumayı amaçlar. Fiziksel güvenlik kontrolleri, bir kuruluşun tesislerine, veri merkezlerine ve diğer kritik altyapılarına erişimi kısıtlamak ve izlemek için tasarlanmıştır. Bu kontroller, caydırıcı, önleyici, tespit edici ve düzeltici önlemleri içerir.

6.3.1 Biyometrik Erişim Kontrol Sistemleri

Biyometrik erişim kontrol sistemleri, parmak izi, yüz veya iris desenleri gibi benzersiz biyolojik özellikleri kullanarak kimlik doğrulama yapar. Geleneksel kimlik kartları veya şifrelerin aksine, biyometrik verinin kaybolması, çalınması veya klonlanması son derece zordur.

Bir biyometrik sistemin işleyişi şu adımları içerir:

1. **Veri Yakalama:** Sisteme kayıtlı kullanıcıların fiziksel özellikleri, sofistike tarayıcılarla yakalanır. Örneğin, bir parmak izi tarayıcı, parmak izinin ızgara çizgilerini gösteren üç boyutlu bir görüntü alır.
2. **Şablon Oluşturma:** Yakalanan verinin benzersiz bir matematiksel şablona dönüştürülmesi ve güvenli, şifrelenmiş bir veritabanında saklanması. Bu, verinin gizliliğini korumak için kritik öneme sahiptir.
3. **Karşılaştırma ve Doğrulama:** Okuyucunun yakaladığı anlık verinin, veritabanındaki şablonla karşılaştırılması. Eşleşme sağlanırsa, sistem kapının kilidini açar.

Biyometrik sistemler, yetkisiz bir kişinin bir başkasının kartını veya şifresini kullanarak erişim sağlamasını, yani "buddy punching" gibi hileleri fiziksel olarak engellediği için geleneksel sistemlere göre önemli avantajlar sunar. Ancak, bu sistemler de risklerden muaf değildir. Biyometrik verilerin çalınması, sahte parmak izleri veya yüz maskeleri gibi yöntemlerle sistemin aldatılması gibi saldırı vektörleri mevcuttur. Bu nedenle, biyometrik güvenlik, veri şifreleme ve çok faktörlü kimlik doğrulama (MFA) gibi ek katmanlarla güçlendirilmelidir.

6.3.2 Akıllı Kart ve RFID Güvenlik Teknolojileri

Akıllı kartlar ve RFID (Radio Frequency Identification), veriyi depolamak ve iletmek için gömülü mikroçipler kullanır. Akıllı kartlar, genellikle gelişmiş şifreleme ve kimlik doğrulama yeteneklerine sahipken, RFID kartlar daha çok temel kimlik numaralarını depolar.

Bu teknolojiler, çeşitli saldırılara karşı savunmasız olabilir:

- **Yan Kanal Saldırıları (Side-Channel Attacks):** Diferansiyel Güç Analizi (DPA), bir kartın kriptografik işlemler sırasında tükettiği güç ve geçen zamanı ölçerek gizli anahtarlarını elde etmeye çalışır.
- **RFID Saldırıları:**
  - **Dinleme (Eavesdropping):** Bir saldırgan, etiket ile okuyucu arasındaki kablosuz iletişimi gizlice yakalayarak hassas bilgileri elde edebilir.
  - **Sinyal Sahteciliği (Spoofing):** Bir saldırgan, yetkili bir RFID etiketinin sinyalini taklit ederek yetkisiz erişim sağlayabilir.

Bu saldırılara karşı korunmak için şu önlemler alınmalıdır:

- **Şifreleme:** Etiket ve okuyucu arasındaki verinin şifrelenmesi, dinleme saldırılarını engeller.
- **Kimlik Doğrulama:** Sadece yetkili etiketlerin ve okuyucuların iletişim kurabilmesini sağlayan mekanizmaların uygulanması önemlidir.
- **DPA Karşı Önlemleri:** Gürültü üretme veya sızıntıyı azaltma gibi tasarımsal önlemlerle DPA saldırılarına karşı direnç sağlanabilir.

6.3.3 Fiziksel İzinsiz Giriş Tespit Sistemleri (PIDS)

Fiziksel İzinsiz Giriş Tespit Sistemleri (PIDS), bir kuruluşa ait fiziksel alana yetkisiz erişimi otomatik olarak izlemek ve tespit etmek için tasarlanmış çözümlerdir. Bu sistemler, hareket, ısı, ses, titreşim ve basınç gibi uyaranları algılayan özel sensörler içerir.

Etkili bir PIDS, bir dizi entegre güvenlik cihazından oluşur:

- **Sensörler:** Dış (perimeter) ve iç mekan (interior) erişim noktalarına yerleştirilen sensörler, anormal aktivite algıladığında anında uyarı gönderir.
- **Kameralar:** Güvenlik kameraları, olayların görsel belgelenmesini sağlar ve saldırganların kimlik tespiti için kanıt sağlar.
- **Alarmlar:** Sensörler tarafından tetiklenen alarmlar, yetkisiz girişlere anında ve duyulabilir bir yanıt sağlar.
- **Entegrasyon:** Modern PIDS'ler, erişim kontrol sistemleri, video gözetim ve bulut tabanlı yönetim platformlarıyla entegre edilerek merkezi bir güvenlik ekosistemi oluşturur.

6.3.4 Video Gözetim ve Analitik Entegrasyonu

Video analitikleri, geleneksel pasif gözetim sistemlerini aktif, gerçek zamanlı güvenlik araçlarına dönüştürür. Yapay zeka (AI) ve makine öğrenimi (ML) algoritmaları, insanları, araçları ve anormal davranışları sınıflandırarak şüpheli aktiviteleri otomatik olarak tespit eder.

Bu entegrasyon, güvenlik operatörlerine gerçek zamanlı, eyleme dönüştürülebilir istihbarat sağlar. Analitikler, önceden tanımlanmış kurallara göre otomatik yanıtları tetikleyebilir:

- Güvenlik operatörlerine mobil bildirim gönderme.
- Otomatik kapı kilitleme veya sistemleri kilit altına alma.
- İlgili nesneleri otomatik olarak takip etmesi için PTZ (Pan-Tilt-Zoom) kameraları programlama.

Video analitikleri, çevresel gürültüden kaynaklanan yanlış alarmları (örneğin, değişen hava koşulları) eleyerek operatörlerin gerçek tehditlere odaklanmasını sağlar. Bu, insan müdahalesine olan bağımlılığı azaltarak ve tepki sürelerini hızlandırarak operasyonel verimliliği artırır.

6.3.5 Çevresel İzleme ve Kurcalama Tespiti

Fiziksel güvenliğin önemli bir yönü, elektronik ekipmanın çalışmasını etkileyebilecek çevresel koşulları izlemektir. Sıcaklık, nem, duman ve su sızıntısı gibi çevresel tehditler, sistem arızalarına veya veri kaybına yol açabilir.

Kurcalama tespiti (tamper detection), bir fiziksel muhafazanın izinsiz açılma, değiştirilme veya müdahale edilme girişimini tespit eden mekanizmaları ifade eder. Bu sensörler, muhafazaya fiziksel olarak entegre edilebilir. Bir kurcalama girişimi tespit edildiğinde, sistem buna çeşitli şekillerde tepki verebilir. En kritik tepkilerden biri, şifreleme anahtarları gibi hassas güvenlik verilerinin silinmesi (key zeroization) veya tüm sistemin çalışamaz hale getirilmesidir.

6.4 Veri Merkezi ve Kritik Altyapı Güvenliği

Veri merkezleri, bir kuruluşun en değerli varlıklarını barındıran ve iş sürekliliği için hayati öneme sahip olan tesislerdir. Bu nedenle, veri merkezi güvenliği, hem fiziksel hem de siber tehditlere karşı çok katmanlı bir savunma stratejisi gerektirir. Kritik altyapı, bir toplumun veya ekonominin işleyişi için temel olan sistemleri ve varlıkları ifade eder. Bu altyapının güvenliği, ulusal güvenlik ve kamu güvenliği için kritik öneme sahiptir.

6.4.1 Veri Merkezi Fiziksel Güvenlik Mimarisi

Bir veri merkezinin fiziksel güvenliği, çok katmanlı erişim kontrolü ile sağlanır. Bu, aşağıdaki katmanları içerir:

1. **Perimeter Güvenliği:** En dış katman, 8 metrelik çitler ve 360 derecelik yüksek çözünürlüklü video gözetimi gibi fiziksel bariyerlerle korunur.
2. **Giriş Noktası Güvenliği:** Girişler, ”mantrap” (giriş tuzağı) adı verilen, bir kişinin geçtikten sonra diğerinin geçişine izin vermeyen güvenli alanlarla korunur. Ziyaretçi yönetimi ve eskort prosedürleri de bu aşamada devreye girer.
3. **İç Mekan Güvenliği:** Tesisin içine girildikten sonra bile, sunucu odaları ve kabinler için daha sıkı erişim kontrolleri uygulanır. Bu, genellikle bir kimlik kartı ve biyometrik parmak izi okuyucu gibi çift faktörlü kimlik doğrulamayı içerir.

Güvenliğin sağlanması sadece teknolojiyle sınırlı değildir. İyi eğitimli bir güvenlik ekibinin 7/24 sahada bulunması esastır. ISO 27001, HIPAA, PCI DSS gibi standartlara uyum, bu güvenliğin sürekli olarak denetlenmesini ve doğrulanmasını sağlar.

6.4.2 Güç ve Soğutma Altyapı Güvenliği

Veri merkezlerinde güç ve soğutma altyapısının güvenliği, kesintisiz çalışma (uptime) ve donanım arızalarını önlemek için kritik öneme sahiptir. Bu altyapılardaki yedeklilik, birincil bileşenlerin arızalanması durumunda bile sistemin çalışmaya devam etmesini sağlar.

Yedeklilik modelleri, veri merkezinin ihtiyaç duyduğu kesintisiz çalışma seviyesini belirler:

Model	Tanım	Tipik Kullanım Alanı	Maliyet Etkisi
N	Bare minimum. Hiç yedekleme yok.	Küçük veri merkezleri, Tier I	En Düşük
N+1	Bir yedek bileşen. Her bileşen için bir yedek.	Kurumsal Tier III veri merkezleri	Orta
2N	Tam yedekleme. Sistemin her bileşeni için tam bir kopya.	Yüksek kullanılabilirlik gerektiren Tier IV veri merkezleri	Yüksek
2N+1	Tam yedekleme + bir yedek.	Ultra-kritik siteler (finans, savunma)	Çok Yüksek

Bu tablo, veri merkezi operasyonlarında teknik bir kararın (yedeklilik modeli seçimi) nasıl finansal bir risk yönetimi kararıyla yakından ilişkili olduğunu gösterir. Yüksek yedeklilik seviyeleri, yüksek sermaye ve işletme maliyetlerine yol açar, ancak veri merkezi kesintilerinin potansiyel maliyeti (dakikada 8,000-15,000 dolar) göz önüne alındığında bu yatırımın haklı olduğu düşünülebilir.

6.4.3 Yangın Söndürme ve Acil Durum Müdahale Sistemleri

Veri merkezleri, hassas elektronik ekipmanlar içerdiği için su bazlı yangın söndürme sistemleri uygun değildir, zira su ekipmanlara onarılamaz zararlar verebilir. Bu nedenle, ”temiz ajan” (clean agent) veya gazlı söndürme sistemleri tercih edilir. Bu sistemler, yangını kalıntı bırakmadan söndüren kimyasal veya inert gazlar kullanır. FM-200 veya inergen gibi ajanlar, insan yaşamı için güvenli kabul edilir ve IT ekipmanlarına zarar vermez. Acil Durum Sesli Alarm İletişim sistemleri (EVAC), bir yangın veya başka bir acil durum sırasında bina sakinlerini yönlendirmek için kullanılır.

6.4.4 Ziyaretçi Yönetimi ve Eskort Prosedürleri

Ziyaretçi yönetimi, fiziksel güvenliğin kritik bir parçasıdır ve tesis güvenliğini sağlamak için titizlikle belirlenmiş prosedürler gerektirir. Güvenli tesislerde, ziyaretçilerin her zaman yetkili bir personel (eskort) tarafından refakat edilmesi esastır.

Ziyaretçi yönetiminin temel adımları şunlardır:

1. **Ön Hazırlık:** Ziyaretçiler gelmeden önce, yasaklı veya kontrollü eşyalar hakkında bilgilendirilmelidir.

2. **Giriş:** Ziyaretçiler, girişte kimliklerini doğrulamalı, bir ziyaretçi kartı almalı ve güvenlik taramasından geçmelidir.

3. **Eskort Prosedürü:** Eskort, ziyaretçiyi tüm ziyaret boyunca görsel olarak kontrol altında tutmalıdır. Ziyaretçinin gizli bilgilere erişmesini (konuşmaları duyma, belgeleri görme) ve DOE'ye ait mülkleri izinsiz çıkarmasını engellemek eskortun sorumluluğundadır.

6.4.5 Fiziksel Varlık Takibi ve Envanter Yönetimi

Fiziksel varlık takibi, veri merkezindeki sunucular, anahtarlar ve depolama birimleri gibi tüm donanım envanterini sürekli olarak izlemeyi ve yönetmeyi amaçlar. Bu, kayıp, hırsızlık veya yetkisiz değiştirme olaylarını önlemeye yardımcı olur.

6.5 IoT ve Gömülü Cihaz Güvenliği

Nesnelerin İnterneti (IoT) ve gömülü cihazlar, günlük yaşamın ve endüstriyel süreçlerin ayrılmaz bir parçası haline gelmiştir. Bu cihazlar, akıllı ev aletlerinden endüstriyel kontrol sistemlerine kadar geniş bir yelpazede kullanılır. Ancak, IoT cihazları genellikle sınırlı işlem gücü, bellek ve güvenlik özellikleriyle tasarlanmıştır. Bu durum, onları siber saldırılara karşı savunmasız hale getirir.

6.5.1 IoT Cihaz Kimlik Doğrulama ve Kimlik Yönetimi

IoT cihazlarının güvenliği, her bir cihazın güvenilir olduğunu kanıtlayan benzersiz bir dijital kimliğe sahip olmasını gerektirir. Açık Anahtar Altyapısı (PKI), cihaz kimliklerinin yönetimi için ”altın standart” olarak kabul edilir.

Güvenli bir kimlik yönetimi süreci aşağıdaki adımları içerir:

1. **Güvenli Provisioning:** Cihazlara benzersiz dijital sertifikalar, üretim aşamasında veya ilk kurulumda bulut tabanlı sertifika verme sistemleriyle güvenli bir şekilde sağlanır.

2. **Yaşam Döngüsü Yönetimi:** Cihazın ömrü boyunca sertifika yenileme ve tehlikeye giren sertifikaları iptal etme mekanizmaları kurulmalıdır.

6.5.2 IoT için Güvenli İletişim Protokolleri (MQTT, CoAP)

MQTT ve CoAP, kısıtlı cihazlar ve ağlar için özel olarak tasarlanmış hafif iletişim protokolleridir.

Protokol Adı	İletişim Modeli	Temel Protokol	Güvenlik Mekanizması
MQTT	Yayın-Abone (Publish-Subscribe)	TCP	SSL/TLS desteği
CoAP	İstemci-Sunucu (Client-Server)	UDP	DTLS (Datagram Transport Layer Security)

Tablo: IoT İletişim Protokolleri

MQTT Güvenli İletişim Örneği: MQTT, veri bütünlüğü ve gizliliği için SSL/TLS desteği sunar. Aşağıdaki Python kodu, bir Solace PubSub+ Event Broker'a TLS ile güvenli bir şekilde nasıl bağlanılacağını gösterir:

```
import ssl
import paho.mqtt.client as paho

def on_connect(client, userdata, flags, rc):
    print("Connect with result: " + str(rc))
    client.subscribe("test/topic", qos=1)

client = paho.Client()
client.on_connect = on_connect
client.tls_set(ca_certs='ca.crt', tls_version=ssl.PROTOCOL_TLSv1_2)
client.connect("mqtt.example.com", 8883)
client.loop_forever()
```

Bu örnek, broker sertifikasının güvenilir bir sertifika otoritesi (CA) tarafından imzalandığını doğrulamak için `ca.crt` dosyasını kullanarak TLS bağlantısını nasıl kuracağınızı gösterir.

CoAP Güvenli İletişim Örneği: CoAP, UDP üzerinde çalıştığı için temel güvenliği DTLS (Datagram Transport Layer Security) ile sağlar. DTLS, UDP üzerinden TLS'ye benzer bir şifreleme sağlar. Aşağıdaki C kodu, bir `libcoap` oturumunda DTLS'nin nasıl başlatıldığını gösterir:

```
// Libcoap ve DTLS kütüphanelerini şbalatma
coap_startup();
coap_dtls_set_log_level(LOG_NOTICE);
coap_set_log_level(LOG_NOTICE);

// Adres çözümlemesi ve DTLS oturumu şoluturma
coap_address_t server;
```

```
coap_address_init(&server);
//...
coap_session_t *session = coap_new_client_session_pki(
    NULL,
    &server,
    COAP_PROTO_DTLS,
    &dtls_pki_info
);

// Geri ğıçarma şileyicilerini (handlers) kaydetme
coap_register_response_handler(ctx, message_handler);
coap_register_nack_handler(ctx, nack_handler);
coap_register_event_handler(ctx, event_handler);
```

Bu örnekte, DTLS ile bir istemci oturumu oluşturulur ve mesajlaşma, yanıt ve hata olaylarını işlemek için geri çağırma işleyicileri kaydedilir.

6.5.3 Uç Bilişim (Edge Computing) Güvenlik Mimarisi

Uç bilişim, verinin kaynağına yakın, ağın ”ucunda” işlendiği dağıtılmış bir mimaridir. Bu mimari, gecikmeyi azaltır ve bant genişliği kullanımını düşürürken, güvenlik merkezsizleştiği için yeni riskler yaratır.

Bu risklere karşı korunmak için şu güvenlik yöntemleri uygulanır:

- **Ağ Segmentasyonu:** Uç ağların daha küçük, izole edilmiş bölümlere (zonalara) ayrılması, bir ihlalin tüm ağa yayılmasını engeller.
- **Yapay Zeka Tabanlı Tehdit Tespiti:** Anormal davranışları gerçek zamanlı olarak tespit etmek için yapay zeka ve makine öğrenimi kullanılması, geleneksel yöntemlerle tespit edilmesi zor olan tehditleri ortaya çıkarabilir.

6.5.4 Endüstriyel IoT (IIoT) Güvenlik Dikkat Edilmesi Gerekenler

Endüstriyel IoT (IIoT), Operasyonel Teknoloji (OT) cihazlarını ağlara bağlayarak IT (Bilgi Teknolojileri) ve OT ağlarının yakınlaşmasına neden olur. Bu durum, endüstriyel ortamlar için yeni saldırı vektörleri yaratır.

IIoT güvenliğindeki temel prensipler şunlardır:

- **Varlık Envanteri ve Görünürlük:** Korunacak tüm IIoT cihazlarının (PLC'ler, SCADA sistemleri vb.) doğru ve güncel bir envanterinin tutulması, saldırı yüzeyini anlamak için ilk adımdır.
- **Ağ Segmentasyonu ve IDMZ:** IT ve OT ağları arasında bir Endüstriyel DMZ (Demilitarized Zone) oluşturulması, bir ağdaki saldırının diğerine yayılmasını önler.
- **Güvenli Cihaz Yaşam Döngüsü:** Cihazların güvenli bir şekilde devreye alınması, yamanması ve yaşam döngüsünün sonunda güvenli bir şekilde kullanımdan kaldırılması önemlidir.

6.5.5 IoT Cihaz Yaşam Döngüsü Yönetimi ve Güvenlik Güncellemeleri

IoT cihazlarının güvenliği, üretimden kullanımdan kaldırılmaya kadar tüm yaşam döngüsü boyunca sürdürülmelidir. Güvenli OTA güncellemeleri, bu yaşam döngüsü yönetiminin temel bir parçasıdır. Bir OTA güncellemesi sırasında, güncelleme paketinin bütünlüğünün doğrulanması (hash) ve orijinalliğinin doğrulanması (dijital imza) kritiktir. Uzaktan güncelleme sürecinde ortaya çıkabilecek hatalara (bağlantı sorunları, düşük pil, yetersiz depolama) karşı bir `rollback` (önceki güvenli duruma dönme) mekanizmasının bulunması, cihazın çalışmasını garanti eder.

6.6 Mobil Cihaz Donanım Güvenliği

Mobil cihazlar, hem kişisel hem de kurumsal verileri depolayan ve işleyen güçlü bilgi işlem platformlarıdır. Bu nedenle, mobil cihazların donanım güvenliği, verilerin gizliliğini, bütünlüğünü ve kullanılabilirliğini sağlamak için kritik

öneme sahiptir. Mobil cihaz donanım güvenliği, cihazın fiziksel bileşenlerini ve bu bileşenler arasındaki etkileşimleri korumayı amaçlar.

6.6.1 Mobil Cihazlarda Güvenli Eleman (SE) Teknolojisi

Güvenli Eleman (Secure Element - SE), hassas verileri depolamak ve güvenli uygulamaları çalıştırmak için tasarlanmış kurcalamaya dayanıklı bir mikroişlemci çipidir. SE, cihazın işletim sistemindeki tipik kötü amaçlı yazılım saldırılarına karşı koruma sağlar.

Kullanım Alanları:

- **Mobil Ödemeler:** Kredi kartı verilerini güvenli bir şekilde depolar.
- **Kimlik Doğrulama:** VPN erişimi gibi hizmetler için güçlü kimlik doğrulama mekanizmalarında kullanılır.
- **Dijital İmza:** Bir belgenin dijital olarak imzalanması için SE'de saklanan bir anahtar kullanılabilir.

Android cihazlarda, `Open Mobile API` standardı, uygulamaların Güvenli Eleman ile iletişim kurmasını sağlar. Bu API, uygulamaların güvenli donanım işlevlerine kontrollü bir şekilde erişimini mümkün kılar.

6.6.2 Donanım Güvenlik Anahtarları ve FIDO2 Uygulaması

FIDO2 standardı, parolaları kriptografik anahtar çiftleriyle değiştiren, kimlik avı (phishing) direncine sahip bir kimlik doğrulama yöntemidir. Donanım güvenlik anahtarları, bu özel anahtarı kurcalanamaz bir ortamda saklayarak güvenliği en üst düzeye çıkarır.

Bir web uygulamasında FIDO2/WebAuthn entegrasyonu aşağıdaki adımları içerir:

1. **Sunucu Kurulumu:** `fido2-lib` gibi bir kütüphane kullanılarak bir sunucu başlatılır.
2. **Kayıt İşlemi (Registration):**
   - Sunucu, istemciye bir kayıt seçeneği (challenge) gönderir.
   - İstemci tarafında, tarayıcı (`navigator.credentials.create`) kullanıcıdan biyometrik veri veya PIN ile doğrulama yapmasını ister.
   - Cihaz, bir anahtar çifti oluşturur (özel anahtar cihazda kalır) ve genel anahtarı sunucuya gönderir.
3. **Giriş İşlemi (Authentication):**
   - Sunucu, istemciye bir giriş seçeneği (challenge) gönderir.
   - Cihaz, özel anahtarını kullanarak bu seçeneği imzalar.
   - Sunucu, daha önce kaydedilen genel anahtarla imzayı doğrular ve erişim izni verir.

Aşağıdaki Node.js kod örneği, bu süreci basitleştirilmiş bir şekilde gösterir:

```
const express = require('express');
const { Fido2Lib } = require('fido2-lib');
const app = express();
const f2l = new Fido2Lib({ /* seçenekler */ });

// Kayıt Başlangıç Endpoint'i
app.post('/auth/register-begin', async (req, res) => {
    // Kullanıcıya özel kayıt seçeneklerini oluştur ve gönder
    const registrationOptions = await f2l.attestationOptions({
        rp: { name: "FIDO Örneği", id: "example.com" },
        user: { /* kullanıcı bilgileri */ }
    });
    //...
});
```

Bu yaklaşım, parolaların neden olduğu güvenlik risklerini ortadan kaldırır ve kullanıcı deneyimini önemli ölçüde iyileştirir.

6.6.3 Biyometrik Kimlik Doğrulama Güvenliği (Parmak İzi, Yüz Tanıma)

Mobil cihazlarda biyometrik kimlik doğrulama, geleneksel şifrelerden daha yüksek güvenlik sağlar. Biyometrik veriler (parmak izi şablonu, yüz haritası), doğrudan cihazın ana işlemcisi yerine Güvenli Eleman (SE) veya Güvenli Enklav gibi izole donanım ortamlarında işlenir ve depolanır. Bu, verilerin ana işletim sistemi tehlikeye girse bile güvende kalmasını sağlar.

6.6.4 Mobil Cihaz Kurcalama Direnci ve Anti-debugging

Mobil uygulama kurcalama (tampering), uygulamanın ikili kodunun yetkisiz bir şekilde değiştirilmesidir. Anti-tampering ve anti-debugging teknikleri, bir uygulamanın kurcalanma veya hata ayıklama (debugging) girişimi altında çalışıp çalışmadığını tespit etmeyi amaçlar.

Kurcalama yöntemleri arasında ikili yama (binary patching), kod enjeksiyonu ve tersine mühendislik yer alır. Bu saldırılara karşı koymak için kullanılan anti-debugging teknikleri şunlardır:

- **İç Gözetim:** Uygulama, kendi kodunun ve verisinin bütünlüğünü sürekli olarak kontrol eder.
- **Hata Ayıklayıcı Tespiti:** Uygulama, bir hata ayıklayıcı (debugger) tespit ettiğinde kilitlenme veya kritik bir fonksiyonu durdurma gibi savunma mekanizmalarını tetikler.
- **Zamanlama Kontrolü:** Uygulama, normal çalışma ile hata ayıklama altında çalışma arasındaki zaman farkını tespit edebilir.

Aşağıdaki C++ kodu, bir uygulamanın hata ayıklayıcı altında çalışıp çalışmadığını kontrol etmenin basit bir yolunu gösterir:

```
#include <iostream>
#include <Windows.h>

int main() {
    if (IsDebuggerPresent()) {
        std::cout << "Program hata ıı1ayklayc ıaltnda ışıçalyor!" << std::endl;
        exit(-1);
    }
    std::cout << "Program güvenli bir şekilde ışıçalyor." << std::endl;
    return 0;
}
```

Bu kod, ‘IsDebuggerPresent‘ API çağrısını kullanarak basit bir kontrol gerçekleştirir.

6.6.5 Donanım Tabanlı Mobil Cihaz Yönetimi (MDM) Özellikleri

Donanım tabanlı MDM, işletim sistemi ve uygulamaların üzerindeki donanım seviyesinde güvenlik politikalarını uygular. Bu, daha yüksek bir koruma seviyesi sağlar. Kurumsal cihazlar için güvenlik politikalarının uygulanması, uzaktan sıfırlama, veri şifreleme ve bütünlük doğrulaması (attestation) gibi özellikler donanım desteğiyle daha güvenli hale gelir.

Bölüm 7

KİMLİK VE ERİŞİM YÖNETİMİ (IAM) SİSTEMLERİ

Giriş

Kimlik ve erişim yönetimi (IAM), modern güvenlik mimarisinin temel taşlarından biridir. Bu bölümde kimlik yönetimi sistemleri, çok faktörlü kimlik doğrulama, yetkilendirme protokolleri ve erişim kontrol modelleri konularını ele alacağız.

7.1 Kimlik Yönetimi Temelleri ve Mimarisi

Kimlik ve Erişim Yönetimi (IAM), bir kuruluşun dijital kimliklerini ve bu kimliklerin

Faktör Tipi	Açıklama	Örnek Teknolojiler
Bilgi (Knowledge)	Yalnızca kullanıcının bildiği bir bilgi.	Parola, PIN, Güvenlik Sorusu Yanıtı
Sahip Olunan Şey (Possession)	Kullanıcının fiziksel veya dijital olarak sahip olduğu bir nesne.	SMS OTP, Mobil Uygulama (Authenticator), Donanım Token'ı (Fob, USB Anahtarı)
Doğuştan Gelen Özellik (Inherence)	Kullanıcının biyometrik bir özelliği.	Parmak İzi, Yüz Tanıma, Retina Tarama, Ses Tanıma

kaynaklara erişim haklarını yönetmek için kullanılan bir çerçevedir. IAM, doğru kişilerin, doğru zamanda, doğru nedenlerle, doğru kaynaklara erişmesini sağlamayı amaçlar. Bu, bir kuruluşun güvenliğini artırır, uyumluluk gereksinimlerini karşılar ve operasyonel verimliliği artırır.

7.1.1 Dijital Kimlik Yaşam Döngüsü Yönetimi (Digital Identity Lifecycle Management)

Dijital kimlik yaşam döngüsü yönetimi (ILM), bir kullanıcının organizasyondaki görev süresi boyunca dijital kimliğinin ve erişim haklarının yönetilmesini sağlayan kapsamlı bir süreçtir. Bu, bir hesabın ilk oluşturulmasından (onboarding), rol değiştikçe erişim haklarının sürekli olarak ayarlanmasına, düzenli olarak denetlenmesine ve sonunda kullanıcı ayrıldığında erişimin güvenli bir şekilde sonlandırılmasına (offboarding) kadar her şeyi kapsar. Otomasyon, bu sürecin kritik bir parçasıdır ve manuel, hataya açık süreçleri ortadan kaldırır.

- **Aşama 1: Onboarding (İşe Alım):** Süreç, yeni bir çalışan için hesap oluşturulmasıyla başlar. Yöneticiler, ILM panosunu kullanarak yeni kullanıcı hesapları oluşturabilir ve kullanıcının beklenen rolüne ve sorumluluklarına göre erişim haklarını atayabilirler. Bu aşama, kimliğin geçerliliğini sağlamak için ilk kimlik doğrulama ve doğrulama süreçlerini de içerir. Bu aşamanın otomasyonu, çalışanların ilk günden itibaren üretken olmasını sağlar ve BT kaynaklarına olan bağımlılığı azaltır.

  Otomasyon Senaryosu ve Komut Örnekleri: Yeni bir çalışanın İK sistemine kaydedilmesi, otomatik olarak bir iş akışını tetikleyebilir. Bu iş akışı, kullanıcının rolüne ve departmanına göre gerekli erişim haklarını ve grup

üyeliklerini otomatik olarak atar. PowerShell, Active Directory (AD) gibi dizin hizmetlerinde bu tür senaryolar için güçlü bir araçtır.

```
# Yeni bir kullanıcı oluşturma ve temel öznitelikleri belirleme
New-ADUser -Name "John Smith" -Path "OU =Marketing,DC =ornek,DC =com" \
-GivenName "John" -Surname "Smith" -SamAccountName "j.smith" \
-AccountPassword (ConvertTo-SecureString "GuvenliParola123!" -AsPlainText -Force) \
-Enabled $true -ChangePasswordAtLogon $true

# Kullanıcıyı belirli bir gruba ekleme
Add-ADGroupMember -Identity "Marketing_Grubu" -Members "j.smith"
```

- **Aşama 2: Erişim Yönetimi (Ongoing Access):** Bu, bir kullanıcının rolü değiştikçe veya yeni bir projeye atandığında erişim haklarının sürekli olarak ayarlanmasını içeren devam eden bir faaliyettir. Bu dinamik ayarlama, zamanla biriken ve yetkisiz erişim riski oluşturan "yetki kaymasını" (privilege creep) önlemek için hayati öneme sahiptir. Örneğin, bir yazılım mühendisi mühendislik yöneticiliğine terfi ettiğinde, proje yönetimi ve ekip denetimiyle ilgili ek izinler alabilir.
- **Aşama 3: Düzenli Denetimler (Regular Audits):** Düzenli izleme ve denetimler, en az ayrıcalık ilkesi gibi kurumsal güvenlik politikalarına ve GDPR veya HIPAA gibi düzenleyici gereksinimlere uyulmasını sağlamak için gereklidir. Bu denetimler, herhangi bir anormalliğin veya yetkisiz erişim girişiminin zamanında tespit edilmesine yardımcı olur.
- **Aşama 4: Offboarding (İşten Ayrılma):** Yaşam döngüsü, bir kullanıcı organizasyondan ayrıldığında veya artık belirli kaynaklara erişim gerektirmediğinde sona erer. Bu aşamada, kullanıcının hesapları ve altyapı genelindeki tüm ilgili haklar derhal devre dışı bırakılır veya silinir. Bu, yönetim zorluklarına yol açan ve saldırganlar için birer giriş noktası haline gelebilen "yetim" (orphan) veya "bayat" (stale) hesapların oluşmasını önler.

  Otomasyon Senaryosu ve Komut Örnekleri: Otomasyon, offboarding süreçlerinde özellikle önemlidir çünkü güvenlik risklerini ve gecikmeleri azaltır. Bir çalışanın İK sisteminde sonlandırılması, ilgili tüm hesapları ve ayrıcalıkları otomatik olarak geri alan bir iş akışını tetikleyebilir.

```
# PagerDuty API'sinden bir kullanıcının kaldırılmasına dair basitleştirilmiş bir Python örneği
import requests

API_TOKEN = 'your_api_token'
USER_ID = 'user_to_be_removed_id'

headers = {
    'Authorization': f'Token token ={API_TOKEN}',
    'Content-Type': 'application/json'
}

# Kullanıcıyı PagerDuty'den de-provisioning etme
response = requests.delete(f'https://api.pagerduty.com/users/{USER_ID}', headers =headers)

if response.status_code == 204:
    print(f"Kullanıcı ID {USER_ID} başarıyla kaldırıldı.")
else:
    print(f"Hata: Kullanıcı kaldırılamadı. Status kodu: {response.status_code}")
```

Manuel ILM süreçleri ile organizasyondaki güvenlik riskleri arasında doğrudan bir sebep-sonuç ilişkisi vardır. Manuel yönetimdeki gecikmeler ve hatalar, yetim hesaplar ve yetki kayması gibi güvenlik açıklarına yol açar. Bu,

otomasyonun yalnızca verimlilik için değil, aynı zamanda temel bir güvenlik ihtiyacını karşılamak için kritik olduğunu göstermektedir.

7.1.2 Kimlik Yönetişimi ve Yönetim (IGA) Çerçevesi (Identity Governance and Administration Framework)

Kimlik Yönetişimi ve Yönetim (IGA), IAM'in sadece bir yönetim aracı olmaktan çıkıp, kimlik ve erişim süreçlerine bir yönetişim katmanı eklediği bütünsel bir yaklaşımdır. IGA, kimlik yaşam döngüsü yönetimini (ILM) erişim yönetişimi (access governance) ile birleştirerek, doğru kişilerin, doğru kaynaklara, doğru zamanda ve doğru nedenlerle erişimini sağlar.

IGA sistemleri, yöneticilerin kimlik kaosu olarak bilinen durumu azaltmasına ve erişimle ilgili riskleri daha etkili bir şekilde azaltmasına olanak tanıyan bir dizi otomasyon yeteneği sunar. Bu otomasyon, kullanıcıların ilk günden itibaren üretken olmasını sağlar, BT kaynaklarına olan bağımlılığı azaltır ve manuel provizyon hatalarıyla ilişkili güvenlik riskini düşürür.

Temel IGA Bileşenleri:

- **Erişim Talepleri:** Kullanıcıların, BT departmanı için bir "iç uygulama mağazasına" benzer bir self-servis portal aracılığıyla belirli kaynaklara erişim talep etmelerini sağlar. Bu talepler, önceden tanımlanmış politikalara dayanan otomatik iş akışlarıyla işlenir.
- **Rol Yönetimi:** İşlevlere dayalı roller tanımlanır ve erişim hakları bu rollere atanır. Bu, yöneticilerin izin atamasını basitleştirir ve en az ayrıcalık ilkesinin korunmasına yardımcı olur.
- **Erişim Sertifikasyonu:** Kullanıcı erişim hakları, periyodik olarak incelenir ve hala uygun olduklarından emin olmak için onaylanır. Bu, gereksiz izinlerin belirlenmesine ve iptal edilmesine yardımcı olur. Mikro-sertifikasyonlar, bir çalışanın beklenenden farklı bir erişime sahip olduğu bir olayın tetiklenmesiyle anormallikleri hızla tespit etmeyi sağlar.
- **Politika ve Uyum Yönetimi:** IGA sistemleri, iç politikaların ve dış düzenlemelerin (GDPR, HIPAA, SOX) uygulanmasını sağlar. Otomatik denetim izleri ve raporları oluşturarak kuruluşların uyumluluğu göstermesine yardımcı olur.

7.1.3 Hizmet Olarak Kimlik (IDaaS) vs. Şirket İçi Çözümler (On-premises Solutions)

Hizmet Olarak Kimlik (IDaaS), kimlik ve erişim yönetimi (IAM) yeteneklerini bir üçüncü taraf sağlayıcı tarafından internet üzerinden sunulan bulut tabanlı bir abonelik modeli olarak tanımlar. Geleneksel şirket içi çözümler ise, bir organizasyonun kendi altyapısında barındırdığı ve yönettiği sistemlerdir. Bu iki model, kontrol ve kolaylık arasında bir denge sunar.

Detaylı Karşılaştırma:

Özellik	IDaaS (Hizmet Olarak Kimlik)	Şirket İçi (On-premises) Çö
Maliyet	Donanım ve uzman personel ihtiyacını ortadan kaldırarak maliyeti düşürür.	Başlangıçta yüksek yatırım uzun vadede tam mülkiyet ava
Ölçeklenebilirlik	Değişen kullanıcı sayısına hızla adapte olabilir.	Büyümeye uyum sağlamak iç ve olası donanım/yazılım yük tirir.
Yönetim Yükü	Uzman bir sağlayıcı hizmetleri yönettiği için operasyonel yükü azaltır.	Bakım, güncelleme ve sorun yönetim yükü organizasyona a
Özelleştirme	Önceden oluşturulmuş çerçeveler içinde yapılandırma seçenekleri sunar; esneklik sağlayıcıya göre değişir.	Organizasyonun tam paramet zümü uyarlamak için maksin nar.
Güvenlik	Sağlayıcı genellikle altyapı güvenliğinden sorumludur, ancak verilerin nihai sorumluluğu müşteriye aittir (Paylaşılan Sorumluluk Modeli).	Güvenlikten tamamen organi sorumludur.

Bulut teknolojilerinin yaygınlaşması ve hibrit çalışma modellerine geçiş, IDaaS'in yükselişini tetikleyen temel faktörlerdir. Geleneksel şirket içi IAM sistemleri, bulut tabanlı kaynaklar ve SaaS uygulamalarıyla entegrasyonda zorlandığından, IDaaS bu zorluğa doğal bir yanıt olarak ortaya çıkmıştır. IDaaS artık bir "alternatif" değil, hibrit ve bulut öncelikli mimariler için varsayılan bir seçim haline gelmektedir.

7.1.4 Dizin Hizmetleri: Active Directory, LDAP, Bulut Dizinleri

Dizin hizmetleri, kimlik doğrulama ve yetkilendirme süreçlerinin temelini oluşturan, kullanıcılar ve kaynaklar hakkında bilgi depolayan merkezi bir veritabanı sağlar. En yaygın dizin hizmetleri arasında Active Directory (AD), LDAP ve modern bulut dizinleri bulunur.

- **Active Directory (AD) ve LDAP Karşılaştırması:** Siber güvenlik profesyonellerinin anlaması gereken en temel kavramlardan biri, **LDAP'ın bir protokol, Active Directory'nin ise bir ürün olduğudur**.
  - **Lightweight Directory Access Protocol (LDAP):** Dizin hizmetleri ile iletişim kurmak için kullanılan açık, satıcıdan bağımsız bir protokoldür. Tıpkı HTTP gibi, dizin verilerini sorgulamak, değiştirmek ve doğrulamak için standart bir "dil" sağlar. LDAP, özellikle çeşitli BT ekosistemlerinde esneklik ve çapraz platform desteği sunar.
  - **Active Directory (AD):** Microsoft'a ait, Windows tabanlı ağlar için tasarlanmış tescilli bir dizin hizmetidir. Kendi veritabanına ve bir dizi hizmete (DNS, Grup Politikaları vb.) sahiptir. AD, dizinle iletişim kurmak için LDAP protokolünü kullanır, bu nedenle iki kavram sıklıkla karıştırılır. AD, Windows istemcileri ve sunucuları ile derinlemesine entegrasyonu sayesinde zengin bir özellik seti ve merkezi yönetim sunar. Ancak, geleneksel olarak şirket içi ortamlar için tasarlanmıştır ve bulut entegrasyonu zor olabilir.
- **Bulut Dizinleri:** Modern IAM çözümleri, geleneksel AD ve LDAP'nin bulut ve hibrit ortamlardaki sınırlılıklarını aşmayı hedefler. Microsoft Entra ID (eski adıyla Azure AD) gibi bulut dizinleri, bulut tabanlı IAM çözümü olarak öne çıkarak Microsoft ekosisteminden yararlanan işletmeler için idealdir. Bu dizinler, kimliklerin hem şirket içi hem de bulut ortamlarında yönetilmesini destekler.
- **Pratik Komut Örnekleri (LDAP):**

```
# Temel arama: ornek.com alanındaki tüm girişleri döndürür
ldapsearch -x -LLL -H ldap://ornek.com -b "dc =ornek,dc =com" "(objectClass =*)"

# Belirli bir kullanıcıyı arama: john.doe UID'sine sahip kullanıcıyı bulur
ldapsearch -x -H ldap://ornek.com -b "ou =users,dc =ornek,dc =com" "(uid =john.doe)"

# Bir kullanıcıya parola ekleme: Güvenli bir şekilde parola atar
ldappasswd -h localhost -D "cn =admin,dc =ornek,dc =com" -w adminpass -S "uid =john.doe,ou =users
```

7.1.5 Kimlik Federasyonu ve Güven İlişkileri (Identity Federation and Trust Relationships)

Kimlik federasyonu, kullanıcıların tek bir kimlik doğrulama işlemiyle birden fazla bağımsız uygulamaya veya kuruluşa erişimini sağlayan bir sistemdir. Bu modelin temelinde, bir kimlik sağlayıcısı (IdP) ve bir hizmet sağlayıcısı (SP) arasındaki "güven ilişkisi" yatar. Bu güven ilişkisi sayesinde, kullanıcı bir etki alanında kimliği doğrulandıktan sonra, başka bir etki alanındaki kaynaklara ayrı bir hesap veya parola oluşturmak zorunda kalmadan erişebilir.

Güven İlişkisinin Kurulması: Güven, genellikle dijital sertifikalar, karşılıklı SSL/TLS kimlik doğrulaması ve SAML veya OAuth 2.0 gibi açık standart protokoller aracılığıyla kurulur. IdP, kullanıcı kimliğini doğrular ve bu doğrulamanın kanıtını (örneğin, SAML Beyanı) dijital olarak imzalar. SP bu beyana güvenir ve kullanıcıya erişim izni verir.

Kimlik federasyonunun ana hedefleri arasında, yedekli kullanıcı yönetimini ortadan kaldırarak maliyeti düşürmek, güvenliği artırmak ve kullanıcıların paylaşılan veriler üzerinde daha fazla kontrol sahibi olmasını sağlamak yer alır.

7.2 Kimlik Doğrulama Teknolojileri ve Çok Faktörlü Kimlik Doğrulama

Kimlik doğrulama (authentication), bir kullanıcının veya sistemin iddia ettiği kişi veya şey olduğunu doğrulama sürecidir. Bu, genellikle bir parola, PIN veya biyometrik veri gibi bir kimlik bilgisi sunularak yapılır. Çok Faktörlü Kimlik Doğrulama (MFA), bir kullanıcının kimliğini doğrulamak için birden fazla kimlik doğrulama faktörünün kullanılmasını gerektiren bir güvenlik önlemidir. MFA, yalnızca bir parolaya dayanan tek faktörlü kimlik doğrulamadan daha güçlü bir güvenlik seviyesi sağlar.

7.2.1 Parola Politikaları ve Parolasız Kimlik Doğrulama

Geleneksel parolalar, unutulması, çalınması veya tahmin edilmesi kolay olduğu için en zayıf güvenlik katmanlarından biridir. Kimlik avı ve kaba kuvvet saldırıları gibi birçok saldırı vektörü parolaları hedefler. Bu zayıflıkları gidermek için, parolasız kimlik doğrulama yöntemleri geliştirilmiştir.

Parolasız kimlik doğrulama, parolaların yerine daha güvenli ve kullanıcı dostu alternatifler kullanmayı amaçlar. Bu yöntemler, kullanıcının sahip olduğu bir şeye (possession) veya bir biyometrik özelliğe (inherence) dayanır.

Teknik Uygulama Örnekleri:

- **Sihirli Bağlantılar (Magic Links):** Kullanıcının e-posta adresine gönderilen tek kullanımlık bir URL'ye tıklayarak oturum açmasını sağlar. Bu yaklaşım, parola girişini tamamen ortadan kaldırır.

  Python ile Uygulama Örneği:

```
# Supabase Python SDK ile sihirli bağlantı gönderme (basitleştirilmiş)
from supabase import create_client, Client

url: str = "https://your_supabase_url"
key: str = "your_anon_key"
supabase: Client = create_client(url, key)

def send_magic_link(email):
    response = supabase.auth.sign_in_with_otp(
        email =email,
        options ={"emailRedirectTo": "https://myapp.com/welcome"}
    )
    return response

send_magic_link("kullanici@ornek.com")
```

- **Biyometri:** Parmak izi, yüz veya retina tanıma gibi kişinin fiziksel özelliklerini kullanır. FIDO2/WebAuthn gibi standartlar, biyometrik kimlik doğrulamanın modern web tarayıcılarına güvenli bir şekilde entegre edilmesini sağlar.

- **Donanım veya Yazılım Belirteçleri (Tokens):** Kullanıcının sahip olduğu bir cihaz (USB anahtarı) veya mobil uygulama tarafından üretilen tek kullanımlık kodları kullanır.

7.2.2 Çok Faktörlü Kimlik Doğrulama (MFA) Yöntemleri ve Teknolojileri

MFA, bir kullanıcının kimliğini doğrulamak için en az iki farklı türde kimlik doğrulama faktörü gerektiren bir güvenlik mekanizmasıdır. Bu, bir parolanın çalınması durumunda bile yetkisiz erişimi önleyerek ek bir koruma katmanı sağlar. Gerçek MFA, aynı türden iki faktör (örneğin, parola ve güvenlik sorusu) kullanmaktan daha güvenlidir, çünkü saldırganın farklı kanalları ve yöntemleri kullanmasını gerektirir.

Kimlik Doğrulama Faktörleri:

Faktör Tipi	Açıklama	Örnek Tek
Bilgi (Knowledge)	Yalnızca kullanıcının bildiği bir bilgi.	Parola, PIN
Sahip Olunan Şey (Possession)	Kullanıcının fiziksel veya dijital olarak sahip olduğu bir nesne.	SMS OTP, henticator), USB Anaht
Doğuştan Gelen Özellik (Inherence)	Kullanıcının biyometrik bir özelliği.	Parmak İzi, ma, Ses Tan

7.2.3 Risk Bazlı ve Adaptif Kimlik Doğrulama

Adaptif kimlik doğrulama (Risk Bazlı Kimlik Doğrulama olarak da bilinir), kullanıcının davranışlarını ve bağlamsal faktörleri (konum, cihaz, IP adresi) gerçek zamanlı olarak analiz ederek kimlik doğrulama adımlarını dinamik olarak ayarlar. Geleneksel kimlik doğrulamanın aksine, bu yaklaşım tek bir oturum açma girişimini bir dizi statik kurala göre değerlendirmek yerine, gerçek zamanlı risk sinyallerine yanıt verir.

İşleyişi:

- **Temel Profil Oluşturma:** Sistem, makine öğrenimi algoritmalarını kullanarak her kullanıcı için tipik davranış kalıplarını belirler. Bu, bir kullanıcının hangi cihazdan, hangi IP adresinden ve günün hangi saatinde oturum açtığını içerir.
- **Anomali Tespiti:** Sistem, geçerli bir oturum açma girişimini bu temel profille karşılaştırır. Tipik davranıştan herhangi bir sapma (örneğin, olağandışı bir konumdan oturum açma, yeni bir cihaz kullanma) bir anomali olarak işaretlenir.
- **Dinamik Yanıt:** Tespit edilen risk seviyesine göre, sistem kimlik doğrulama deneyimini dinamik olarak ayarlar. Düşük riskli bir durumda (örneğin, rutin bir oturum açma), ek bir doğrulama adımı istenmeyebilir. Ancak, yüksek riskli bir durumda, sistem ek bir MFA adımı talep edebilir veya erişimi tamamen engelleyebilir.

Bu yaklaşım, güvenlik gereksinimleri ile kullanıcı deneyimi arasında bir denge kurar. Adaptif kimlik doğrulama, gereksiz MFA istemlerini en aza indirerek ”MFA yorgunluğunu” azaltır ve kullanıcı verimliliğini artırırken, güvenliği ihlal etmez.

7.2.4 Biyometrik Kimlik Doğrulama Sistemleri ve Doğruluk Değerlendirmeleri

Biyometrik kimlik doğrulama, bir kişinin benzersiz fiziksel veya davranışsal özelliklerini (parmak izi, yüz veya retina tanıma) kullanarak kimliğini doğrulayan bir güvenlik sürecidir. Parolalardan farklı olarak, bunlar çalınması veya unutulması imkansız olan ”doğuştan gelen” faktörlerdir.

Teknolojiler ve Doğruluk Değerlendirmesi:

- **Parmak İzi:** Ucuz, yaygın ve kullanışlıdır. Bununla birlikte, bazı tüketici sınıfı sensörler sahte parmak izleri ile aşılabilir.
- **Yüz Tanıma:** Yüz tanıma teknolojisi, mobil cihazlarda ve diğer sistemlerde yaygın olarak kullanılır. Modern çözümler, fotoğraf veya maske ile aldatmayı önlemek için ”canlılık” testi (baş hareketi, göz kırpma) kullanır.
- **Retina ve İris Tanıma:** Oldukça doğru ve güvenlidir, ancak maliyetli ve özel donanım gerektirir.
- **Damar Tanıma:** Cilt altındaki kan damarı modellerini haritalandırır. İris tanımadan bile daha doğru olabilir ve taklit edilmesi oldukça zordur.

Bir biyometrik sistemin doğruluğu, yanlış kabul oranı (FAR) ve yanlış reddetme oranı (FRR) gibi metriklerle değerlendirilir. Çok modlu biyometrik kimlik doğrulama (birden fazla biyometri kontrolü), taklit edilmesini daha zor hale getirerek bu sistemlerin güvenliğini artırır.

7.2.5 Sertifika Tabanlı Kimlik Doğrulama ve Akıllı Kart Entegrasyonu

Sertifika tabanlı kimlik doğrulama, kullanıcının kimliğini kriptografik bir anahtar çiftine dayanan bir X.509 dijital sertifikası aracılığıyla doğrular. Bu yöntem, parolalara kıyasla çok daha güçlü bir güvenlik sunar.

İşleyiş: Kullanıcının özel anahtarı, güvenli bir donanım aygıtında (akıllı kart veya USB anahtarı) saklanır. Kimlik doğrulama sırasında, sistem bir meydan okuma gönderir ve kullanıcı özel anahtarını kullanarak bu meydan okumayı imzalar. Sunucu, sertifikanın ait olduğu genel anahtarı kullanarak imzayı doğrular ve kullanıcıya erişim izni verir.

Akıllı kartlar, özel anahtarları izole ve güvenli bir şekilde sakladıkları için sertifika tabanlı kimlik doğrulama için ideal bir çözümdür. Kart çalınsa bile, PIN kodu veya biyometrik veri gibi ek bir faktör olmadan özel anahtara erişilemez. Bu, sertifika tabanlı kimlik doğrulamanın temel güvenliğini sağlar.

7.3 Yetkilendirme Modelleri ve Erişim Kontrol Çerçeveleri

Yetkilendirme (authorization), bir kullanıcının veya sistemin kimliği doğrulandıktan sonra, hangi kaynaklara erişebileceğini ve bu kaynaklar üzerinde hangi işlemleri yapabileceğini belirleme sürecidir. Erişim kontrol modelleri, bu yetkilendirme kararlarını uygulamak için kullanılan kurallar ve politikalar bütünüdür. Bu modeller, bir kuruluşun güvenlik gereksinimlerine ve iş süreçlerine göre seçilir ve uygulanır.

7.3.1 Rol Tabanlı Erişim Kontrolü (RBAC) Tasarımı ve Uygulaması

RBAC, erişim haklarını bireysel kullanıcılara değil, onların organizasyondaki rollerine (işlev) göre atayan bir yetkilendirme modelidir. Bu, yöneticiler için erişim yönetimini basitleştirir, çünkü bir kullanıcı bir role atandığında, o rolün önceden tanımlanmış tüm izinlerini otomatik olarak devralır.

Adım Adım Tasarım ve Uygulama Kılavuzu:

1. **Strateji Geliştirme:** Mevcut erişim kontrol mekanizmaları değerlendirilir ve RBAC ile ulaşılmak istenen sonuçlar (örneğin, kullanıcı provizyonunu otomatikleştirmek) belirlenir.

2. **Sistem Envanteri Çıkarma:** Erişim kontrolü gerektiren her kaynak (e-posta, bulut uygulamaları, veritabanları, vb.) listelenir.

3. **İş Gücü Analizi:** BT, İK ve yöneticilerle iş birliği içinde, çalışanlar ortak erişim ihtiyaçlarına göre rollere göre gruplandırılır. Bu, ”rol patlaması” (role explosion) olarak bilinen aşırı sayıda rol oluşturma tuzağından kaçınmak için kritik öneme sahiptir.

4. **Rollerin Tanımlanması:** Envanter ve iş gücü analizi sonuçları, en az ayrıcalık (least privilege) ilkesi temelinde eşleştirilir.

   Pratik Örnekler:

   - `Temel Kullanıcı` rolü: Tüm kullanıcılar için geçerli olan e-posta ve sohbet uygulamalarına erişim sağlar.
   - `Pazarlama Yöneticisi` rolü: Pazarlama ekibi için gerekli olan kampanya yönetimi araçlarına erişim sağlar.
   - `Finans Muhasebecisi` rolü: Yalnızca finansal raporlara okuma/yazma erişimi vardır.

7.3.2 Öznitelik Tabanlı Erişim Kontrolü (ABAC) Gelişmiş Senaryolar

ABAC, RBAC'tan daha esnek ve daha ayrıntılı yetkilendirme sağlayan bir yetkilendirme modelidir. Erişim kararlarını, kullanıcının, kaynağın, eylemin ve çevrenin özniteliklerini (attributes) değerlendirerek verir. Bu dinamik ve bağlamsal yaklaşım, RBAC'ın karşılaştığı ”rol patlaması” sorununu çözer, çünkü birden fazla rol tanımlamak yerine mevcut özniteliklerin kombinasyonları kullanılır.

Temel Bileşenler ve Politika Örnekleri:

- **Öznitelikler:** Kullanıcı (`departman =Finans`), Kaynak (`gizlilik =Gizli`), Eylem (`okuma`), Ortam (`saat =16:00`) gibi karakteristiklerdir.

- **Politika:** Erişim kuralını belirleyen, mantıksal bir ‘if-then‘ ifadesidir.

Gelişmiş Senaryo Örnekleri:

- **Finans Sektörü:** Bir politika, bir kullanıcının yalnızca `departman =Finans` ise VE `istihdam_durumu =Tam-Zamanlı` ise VE `ağ_güvenlik_seviyesi =şirket_içi_güvenli` ise gizli bir finansal raporu indirmesine izin verebilir.
- **Sağlık Sektörü:** Bir doktorun bir hastanın acil durumdaki tıbbi kayıtlarına, yalnızca rolü (`doktor`) ve aciliyet durumu (`acil_servis`) gibi özniteliklere dayalı olarak erişim izni verilebilir.
- **Perakende:** Bir mağaza yöneticisinin envanter seviyelerini ayarlamasına, sadece `konum =mağaza_içi` ve `saat =çalışma_` gibi özniteliklere dayalı olarak izin verilebilir.

ABAC'ı uygulamak, karmaşık politika yönetimi ve öznitelik yönetimi zorluklarını beraberinde getirebilir. Ancak, özellikle bulut ve hibrit ortamlarda, bağlamsal ve dinamik yetkilendirme gerektiren durumlarda ABAC tercih edilen bir model haline gelmektedir. Open Policy Agent (OPA) gibi araçlar, bu tür politikaların kod olarak yazılmasını ve merkezi olarak uygulanmasını sağlayarak bu karmaşıklığı yönetmeye yardımcı olur.

7.3.3 Politika Tabanlı Erişim Kontrolü ve Dinamik Yetkilendirme

Politika tabanlı erişim kontrolü, yetkilendirme kararlarını merkezi olarak yönetilen ve dinamik olarak değerlendirilen politikalar aracılığıyla verir. Bu, XACML (Extensible Access Control Markup Language) gibi standartlaştırılmış dillerle veya OPA gibi araçlarla uygulanabilir.

Bir kullanıcının bir kaynağa erişim talebi, bir Politika Uygulama Noktası (PEP) tarafından yakalanır. Bu istek, bir Politika Karar Noktası'na (PDP) iletilir. PDP, istekle ilgili öznitelikleri toplar ve politika deposundaki kurallarla eşleştirerek bir karar (izin ver/reddet) verir. Bu karar daha sonra PEP tarafından uygulanır.

7.3.4 İsteğe Bağlı (DAC) vs. Zorunlu (MAC) Erişim Kontrol Modelleri

Bu yetkilendirme modelleri, en kısıtlayıcıdan en esneğe doğru bir spektrumun iki ucunu temsil eder.

- **Zorunlu Erişim Kontrolü (MAC):** En katı modeldir. Erişim kararları, merkezi bir yetkili (sistem yöneticisi) tarafından, hassasiyet seviyelerine dayalı olarak verilir. Kullanıcıların kendi erişim haklarını değiştirmesi veya başkalarına hak vermesi mümkün değildir. Genellikle askeri veya devlet kurumları gibi en yüksek güvenlik gereksinimleri olan ortamlarda kullanılır.
- **İsteğe Bağlı Erişim Kontrolü (DAC):** En az kısıtlayıcı modeldir. Kaynakların sahibi, bu kaynaklara kimlerin erişebileceğine kendi takdiriyle karar verir. Bu, esneklik sağlasa da, yetkiyi kötüye kullanma veya yanlışlıkla yüksek ayrıcalıklar verme riskini taşır.

Özellik	Zorunlu Erişim Kontrolü (MAC)	İsteğe Bağlı Erişim Kontrolü (DAC)	Rol Tabanlı Erişim Kontrolü (RBAC)
Yönetim	Merkezi ve katı. Sadece yöneticiler.	Merkezi değil. Kaynak sahipleri.	Merkezi ve yapılandırılmış.
Esneklik	Çok düşük.	Yüksek.	Orta.
Kullanım Alanı	Askeri, istihbarat, çok gizli veriler.	Kişisel dosyalar, küçük iş grupları.	Büyük kuruluşlar, kurumsal uygulamalar.

7.3.5 Sıfır Güven (Zero Trust) Erişimi ve Sürekli Yetkilendirme

Sıfır Güven, "Asla güvenme, daima doğrula" prensibi üzerine kurulu bir güvenlik çerçevesidir. Geleneksel güvenlik, ağın içindeki kullanıcılara otomatik olarak güvenirken, Sıfır Güven, ağ içi veya dışı tüm erişim denemelerini şüpheli kabul eder ve her seferinde kimliği, cihazı ve bağlamı doğrular.

Sürekli Yetkilendirme: Bu yaklaşım, yetkilendirmenin tek bir andan ibaret olmadığını kabul eder. Kullanıcı bir kaynağa erişim sağladıktan sonra bile, yetkilendirme sürekli olarak izlenir ve yeniden değerlendirilir. Bu, riskli bir davranış veya bağlamsal bir değişiklik (örneğin, kullanıcının konumunun aniden değişmesi) durumunda erişimi dinamik olarak sınırlamayı veya tamamen sonlandırmayı mümkün kılar. Sıfır Güven, uzaktan çalışmanın yaygınlaşması ve geleneksel güvenlik çevresinin ortadan kalkmasıyla modern IAM stratejisinin temel bir bileşeni haline gelmiştir.

7.4 Ayrıcalıklı Erişim Yönetimi (PAM) Çözümleri

Ayrıcalıklı Erişim Yönetimi (PAM), bir kuruluşun en kritik sistemlerine ve verilerine erişimi olan ayrıcalıklı hesapları (privileged accounts) yönetmek ve güvence altına almak için kullanılan bir siber güvenlik stratejisidir. Ayrıcalıklı hesaplar, genellikle sistem yöneticileri, veritabanı yöneticileri ve ağ mühendisleri gibi BT personeli tarafından kullanılır. Bu hesaplar, bir kuruluşun altyapısı üzerinde tam kontrole sahip oldukları için, siber saldırganlar için birincil hedeftir.

7.4.1 Ayrıcalıklı Hesap Keşfi ve Envanter Yönetimi

Bir PAM yaşam döngüsünün ilk ve en kritik adımı, organizasyon içindeki tüm ayrıcalıklı hesapların keşfedilmesi ve envanterinin çıkarılmasıdır. Bu aşama, yönetilmeyen veya unutulmuş hesapları (”gölge” veya ”yetim” hesaplar) belirleyerek büyük güvenlik kör noktalarını giderir. Otomatik keşif araçları, hem bulut hem de şirket içi ortamları tarayarak alan yöneticisi hesapları, hizmet hesapları, yerel yönetici hesapları ve hatta kod içine gömülü sırları bulur ve merkezi bir envanterde toplar.

7.4.2 Parola Kasaları, Döndürme ve Oturum Yönetimi

PAM çözümleri, ayrıcalıklı hesapların parolalarını ”parola kasalarında” (password vaults) saklayarak en az ayrıcalık ilkesini güçlendirir ve parolaların doğrudan kullanıcılar tarafından bilinmesini engeller.

Parola Kasaları (Password Vaulting): Hassas parolalar, şifreli bir kasada güvenli bir şekilde saklanır. **Parola Döndürme (Rotation):** Kasadaki parolalar, manuel müdahaleye gerek kalmadan otomatik olarak belirli aralıklarla veya her kullanımdan sonra değiştirilir. **Oturum Yönetimi (Session Management):** Ayrıcalıklı bir kullanıcı, bir aracı (proxy) sunucu aracılığıyla hedeflenen sisteme bağlanır. Parola aracı tarafından otomatik olarak enjekte edilir ve kullanıcıya asla gösterilmez. Bu aracı, aynı zamanda oturum izleme ve kaydını da sağlar. Kayıtlar, denetim, adli tıp analizi ve kök neden analizi için video formatında saklanabilir ve tuş vuruşlarını ve komutları içerebilir.

7.4.3 Tam Zamanında (JIT) Erişim ve Geçici Ayrıcalık Yükseltme

Tam Zamanında (JIT) erişim, kullanıcılara ayrıcalıklı hesaplara veya kaynaklara yalnızca belirli bir görev için ve sınırlı bir süre boyunca erişim veren bir yöntemdir. Bu, ”kalıcı ayrıcalıkların” (standing privileges) neden olduğu riskleri ortadan kaldırır. Geleneksel ”her zaman açık” ayrıcalıklar, saldırganların ağda yanal hareket etmesi için birincil vektörlerdir. JIT erişim, bu saldırı yüzeyini önemli ölçüde küçülterek bu riski doğrudan azaltır ve Sıfır Güven (Zero Trust) modelinin en temel uygulamalarından biridir.

İşleyiş:

1. Bir kullanıcı, belirli bir görev için ayrıcalıklı erişim talebinde bulunur.
2. Bu talep, önceden tanımlanmış politikalara veya bir yöneticinin manuel onayına göre doğrulanır.
3. Onay alındıktan sonra, kullanıcının ayrıcalıkları geçici olarak yükseltilir.
4. Belirlenen süre sonunda veya görevin tamamlanmasının ardından ayrıcalıklar otomatik olarak geri alınır.

7.4.4 Ayrıcalıklı Oturum İzleme ve Kayıt

Ayrıcalıklı oturum izleme ve kayıt, ayrıcalıklı hesapların faaliyetlerini gerçek zamanlı olarak izleme ve video veya metin olarak kaydetme yeteneğidir. Bu, kuruluşların denetim, adli tıp ve uyumluluk gereksinimlerini karşılamalarına yardımcı olur.

Teknik Detaylar:

- **Gerçek Zamanlı İzleme:** Yöneticiler, canlı oturumları izleyebilir ve şüpheli eylemler durumunda oturumu anında sonlandırabilir.
- **Kayıt:** Oturumlar, daha sonraki analizler için video formatında kaydedilir. Kayıtlar, tuş vuruşlarını, komut satırı çıktılarını ve diğer kullanıcı etkinliklerini içerir.
- **Denetim İzi (Audit Trail):** Kaydedilen her oturum, yasal uyumluluk gereksinimlerini (HIPAA, PCI DSS) karşılamak için değiştirilemez bir denetim izi oluşturur. Bu, "kimin neyi, ne zaman ve nerede yaptığını" belirlemeyi sağlar ve siber sigorta poliçeleri için de giderek daha fazla talep edilmektedir.

7.4.5 Bulut ve DevOps Ortamlarıyla PAM Entegrasyonu

DevOps ekiplerinin bulut ve CI/CD ortamlarında kullandığı ayrıcalıklı erişimi güvenli bir şekilde yönetmek için PAM çözümlerinin entegrasyonu esastır.

Pratik Senaryolar:

- **Merkezi Yönetim:** Bir PAM çözümü, geliştiricilerin ve BT operasyon ekibinin çeşitli bulut sistemlerine (AWS, Azure) erişimini merkezi olarak yönetebilir. Bu, politikaların güvenlik sistemlerini atlamasını önler.
- **API'ler ve Sırlar:** Otomasyon betikleri, hassas verilere (veritabanı şifreleri, API anahtarları) doğrudan erişmek yerine, PAM çözümünün parola kasasından bu sırları talep edebilir. Bu, kodun içine gömülü sırları ortadan kaldırır.
- **Aracısız Mimari (Agentless Architecture):** Bazı PAM çözümleri, her sunucuya veya cihaza bir aracı (agent) kurma gereksinimini ortadan kaldırarak bulut ve DevOps ortamlarında entegrasyonu basitleştirir ve uygulama gecikmelerini önler.

7.5 Tekli Oturum Açma (SSO) ve Kimlik Federasyonu

Tekli Oturum Açma (SSO), bir kullanıcının birden fazla uygulamaya ve hizmete tek bir kimlik bilgisi setiyle erişmesini sağlayan bir kimlik doğrulama yöntemidir. Bu, kullanıcıların her bir uygulama için ayrı bir parola hatırlama zorunluluğunu ortadan kaldırır ve kullanıcı deneyimini iyileştirir. Kimlik federasyonu, farklı kuruluşlar veya alanlar arasında kimlik bilgilerinin güvenli bir şekilde paylaşılmasını sağlayan bir sistemdir. Bu, kullanıcıların bir kuruluştaki kimlik bilgilerini kullanarak başka bir kuruluştaki hizmetlere erişmesine olanak tanır.

7.5.1 SAML 2.0 Federasyonu Uygulaması

SAML (Security Assertion Markup Language), bir kimlik sağlayıcısı (IdP) ve bir hizmet sağlayıcısı (SP) arasında kimlik doğrulama ve yetkilendirme bilgilerini güvenli bir şekilde iletmek için kullanılan açık bir XML tabanlı protokoldür. SAML 2.0, tekli oturum açma (SSO) için endüstri standardı olarak yaygın olarak kullanılır.

Adım Adım SAML Akışı:

1. **Güvenin Kurulması:** IdP ve SP, SAML meta verilerini (varlık kimlikleri, uç noktalar, sertifikalar) değiş tokuş ederek aralarında bir güven ilişkisi kurar.
2. **İstek Gönderme:** Kullanıcı bir web tarayıcısı aracılığıyla SP'deki bir kaynağa erişmek ister. SP, kullanıcıya bir SAML isteği (AuthnRequest) oluşturur ve tarayıcıyı IdP'ye yönlendirir.
3. **Kimlik Doğrulama:** Tarayıcı, SAML isteğini IdP'ye iletir. IdP, kullanıcıdan kimliğini doğrulamak için oturum açmasını ister.
4. **Beyan Oluşturma:** Kimlik doğrulandıktan sonra, IdP bir SAML Beyanı (Assertion) oluşturur. Bu, kullanıcının kimlik ve yetkilendirme bilgilerini içeren, dijital olarak imzalanmış bir XML belgesidir.

5. **Beyanı Gönderme ve Erişim Verme:** IdP, SAML Beyanını tarayıcıya geri gönderir. Tarayıcı, SAML Beyanını SP'ye iletir. SP, beyanın geçerliliğini doğrular. Eğer doğrulama başarılı olursa, kullanıcıya istenen kaynağa erişim izni verir ve oturum başlatılır.

7.5.2 OAuth 2.0 ve OpenID Connect Protokolleri

Bu iki protokol birbirini tamamlar ancak farklı amaçlara hizmet eder.

- **OAuth 2.0:** Bir yetkilendirme protokolüdür. Kullanıcının bir uygulamaya, parolalarını paylaşmadan başka bir servisteki (örneğin, Google) kaynaklarına erişim izni vermesini sağlar. Amacı "erişim yetkisi" vermektir, "kimlik doğrulamak" değil.
- **OpenID Connect (OIDC):** OAuth 2.0'ın üzerine inşa edilmiş bir kimlik doğrulama katmanıdır. Amacı, bir kullanıcının kimliğini güvenli bir şekilde doğrulamaktır. Kullanıcı kimliği hakkında bilgi içeren bir ID belirteci (ID Token) sağlar. Bu, SSO'nun temelini oluşturur.

7.5.3 Web Erişim Yönetimi (WAM) Çözümleri

Web Erişim Yönetimi (WAM), web kaynaklarına erişimi kontrol eden bir kimlik yönetimi biçimidir. Kimlik doğrulama, politika tabanlı yetkilendirme ve SSO yeteneklerini birleştirir.

WAM Mimarileri:

- **Eklenti Tabanlı (Agent-based):** Her web/uygulama sunucusuna bir eklenti (web agent) yüklenir. Bu eklentiler, her istekte harici bir politika sunucusuyla iletişim kurarak erişim kararı alır.
- **Vekil Sunucu Tabanlı (Proxy-based):** Tüm web istekleri, arka uç sunucularına iletilmeden önce bir vekil sunucu üzerinden yönlendirilir. Bu, sunucu başına eklenti ihtiyacını ortadan kaldırır ancak ek donanım ve ağ darboğazı riski gerektirebilir.
- **Belirteç Tabanlı (Tokenization):** Kullanıcı, kimlik doğrulandıktan sonra bir belirteç (token) alır ve bu belirteci doğrudan arka uç sunucularına erişim için kullanır.

7.5.4 Alanlar Arası Kimlik Yönetimi (SCIM) Protokolü

SCIM, alanlar arasında kullanıcı verilerini güvenli bir şekilde yönetmek ve iletmek için kullanılan bir uygulama düzeyi protokoldir. Amacı, kullanıcı yaşam döngüsü süreçlerini (oluşturma, güncelleme, silme) otomatikleştirmektir.

Nasıl Çalışır? SCIM, REST API'leri ve JSON formatını kullanarak CRUD (Create, Read, Update, Delete) operasyonlarını gerçekleştirir. Bir kimlik sağlayıcıdaki (IdP) bir kullanıcı profili değiştirildiğinde, SCIM bu değişikliği otomatik olarak hedef uygulamalara senkronize eder. Bu, manuel girişi azaltarak insan hatasını büyük ölçüde düşürür.

Uygulama Örnekleri:

- **Okta ile SCIM:** Okta'da bir kullanıcı pasifize edildiğinde, SCIM bu değişikliği otomatik olarak bir Snowflake veritabanına iletir ve kullanıcının erişimi hemen sonlandırılır.
- **Azure AD ile SCIM:** Azure AD'de bir kullanıcı oluşturulduğunda veya bir grup atandığında, SCIM protokolü bu kullanıcının TeamRetro gibi bir SaaS uygulamasında da otomatik olarak oluşturulmasını sağlar.

7.5.5 Sosyal Kimlik Entegrasyonu ve Harici Kimlik Sağlayıcıları

Sosyal kimlik entegrasyonu, bir uygulamanın, kullanıcıların zaten sahip olduğu sosyal medya hesaplarını (örneğin Google, Facebook, Twitter) kimlik doğrulama amacıyla kullanmasına izin verme sürecidir. Bu, kullanıcı deneyimini iyileştirirken, geliştiricilerin kimlik doğrulama altyapısı kurma yükünü azaltır. OAuth ve OIDC protokolleri bu entegrasyon için temeldir.

7.6 Kimlik Analitiği ve Kullanıcı Davranışı İzleme

Kimlik analitiği, bir kuruluşun kimlik ve erişim yönetimi (IAM) sistemlerinden toplanan verileri analiz ederek, güvenlik risklerini belirlemek ve kullanıcı davranışlarındaki anormallikleri tespit etmek için kullanılan bir süreçtir. Kullanıcı Davranışı Analitiği (UBA), normal kullanıcı davranışının bir temel çizgisini oluşturur ve bu temel çizgiden sapan şüpheli aktiviteleri belirler. Bu, içeriden gelen tehditleri, ele geçirilmiş hesapları ve diğer gelişmiş saldırıları tespit etmek için etkili bir yöntemdir.

7.6.1 Kullanıcı ve Varlık Davranış Analizi (UEBA) Uygulaması

Kullanıcı ve Varlık Davranış Analizi (UEBA), makine öğrenimi ve davranış analitiğini kullanarak bir organizasyonun ağındaki kullanıcıların ve varlıkların davranışlarındaki anormallikleri tespit eden gelişmiş bir siber güvenlik yaklaşımıdır. Geleneksel kural tabanlı sistemlerden farklı olarak, UEBA ”normal” davranışın ne olduğunu öğrenir ve bu temelden sapmaları işaretler.

Adım Adım Uygulama Süreci:

1. **Veri Toplama:** Ağ trafiği, sistem günlükleri, uygulama kullanım metrikleri, oturum açma faaliyetleri ve veri erişim desenleri gibi çeşitli kaynaklardan veri toplar.
2. **Modelleme ve Temel Oluşturma:** Toplanan verilerle her kullanıcı ve varlık için bir ”normal davranış profili” oluşturulur.
3. **Anomali Tespiti:** Sistem, gerçek zamanlı olarak bu temelden sapmaları sürekli olarak izler. Örneğin, bir kullanıcının normalde belirli bir sunucudan küçük dosyalar indirdiği öğrenilirse, aniden büyük miktarda veri indirmesi bir anomali olarak işaretlenir.
4. **Uyarı ve Yanıt:** Anomali, bir risk puanıyla birlikte güvenlik ekibine bildirilir. Ekip, potansiyel bir ihlali araştırır ve gerekli önlemleri alır. Bazı sistemler, bir saldırıyı durdurmak için anında müdahale edebilir.

7.6.2 Kimlik Risk Puanlaması ve Anomali Tespiti

UEBA, tespit ettiği her anomaliye, sapmanın ciddiyetine ve kullanıcının veya varlığın hassasiyetine bağlı olarak bir risk puanı atar. Bu puan, analistlerin en kritik olaylara öncelik vermesine yardımcı olur. Bir risk puanı, genellikle sıfır ile 100 arasında bir değer olarak belirlenir ve sapma ne kadar büyükse puan o kadar artar.

Risk Faktörleri:

- **Bağlamsal Faktörler:** Kullanılan cihaz, IP adresi, coğrafi konum, oturum açma zamanı.
- **Davranışsal Faktörler:** Hatalı oturum açma denemesi sayısı, olağandışı veri aktarımı, erişim kalıplarından sapma.
- **Varlık Faktörleri:** Kullanıcının rolü, ayrıcalık seviyesi veya hesabın doğası.

Kimlik risk puanlaması, adaptif kimlik doğrulama ile doğrudan entegre edilebilir. Yüksek bir risk puanı, kimlik doğrulama sırasında ek bir MFA adımı veya tam bir erişim engellemesi gibi güvenlik önlemlerini otomatik olarak tetikleyebilir. Bu, sistemlerin güvenlik kararlarını gerçek zamanlı ve dinamik bir şekilde almasını sağlar.

7.6.3 Erişim Sertifikasyonu ve Yeniden Sertifikasyon Süreçleri

Erişim sertifikasyonu, bir kullanıcının belirli kaynaklara sahip olduğu erişim haklarının hala geçerli ve işlevleri için gerekli olup olmadığının periyodik olarak incelenmesi ve onaylanmasıdır. Bu süreç, çalışanlar rollerini değiştirdikçe veya yeni projeler üstlendikçe biriken ”yetki kayması” (privilege creep) sorununu çözer.

Uygulama Adımları:

1. **Hakların Tespiti:** Tüm kullanıcı hesapları, roller ve izinler dahil olmak üzere kimin neye erişimi olduğu kapsamlı bir şekilde belirlenir.

2. **Gözden Geçirenlerin Atanması:** Erişim incelemeleri, genellikle kullanıcının yöneticisi veya uygulama sahibi gibi iş bağlamını en iyi bilen kişiler tarafından yapılır.
3. **İnceleme ve Onaylama:** Gözden geçirenler, her erişim hakkının hala gerekli olup olmadığına karar verir.
4. **İyileştirme ve Geri Alma:** Onaylanmayan tüm gereksiz haklar otomatik olarak geri alınır.
5. **Günlükleri Tutma ve Raporlama:** Her kararın ayrıntılı günlükleri ve denetim raporları, uyumluluk gereksinimlerini (SOX, HIPAA, GDPR) karşılamak için tutulur.

7.6.4 Görevler Ayrılığı (SoD) İzleme ve İhlal Tespiti

Görevler Ayrılığı (SoD), tek bir kişinin bir organizasyonda dolandırıcılığa veya hataya yol açabilecek tüm adımları tek başına gerçekleştirmesini önlemek için iş süreçlerinin birden fazla kişiye dağıtılması prensibidir. IAM/IGA çözümleri, iki veya daha fazla çelişkili ayrıcalığa sahip kullanıcıları (örneğin, hem bütçe oluşturma hem de onaylama hakları) otomatik olarak izler. Bir ihlal tespit edildiğinde, sistem bir uyarı verir ve iç kontrolü güçlendirir.

7.6.5 Kimlik Yönetişimi Raporlama ve Uyum Panoları

Raporlama ve panolar, yöneticilere ve denetçilere kimlik ve erişim ortamının durumu hakkında gerçek zamanlı görünürlük sağlar. Bu araçlar, güvenlik ve uyum duruşunu değerlendirmek için kritik öneme sahiptir.

Örnek Metrikler:

- Yetim ve gölge hesap sayısı
- Onaylanan ve reddedilen erişim taleplerinin yüzdesi
- Denetlenen ayrıcalıklı oturumların yüzdesi
- Tespit edilen politika ihlallerinin sayısı
- Rol değişikliği veya ayrılıktan sonra erişim haklarını geri alma için geçen ortalama süre.

Bölüm 8

SİBER TEHDİT İSTİHBARATI VE TEHDİT AVCILIĞI

Giriş

Siber tehdit istihbaratı ve tehdit avcılığı, modern siber güvenlik operasyonlarının proaktif boyutunu oluşturan kritik disiplinlerdir. Bu bölümde tehdit istihbaratı toplama, analiz etme ve uygulama süreçlerini, ayrıca tehdit avcılığı metodolojilerini detaylı olarak inceleyeceğiz.

8.1 Cyber Threat Intelligence (CTI) Fundamentals

Siber Tehdit İstihbaratı (CTI), bir kuruluşun siber tehditlere karşı savunma yeteneklerini güçlendirmek için toplanan, işlenen ve analiz edilen bilgidir. CTI, bir kuruluşun potansiyel saldırganları, onların motivasyonlarını, yeteneklerini ve kullandıkları altyapıyı anlamasına yardımcı olur. Bu bilgiler, güvenlik ekiplerinin proaktif bir savunma stratejisi oluşturmasına, olay müdahale süreçlerini iyileştirmesine ve kaynaklarını daha etkili bir şekilde tahsis etmesine olanak tanır.

8.1.1 Threat Intelligence Lifecycle ve Collection Methods

Tehdit istihbaratı yaşam döngüsü, ham veriyi değerli istihbarata dönüştüren ve sürekli bir geri bildirim döngüsü ile kendini yenileyen, yapılandırılmış bir süreçtir. Bu döngü, CTI programının temelini oluşturur ve reaktif bir modelden proaktif bir savunma yaklaşımına geçişin temelini atar.

Yaşam Döngüsü Aşamaları:

1. **Gereksinimler (Requirements):** Döngünün ilk ve en kritik aşamasıdır. Bu aşamada, güvenlik ekipleri, iş birimleri ve üst düzey yöneticiler bir araya gelerek istihbarat ihtiyaçlarını net bir şekilde tanımlar. Bu, korunması gereken en kritik varlıkları (“Crown Jewels”), organizasyonun karşılaştığı riskleri ve bu riskleri azaltmak için hangi bilgilerin gerekli olduğunu belirlemeyi içerir. Gereksinimlerin net bir şekilde belirlenmesi, istihbarat toplama çabalarının boşa gitmesini engeller ve kaynakların doğru hedeflere yönlendirilmesini sağlar. Aksi halde, ekipler alakasız verileri takip ederek zaman ve kaynak kaybedebilir veya kritik tehditleri gözden kaçırabilir.

2. **Toplama (Collection):** Bu aşama, tanımlanan gereksinimleri karşılamak için geniş bir yelpazedeki kaynaklardan ham veri toplamayı içerir. Toplanan veriler hem teknik (Indicator of Compromise - IOC) hem de bağlamsal bilgiler (TTP’ler, motivasyonlar) içermelidir.

    - **OSINT (Open-Source Intelligence):** Genel kullanıma açık ve serbestçe erişilebilen kaynaklardan bilgi toplama. Bu kaynaklar arasında haberler, bloglar, sosyal medya platformları, akademik çalışmalar ve endüstri raporları bulunur.

- **Ticari Tehdit Beslemeleri:** Güvenlik firmaları tarafından sağlanan ve genellikle otomatik sistemlere entegre edilen ücretli veri akışlarıdır. Bu beslemeler, binlerce IOC ve TTP bilgisini yüksek hacimde sunabilir.

- **Bilgi Paylaşım Toplulukları (ISACs):** Aynı sektördeki veya coğrafi bölgedeki organizasyonların tehdit istihbaratı paylaşımı için bir araya geldiği güvenilir platformlardır.

- **Dahili Kaynaklar:** Organizasyonun kendi güvenlik araçlarından (SIEM, EDR, IDS/IPS), ağ günlüklerinden, uç nokta telemetri verilerinden ve kimlik doğrulama kayıtlarından elde edilen verilerdir.

- **Derin ve Karanlık Web İzleme:** Gizli veya şifreli forumlar, siber suç pazarları ve sızdırılmış veri depolarından (örneğin, sızan kimlik bilgileri, saldırı planları) bilgi toplama.

3. **İşleme (Processing):** Toplanan ham verinin analiz edilebilir, yapılandırılmış ve temiz bir formata dönüştürüldüğü aşamadır. Bu, veri normalizasyonu (farklı formatlardaki verileri standartlaştırma), tekilleştirme (yinelenen kayıtları kaldırma), şifre çözme ve etiketleme işlemlerini içerir. Büyük veri kümeleri için otomasyon, bu aşamada hayati öneme sahiptir.

4. **Analiz ve Yorumlama (Analysis and Interpretation):** İşlenmiş verinin anlamlı, eyleme dönüştürülebilir istihbarata dönüştüğü aşamadır. Analistler, kalıpları, eğilimleri, saldırı kampanyalarını ve potansiyel tehditleri belirlemek için verileri derinlemesine incelerler. Bu aşamada, insan uzmanlığı ve otomatik korelasyon mekanizmaları birleşir.

5. **Yayma (Dissemination):** Analiz edilen istihbaratın ilgili paydaşlara zamanında ve uygun bir formatta sunulmasıdır. İstihbaratın formatı ve içeriği, hedef kitleye göre uyarlanmalıdır. Örneğin, CISO'lar için üst düzey yönetici brifingleri hazırlanırken, SOC analistleri için teknik IOC beslemeleri sunulur.

6. **Geri Bildirim (Feedback):** Sürecin sürekli iyileştirilmesini sağlayan bu aşamada, istihbarat tüketicileri (örneğin, SOC analistleri), sunulan bilgilerin yararlılığı, alaka düzeyi ve doğruluğu hakkında geri bildirimde bulunur. Bu geri bildirim, bir sonraki döngüde daha iyi gereksinimlerin belirlenmesini ve istihbarat toplama stratejilerinin hassaslaştırılmasını sağlar.

Bu yaşam döngüsü, bir organizasyonun reaktif tehdit algılama modelinden proaktif bir savunma duruşuna geçiş yapmasına olanak tanır. Bu sürekli ve döngüsel süreç, savunma mekanizmalarının saldırganların evrilen taktik, teknik ve prosedürlerine (TTP'ler) göre dinamik olarak ayarlanmasını sağlar. Bu dinamik adaptasyon, bir saldırganın bir sistemde kalma süresini (dwell time) doğrudan azaltır. Saldırganın sistemde kalma süresinin azalması, fidye yazılımı veya veri sızdırma gibi nihai hedeflere ulaşma şansını düşürürken, ihlalden kaynaklanan maliyetleri de önemli ölçüde azaltır. Bu sürekli öğrenme ve adaptasyon mekanizması, tehdit istihbaratı programının değerini ve etkinliğini katlanarak artırır.

CTI Yaşam Döngüsü Aşamaları ve Uygulamaları

Aşama	Kısa Tanım	Hedef	Pratik Uygulama Örnekleri
Gereksinimler	İhtiyaç duyulan istihbaratın belirlenmesi.	Kaynakları en kritik risklere odaklamak.	Finansal dolandırıcılık veya veri sızıntılarını hedef alan tehdit aktörlerinin TTP'leri hakkında bilgi toplama.
Toplama	Ham verinin farklı kaynaklardan alınması.	Tanımlanan gereksinimlere yönelik verileri toplama.	SIEM günlükleri, OSINT, karanlık web forumları, ticari tehdit beslemeleri, ISAC'ler.
İşleme	Toplanan ham veriyi analiz edilebilir formata getirme.	Veriyi yapılandırma, tekilleştirme ve zenginleştirme.	CSV dosyalarının normalizasyonu, VirusTotal'dan dosya karması zenginleştirme.
Analiz	İşlenmiş veriden eyleme dönüştürülebilir içgörüler çıkarma.	Tehditleri bağlamsallaştırma, kalıpları ve eğilimleri belirleme.	Bir oltalama kampanyasında kullanılan yeni bir tekniğin analizi.
Yayma	Elde edilen istihbaratı ilgili paydaşlara iletme.	Bilginin doğru kitleye, doğru zamanda ve doğru formatta sunulması.	C-seviyesi için yönetici brifingi, SOC analistleri için IoC beslemesi, otomatik güvenlik duvarı kuralı.
Geri Bildirim	İstihbaratın yararlılığı ve etkinliği hakkında geri bildirim toplama.	Sürekli iyileştirmeyi teşvik etme.	SOC ekibinden gelen istihbaratın yanlış pozitif oranına dair rapor, olay sonrası değerlendirme toplantıları.

8.1.2 Strategic, Tactical, Technical ve Operational Intelligence

Siber tehdit istihbaratı, hedef kitlenin ihtiyaçlarına göre dört ana kategoriye ayrılır. Bu istihbarat türleri, bir organizasyonun savunma yeteneklerini tüm seviyelerde güçlendirmek için birlikte çalışır.

- **Stratejik İstihbarat:** Genel tehdit ortamına ilişkin üst düzey bir bakış açısı sunar. Bu istihbarat, teknoloji dışı terimlerle hazırlanır ve öncelikli olarak üst düzey yöneticilere, CISO'lara ve risk yöneticilerine yöneliktir.
  - **Amacı:** Güvenlik yatırımları, bütçe tahsisi ve kurumsal politikalar gibi uzun vadeli stratejik kararlara rehberlik etmektir.
  - **Örnekler:** Bir APT grubunun bir sektörü veya belirli bir coğrafyayı hedeflemesi, fidye yazılımı eğilimleri veya jeopolitik olayların siber saldırı risklerine etkisi hakkında raporlar.
- **Operasyonel İstihbarat:** Belirli bir tehdit aktörünün TTP'lerini, motivasyonlarını ve altyapısını anlamayı sağlar. Bu istihbarat, olay müdahale (IR) ekipleri ve tehdit avcıları için hayati öneme sahiptir.
  - **Amacı:** Yaklaşan veya devam eden bir saldırının "kim, ne zaman, nerede, nasıl, neden" sorularına bağlamsal cevaplar sunmak. Bu, proaktif tehdit avcılığı ve olay müdahale planlaması için bir temel oluşturur.
  - **Örnekler:** Bir saldırıda kullanılan belirli bir oltalama e-postası kampanyasının detayları veya bir Komuta ve Kontrol (C2) sunucusunun iletişim yöntemleri.
- **Taktik İstihbarat:** Ağ ve uç noktalarda tehditleri tespit etmeye yardımcı olan, genellikle kısa ömürlü ve teknik göstergelerden oluşan bir istihbarat türüdür.
  - **Amacı:** Otomatik tehdit tespiti için SIEM ve güvenlik duvarı gibi güvenlik kontrollerine entegre edilmek.
  - **Örnekler:** Kötü amaçlı IP adresleri, dosya karmaları (hashes), kötü amaçlı alan adları.
- **Teknik İstihbarat:** Taktik istihbaratın daha derin teknik detaylarını içerir ve genellikle kötü amaçlı yazılım analizi, tersine mühendislik ve IoC'lerin oluşturulmasını kapsar.

– **Amacı:** Saldırıların teknik işleyişini anlamak ve bu bilgilere dayanarak yeni imza tabanlı tespit kuralları oluşturmaktır.

Bu dört istihbarat türü, birbirini besleyen ve destekleyen bir hiyerarşi içinde çalışır. Üst yönetim, genel güvenlik stratejisini belirlemek için stratejik istihbarata ihtiyaç duyar. Bu strateji, operasyonel ve taktiksel gereksinimlere dönüşür. Örneğin, stratejik istihbarat, jeopolitik gerilimlerin bir APT grubunun faaliyetlerini artıracağını gösterdiğinde, operasyonel ekip bu grubun TTP'lerini incelemeye odaklanır ve bu operasyonel bilgiler taktiksel IoC'lere dönüştürülerek güvenlik kontrollerine entegre edilir. Bu entegrasyon, istihbaratın tüm organizasyonel katmanlarda değer yaratmasını sağlar.

İstihbarat Türleri Karşılaştırma Tablosu

Özellik	Stratejik	Operasyonel	Taktik
Hedef Kitle	C-seviyesi yöneticiler, iş liderleri	Olay müdahale (IR) ekipleri, tehdit avcıları	SOC analistleri, güvenlik mühendisleri
Odak Noktası	Genel eğilimler, motivasyonlar, risk yönetimi	Saldırı kampanyaları, TTP'ler, saldırganın hedefleri	IOC'ler (IP, hash, alan adı), ağ etkinlikleri
Zaman Çerçevesi	Uzun vadeli (aylar, yıllar)	Orta vadeli (haftalar, aylar)	Kısa vadeli (gerçek zamanlı)
Amaç	Güvenlik yatırımlarını ve politikalarını belirleme	Saldırılara müdahale etme ve proaktif avcılık yapma	Otomatik tespit kuralları oluşturma ve tehditleri engelleme

8.1.3 Threat Actor Profiling ve Attribution Challenges

Tehdit aktörü profilleme, saldırıların arkasındaki "kim" sorusuna cevap aramayı amaçlar. Bir tehdit aktörü profili, sadece bir isimden ibaret değildir; bir saldırganın kimliğini, hedeflerini, TTP'lerini, motivasyonlarını, coğrafi konumunu ve kullandığı altyapıyı kapsamlı bir şekilde analiz eder. Bu profiller, güvenlik ekiplerine saldırganın olası davranışları hakkında bir resim sunar ve savunma stratejilerini saldırganın niyetleriyle uyumlu hale getirmeye yardımcı olur.

Ancak, bir siber saldırıyı kesin olarak belirli bir tehdit aktörüne atfetmek (attribution), son derece karmaşık ve zorlu bir süreçtir. Bu zorlukların birkaç temel nedeni vardır:

- **Gizleme ve Obfüskasyon:** Saldırganlar, kimliklerini gizlemek için çok sayıda katman kullanır. Botnetler, vekil sunucular ve kiralanmış altyapı, saldırının gerçek kaynağını maskeler. Özellikle dağıtık servis dışı bırakma (DDoS) saldırılarında, trafik binlerce farklı cihazdan gelebilir, bu da kesin bir atıf yapmayı neredeyse imkansız hale getirir.
- **Yanlış Bayrak Operasyonları:** Bazı siber suç grupları, dikkat çekmek veya yanlış bilgi yaymak amacıyla gerçekleştirmediği saldırıların sorumluluğunu üstlenebilir. Bu tür eylemler, atıf sürecini manipüle etmeyi amaçlar.
- **Araç ve TTP Paylaşımı:** Tehdit grupları, sıklıkla araç setlerini ve TTP'lerini birbirleriyle paylaşır veya değiştirir. Bir saldırıda kullanılan belirli bir teknik, daha önce bilinen bir grubun imzası olsa bile, başka bir grup tarafından da kullanılmış olabilir. Bu durum, yalnızca teknik IOC'lere dayalı atıf yapmanın güvenilirliğini azaltır.

Bu nedenle, atıf birincil hedef olmamalıdır. Bir saldırı meydana geldiğinde, ilk ve en acil öncelik, hasarı durdurmak, sistemleri güvence altına almak ve devam eden tehditleri ortadan kaldırmaktır. Atıf, bu acil riskler ortadan kaldırıldıktan sonra, olay sonrası analiz aşamasında daha sonraki savunmaları güçlendirmek için bir araç olarak ele alınmalıdır. Atıf, bir "evet/hayır" cevabı yerine, farklı güven seviyelerine sahip (düşük, orta, yüksek) analitik bir değerlendirme süreci olarak görülmelidir. Güvenilir bir atıf için, sadece IOC'ler yerine, davranışsal kanıtlar ve TTP'lere odaklanmak daha geçerli bir yaklaşımdır.

8.1.4 Diamond Model ve Kill Chain Analysis

Siber saldırıları anlamak ve analiz etmek için kullanılan iki önemli analitik çerçeve, Lockheed Martin'in Siber Kill Chain'i ve MITRE'nin Saldırı Analizinin Elmas Modeli'dir (Diamond Model of Intrusion Analysis). Bu modeller, saldırı sürecini farklı açılardan ele alarak birbirini tamamlar.

- **Siber Kill Chain (Siber Saldırı Zinciri):** Bu model, bir saldırının yedi aşamalı, doğrusal bir sürecini sunar. Her aşama, bir saldırganın başarılı bir siber saldırı gerçekleştirmek için tipik olarak izlediği adımları tanımlar.

  1. **Keşif (Reconnaissance):** Saldırgan, hedefin sistemleri, ağ yapısı ve çalışanları hakkında bilgi toplar.
  2. **Silahlandırma (Weaponization):** Bir exploit ve bir arka kapı (payload) bir araya getirilerek tek bir saldırı paketi oluşturulur.
  3. **Teslimat (Delivery):** Saldırı paketi, hedef sisteme iletilir (örneğin, oltalama e-postası veya kötü amaçlı web sitesi aracılığıyla).
  4. **İstismar (Exploitation):** Bir sistemdeki zafiyet kullanılarak ilk erişim elde edilir.
  5. **Kurulum (Installation):** Saldırgan, sistemde kalıcılığı sağlamak için arka kapıyı kurar.
  6. **Komuta ve Kontrol (Command and Control):** Saldırganın ele geçirilmiş sistemle uzaktan iletişim kurarak onu kontrol etmesini sağlar.
  7. **Hedefler Üzerindeki Eylemler (Actions on Objectives):** Saldırgan, veri sızdırma, sistem bozulması veya fidye gibi nihai hedeflerine ulaşır.

  Kill Chain, saldırıların nasıl ilerlediğine dair adım adım bir yol haritası sunar ve özellikle olay müdahalesi ve taktiksel savunma kararları için çok değerlidir.

- **Diamond Model of Intrusion Analysis (Saldırı Analizinin Elmas Modeli):** Bu model, siber saldırıları dört temel bileşen arasındaki ilişkilere odaklanarak inceler. Doğrusal bir model değildir, daha çok saldırının bütüncül bir resmini sunar.

  1. **Saldırgan (Adversary):** Saldırının arkasındaki tehdit aktörü.
  2. **Kabiliyet (Capability):** Saldırganın kullandığı araçlar, teknikler ve yöntemler.
  3. **Altyapı (Infrastructure):** Saldırıyı desteklemek için kullanılan ağ altyapısı (örneğin, C2 sunucuları, vekil sunucular).
  4. **Kurban (Victim):** Saldırıya uğrayan kişi, organizasyon veya sistem.

  Elmas Modeli, tehdit istihbaratı ve proaktif tehdit avcılığı için daha kullanışlıdır. Saldırıdan elde edilen herhangi bir bilgi parçası (örneğin, bir C2 sunucusu), diğer üç bileşenle ilişkilendirilerek saldırganın potansiyel diğer faaliyetlerini ve gelecekteki olası hedeflerini tahmin etmek için kullanılabilir.

Kill Chain, bir saldırının *nasıl* ilerlediğini detaylandırırken, Diamond Model saldırının *neden* ve *kim tarafından* yapıldığına dair daha geniş bir bakış açısı sunar. Bu modeller birbirini tamamlar. Bir saldırı tespit edildiğinde, Kill Chain modeli olayın taktiksel olarak yönetilmesine yardımcı olurken, Diamond Model olayın bileşenlerini analiz ederek bu olayı bilinen diğer tehdit gruplarıyla ilişkilendirmeyi ve gelecekteki potansiyel saldırıları tahmin etmeyi mümkün kılar.

Kill Chain ve Diamond Model Karşılaştırma Tablosu

Özellik	Siber Kill Chain	Diamond Model
Odak Noktası	Saldırı aşamaları ve süreci	Saldırgan, kabiliyet, altyapı ve kurban arasındaki ilişkiler
Granülarite	Daha spesifik ve doğrusal	Daha geniş, bağlamsal ve ilişkisel
Uygulama Alanı	Olay müdahalesi ve taktiksel savunma	Tehdit istihbaratı ve proaktif tehdit avcılığı
Avantajlar	Saldırıyı durdurmak için adım adım yol haritası sağlar	Saldırganın motivasyon ve kabiliyetlerini anlama olanağı sunar

8.1.5 Intelligence Requirements ve Priority Intelligence Requirements (PIR)

İstihbarat gereksinimlerini tanımlamak, CTI yaşam döngüsünün ilk ve en önemli adımıdır. Priority Intelligence Requirements (PIR), bir organizasyonun en acil ve stratejik bilgi ihtiyaçlarını belirleyen, odaklanmış ve eyleme dönüştürülebilir sorulardır. PIR'ler, istihbarat toplama çabalarını yönlendirir ve kaynakların en kritik risklere odaklanmasını sağlar.

Gereksinim Geliştirme Süreci (Senaryo Odaklı):

Bir finansal kurumun siber güvenlik ekibinin, yeni bir dijital bankacılık platformu başlatmaya hazırlandığını varsayalım. Ekibin ve üst yönetimin temel endişesi, finans sektörünü hedef alan ve müşteri verilerini sızdırmayı amaçlayan saldırılarla ilgili riskleri proaktif olarak anlamaktır.

1. **Stratejik Hedefleri Anlama:** Kuruluşun stratejik hedefi, yeni bir platformu başarıyla başlatmak ve müşteri güvenini korumaktır. Bu hedefle doğrudan ilişkili risk, veri sızıntıları ve finansal dolandırıcılık potansiyidir.
2. **Risk Değerlendirmesi Yapma:** Kuruluşun en büyük riskleri, kimlik avı (phishing) saldırılarıyla kimlik bilgilerinin çalınması, sıfır gün (zero-day) zafiyetlerinin istismarı ve tedarik zincirine (supply chain) yönelik saldırılar olarak belirlenir.
3. **PIR'leri Formüle Etme:** Bu riskler, belirli, tekil ve eyleme dönüştürülebilir sorulara dönüştürülür.
    - **PIR 1:** "Son 6 ayda finans sektörünü hedef alan APT grupları tarafından en sık kullanılan kimlik avı TTP'leri nelerdir?"
    - **PIR 2:** "Organizasyonumuzun tedarik zinciri içinde bulunan üçüncü taraf yazılımlar, bilinen APT grupları tarafından en sık istismar edilen zafiyetleri barındırmakta mıdır?"
    - **PIR 3:** "Yeni dijital bankacılık platformumuzla ilgili olarak, derin ve karanlık webde hedef alınma niyetini gösteren herhangi bir tartışma veya plan var mıdır?"

Bu PIR'lar, istihbarat ekibine bir görev verir. Örneğin, PIR 1'i yanıtlamak için analistin sadece kimlik avı IOC'lerini toplamasını değil, aynı zamanda ilgili saldırıların davranışsal özelliklerini (örneğin, belirli bir spear-phishing tekniği, kullanılan araçlar) araştırmasını da gerektirir. Bu yaklaşım, toplanan verinin, organizasyonun gereksinimlerine göre filtrelenmesini ve analiz edilmesini zorunlu kılar, böylece istihbarat üretim süreci optimize edilir ve üretilen istihbarat daha yüksek bir değer taşır.

PIR Geliştirme Şablonu

Stratejik Hedef	**İlişkili Risk**	**PIR Sorusu**
Müşteri güvenliğini korumak ve veri sızıntılarını önlemek.	Müşteri veritabanına yetkisiz erişim.	"Finans sektörünü hedefleyen ve hassas müşteri verilerini çalan APT gruplarının bilinen TTP'leri nelerdir?"
Yeni dijital platformun güvenli bir şekilde başlatılması.	Platformun dış kaynaklı bileşenlerindeki zafiyetler.	"Dijital bankacılık platformumuzun kullandığı üçüncü taraf kütüphanelerde sıfır gün zafiyetleri hedefleyen saldırgan grupları var mıdır?"
Marka itibarını korumak ve dolandırıcılığı önlemek.	Oltalama (phishing) ve marka sahteciliği kampanyaları.	"Oltalama kampanyaları oluşturmak için en sık taklit edilen finansal markalar hangileridir ve bu kampanyalarda hangi teknikler kullanılmaktadır?"

8.2 Threat Intelligence Platforms ve Standards

Tehdit İstihbarat Platformları (TIP'ler), siber tehdit istihbaratını toplamak, analiz etmek ve paylaşmak için kullanılan merkezi sistemlerdir. Bu platformlar, farklı kaynaklardan gelen tehdit verilerini bir araya getirir, bunları normalleştirir ve güvenlik ekiplerinin eyleme geçirilebilir istihbarat elde etmesini sağlar. Tehdit istihbaratı standartları ise, farklı

sistemler ve kuruluşlar arasında tehdit bilgilerinin tutarlı bir şekilde paylaşılmasını sağlayan ortak formatlar ve protokollerdir.

8.2.1 MITRE ATT&CK Framework Integration

MITRE ATT&CK, saldırganların taktik, teknik ve prosedürlerini (TTP'ler) gerçek dünya gözlemlerine dayalı olarak kategorize eden küresel bir bilgi bankasıdır. Bu çerçeve, güvenlik profesyonellerine bir saldırının "nasıl" ve "neden" gerçekleştiğini anlamaları için ortak bir dil ve yapı sunar.

- **Yapısal Analiz:** ATT&CK, saldırgan davranışını hiyerarşik olarak düzenler:
  - **Taktikler (Tactics):** Saldırganın yüksek seviyeli hedefleri (örneğin, İlk Erişim, Kalıcılık, Yan Hareket).
  - **Teknikler (Techniques):** Taktiklere ulaşmak için kullanılan belirli yöntemler (örneğin, Kimlik Avı Bağlantısı, Uzaktan Hizmetler).
  - **Prosedürler (Procedures):** Tekniklerin gerçek dünya saldırılarında nasıl uygulandığının somut örnekleri.
- **Entegrasyon ve Uygulamalı Kullanım:**
  - **Tehdit Modelleme:** ATT&CK, bir saldırganın sistem içinde nasıl hareket edebileceğini haritalamak ve potansiyel saldırı yollarını görselleştirmek için kullanılır.
  - **Savunma Boşluk Analizi:** Güvenlik kontrollerinin hangi teknikleri tespit edip engelleyebildiğini belirleyerek savunmadaki zayıf noktaları ortaya çıkarır. Bu analiz, güvenlik yatırımlarının doğru alanlara yönlendirilmesine yardımcı olur.
  - **Tehdit Avcılığı:** Tehdit avcıları, hipotezlerini belirli ATT&CK tekniklerine dayandırarak avlanma süreçlerini yapılandırabilir.
- **Pratik Senaryo (TTP Analizi):** Bir saldırganın, bir yöneticinin kimlik bilgilerini çaldığını ve ağda yanal hareket ettiğini varsayalım.
  - **Saldırı Olayı:** Yönetici, oltalama (phishing) e-postasıyla kötü amaçlı bir bağlantıya tıklar. Kimlik bilgileri çalınır. Saldırgan, bu kimlik bilgilerini kullanarak ağda başka bir sunucuya RDP (Uzak Masaüstü Protokolü) ile bağlanır.
  - **ATT&CK Eşleştirme:**
    - **Taktik: Initial Access (İlk Erişim)** → **Teknik: Spearphishing Link (Kimlik Avı Bağlantısı)** (T1566.002).
    - **Taktik: Credential Access (Kimlik Bilgisi Erişimi)** → **Teknik: OS Credential Dumping** (T1003) veya **Credential from Web Browsers** (T1555).
    - **Taktik: Lateral Movement (Yan Hareket)** → **Teknik: Remote Services** (T1021) veya **Remote Desktop Protocol** (T1021.001).

ATT&CK, güvenlik ekipleri için sadece bir teknik listesi olmaktan öteye geçer ve saldırgan davranışını standartlaştıran bir ortak dil haline gelir. Bu standardizasyon, farklı güvenlik birimleri (mavi takım, kırmızı takım, tehdit istihbaratı) arasındaki iletişimi kolaylaştırır. Olay müdahale planlarının ve otomatik yanıtların geliştirilmesini kolaylaştırır ve ekiplerin daha etkili bir şekilde işbirliği yapmasına olanak tanır. Bu çerçeve, güvenlik stratejisinin IOC'lere bağımlı olmaktan TTP'lere odaklanmaya kaymasını teşvik eder. Saldırganlar IP adresleri veya dosya karmaları gibi IOC'leri kolayca değiştirebilirken, yetki yükseltme veya kalıcılık elde etme yöntemleri (TTP'ler) daha kalıcı ve ayırt edicidir. Bu, daha uzun vadeli ve dayanıklı bir savunma sağlar.

ATT&CK Saldırı Eşleştirme Tablosu

Saldırı Aşaması	ATT&CK Taktikleri	ATT&CK Teknikleri	Örnek Davranış
İlk Erişim	Initial Access	Spearphishing Link (T1566.002)	Yöneticinin oltalama e-postasına tıklaması.
Keşif	Discovery	Network Service Scanning (T1046)	Saldırganın ağda açık portları taraması.
Kimlik Bilgisi Erişimi	Credential Access	OS Credential Dumping (T1003)	Saldırganın bellekten parolaları çekmesi.
Yan Hareket	Lateral Movement	Remote Services (T1021)	Saldırganın çalıntı kimlik bilgileriyle bir RDP oturumu başlatması.
Kalıcılık	Persistence	Scheduled Task/Job (T1053.005)	Saldırganın gelecekteki erişim için zamanlanmış bir görev oluşturması.

8.2.2 STIX/TAXII Standards ve Information Sharing

STIX ve TAXII, siber tehdit istihbaratını standart bir formatta paylaşmak ve otomatikleştirmek için geliştirilmiş uluslararası standartlardır. Bu standartlar, farklı platformlar ve organizasyonlar arasında kesintisiz bir bilgi akışı sağlayarak, tehdit istihbaratının değerini artırır.

- **STIX (Structured Threat Information eXpression):** Tehdit istihbaratını tanımlamak için kullanılan JSON tabanlı, makine tarafından okunabilir bir dildir. STIX, saldırı motivasyonları, kabiliyetleri, IOC'ler ve TTP'ler gibi tehdit verilerini tutarlı bir şekilde ifade etmeyi sağlar.
- **TAXII (Trusted Automated eXchange of Intelligence Information):** STIX verilerinin güvenli ve otomatik olarak nasıl aktarılacağını tanımlayan bir protokoldir. TAXII, bir tehdit beslemesinden yeni istihbaratın çekilmesini (pull) veya itilmesini (push) mümkün kılar.

Pratik Senaryo ve Örnek (Python): Bir tehdit istihbaratı beslemesinden alınan bir STIX nesnesinin Python'da nasıl oluşturulacağını gösteren basit bir örnek. Bu örnek, `stix2` kütüphanesini kullanarak bir `AttackPattern` nesnesinin nasıl oluşturulduğunu ve bir `ThreatActor`'e nasıl bağlandığını gösterebilir. Bu, STIX'in veriyi yapılandırmadaki rolünü vurgular.

```
import stix2
from stix2 import AttackPattern, ThreatActor

# Basit bir saldırı tekniği tanımlama
attack_pattern = stix2.AttackPattern(name ="Kimlik Avı Bağlantısı",
    external_references =)

# Bir tehdit aktörü tanımlama
threat_actor = stix2.ThreatActor(name ="Örnek Tehdit Aktörü",
    description ="Finansal kazanç amaçlı faaliyet gösteren siber suç grubu.",
    aliases =)

# Saldırı tekniği ile tehdit aktörünü ilişkilendirme
relationship = stix2.Relationship(relationship_type ="uses",
    source_ref =threat_actor.id,
    target_ref =attack_pattern.id)

# Nesneleri bir pakete ekleme
```

```
bundle = stix2.Bundle(attack_pattern, threat_actor, relationship)

print(bundle.serialize(pretty =True))
```

Bu örnek, STIX'in veriyi sadece bir liste olarak değil, aynı zamanda ilişkileri ve bağlamı ile birlikte yapılandırmadaki önemini gösterir. Bu standartlar sayesinde, istihbarat paylaşımı manuel yöntemlerden (e-posta, PDF raporları) makine tarafından okunabilir ve otomatikleştirilebilir bir sürece taşınır, bu da reaksiyon süresini önemli ölçüde kısaltır ve güvenlik operasyonlarının verimliliğini artırır.

8.2.3 Threat Intelligence Platform (TIP) Selection ve Implementation

Tehdit istihbaratı platformları (TIP), çeşitli kaynaklardan gelen tehdit verilerini merkezileştiren, işleyen ve eyleme dönüştürülebilir istihbarata dönüştüren yazılımlardır. Bir TIP'nin değeri, topladığı ham veri miktarından ziyade, bu verileri güvenlik ekibine eyleme dönüştürülebilir bir şekilde sunma yeteneğiyle ölçülür.

Seçim ve Uygulama Kılavuzu:

1. **Gereksinimleri Tanımlama:** Organizasyonun hangi istihbarat türlerine (stratejik, taktik, operasyonel) en çok ihtiyaç duyduğunu belirleyin. Bir olay müdahale ekibinin öncelikleri, üst yönetimin önceliklerinden farklıdır.
2. **Mevcut Ortamı Değerlendirme:** Hangi SIEM/SOAR platformlarının, EDR çözümlerinin ve veri kaynaklarının mevcut olduğunu değerlendirin. TIP'nin mevcut altyapı ile entegrasyonu, operasyonel verimlilik için kritik öneme sahiptir.
3. **TIP Seçimi:** Birden fazla istihbarat türünü entegre edebilen, iyi API ve bağlayıcılara sahip ve geniş bir veri kapsamı sunan bir platform arayın. Yalnızca göstergelere dayalı bir platform, sofistike ve sürekli değişen tehdit ortamına karşı yetersiz kalır. Gerçek değer, saldırganın motivasyonunu, kabiliyetini ve TTP'lerini anlamak için verileri bağlamsallaştıran platformlardan gelir.
4. **Veri Alımını Yapılandırma:** Dahili ve harici beslemelerden (örneğin, ticari beslemeler, OSINT, dahili günlükler) veri alımını ayarlayın ve bu verileri standart bir formata dönüştürecek normalizasyon kurallarını oluşturun.
5. **İş Akışlarını ve Otomasyon Kurallarını Oluşturma:** Tehditlerin nasıl önceliklendirileceği, zenginleştirileceği ve diğer güvenlik araçlarına nasıl aktarılacağı için otomasyon kuralları oluşturun. Bu, olay müdahale süreçlerini hızlandırır.
6. **Test ve Optimizasyon:** Uçtan uca iş akışlarını gerçek dünya senaryolarıyla test edin ve geri bildirimlerle sistemi sürekli optimize edin.

Saldırganlar, kullandıkları araçları ve altyapıyı hızla değiştirebildiğinden, yalnızca dosya karmaları ve IP adresleri gibi IOC'leri takip eden bir TIP, yeni veya bilinmeyen tehditlere karşı yetersiz kalır. Bir TIP'nin gerçek değeri, veriyi bağlamsallaştırma ve saldırgan davranışını anlama yeteneğinden gelir.

8.2.4 Indicators of Compromise (IOC) Management

Indicators of Compromise (IOC), bir sistemin veya ağın tehlikeye atıldığının dijital kanıtıdır. IOC'ler, bir saldırının "olduğunu" gösteren adli kanıtlardır ve olay sonrası adli analizler için çok değerlidir.

- **Yaygın IOC Örnekleri:**
  - **Ağ Trafiği Anormallikleri:** Normalin dışında giden ağ trafiği veya olağandışı port ve protokol kullanımı.
  - **Kimlik Bilgisi Anomalileri:** Olağandışı saatlerde veya coğrafyalardan gelen oturum açma denemeleri.
  - **Dosya Karmaları (Hashes):** Bilinen kötü amaçlı yazılım dosyalarının benzersiz SHA256 veya MD5 karmaları.

  - **Sistem Değişiklikleri:** Şüpheli kayıt defteri (registry) veya sistem dosyası değişiklikleri.

Yönetim ve Playbook Örneği: Bir tehdit istihbaratı beslemesinden yeni bir IOC alındığında, otomatik bir playbook tetiklenebilir. Bu otomasyon, güvenlik ekiplerinin reaksiyon süresini önemli ölçüde kısaltır ve insan hatasını azaltır.
Senaryo: Yeni bir oltalama alan adı (domain) IOC olarak alınmıştır.

1. **Tetkikleme:** TIP, yeni alan adını alır ve otomatik olarak risk puanı atar.
2. **Zenginleştirme:** SOAR platformu, bu alan adını VirusTotal gibi kaynaklarda sorgulayarak ek bağlam (ilgili IP adresleri, diğer kötü amaçlı yazılım kampanyaları) toplar.
3. **Otomatik Yanıt:** Bu alan adı yüksek riskli olarak işaretlendiğinde, SOAR platformu otomatik olarak güvenlik duvarlarına ve ağ filtreleme çözümlerine bu alanı engelleme talimatı gönderir.
4. **Uyarı ve Eşleştirme:** SIEM'de bu IOC ile ilgili geçmiş günlükler taranır ve ilgili tüm olaylara bir uyarı gönderilir. Tespit edilen her olay, bir SOC analisti için otomatik olarak bir olay bileti oluşturur.

IOC'ler, bir saldırının *olduğunu* gösterirken, saldırı göstergeleri (Indicators of Attack - IoA), bir saldırının *muhtemelen olacağını* gösterir. Bu ayrım, reaktif güvenlikten (olay sonrası) proaktif güvenliğe (olay öncesi) geçişin anahtarıdır. Örneğin, bir kullanıcının olağan dışı sayıda başarısız oturum açma denemesi yapması bir IoA'dır ve bu, daha erken bir aşamada müdahale edilerek saldırıyı önleme fırsatı sunar.

8.2.5 Tactics, Techniques, and Procedures (TTP) Analysis

TTP'ler, saldırganların hedeflerine ulaşmak için kullandıkları yöntemleri detaylandıran temel bileşenlerdir ve bir saldırının "hikayesini" anlatır.

- **TTP Analizinin Önemi:**
  - **Gelişmiş Tehdit Tespiti:** TTP'ler, geleneksel imza tabanlı tespit yöntemlerini atlatmaya çalışan karmaşık saldırıları tespit etmeye yardımcı olur.
  - **Proaktif Savunma:** Saldırgan davranışını anlayarak, kuruluşlar doğrudan bu saldırı yöntemlerini hedef alan güvenlik kontrollerini uygulayabilir.
  - **Saldırgan Atfı:** Belirli TTP'lerin benzersiz kombinasyonları, bir saldırının arkasındaki tehdit aktörünü belirlemek için bir ipucu olabilir.
- **Uygulamalı Analiz Adımları:**
  1. **Olayı TTP'lerle Eşleştirme:** Tespit edilen bir saldırının her adımını (örneğin, ilk erişim, kalıcılık) ATT&CK çerçevesindeki ilgili Taktik ve Tekniklerle eşleştirin.
  2. **Davranışsal Anomali Tespiti:** Bu TTP'leri gerçekleştirmek için kullanılan davranışları (örneğin, bir yöneticinin normalde kullanmadığı bir komutu çalıştırması) belirleyin.
  3. **Özel Algılama Kuralları Oluşturma:** Bu davranışları tespit etmek için SIEM sistemlerinde veya EDR çözümlerinde özel kurallar oluşturun.

SIEM Kural Örneği: Bir saldırganın `whoami.exe` komutunu kullanarak yetki yükseltmeye çalıştığını varsayalım. `index =winlogs sourcetype =WinEventLog:Security EventCode =4688 "CommandLine" =*whoami.exe* | stats count by user, host | sort -count`. Bu basit sorgu, bir saldırganın "keşif" (discovery) taktiğini kullandığını gösterir ve daha fazla araştırma için bir başlangıç noktası sunar.

TTP analizi, geleneksel güvenlik yöntemlerinin sadece "ne oldu?" sorusuna odaklandığı durumlarda, bir saldırının "nasıl" ve "neden" yapıldığını anlamamızı sağlar. Bu, güvenlik ekiplerinin sadece reaktif olarak yanıt vermek yerine, bir sonraki adımı tahmin ederek proaktif bir savunma stratejisi geliştirmesini mümkün kılar.

TTP Odaklı Tespit Kuralı Geliştirme Tablosu

Taktik / Teknik	İlişkili Davranışsal Anomali	Örnek SIEM Sorgusu (Splunk)
Lateral Movement (Yan Hareket)	Kullanıcının normalde erişmediği bir sunucuda oturum açması.	`index =winlogs sourcetype =WinEventLog:Security EventCode =4624 AND Logon_Type =3 NOT user ="*$" | stats count by user, host, src_ip | sort -count`
Credential Access (Kimlik Bilgisi Erişimi)	`lsass.exe` bellek dökümü almaya çalışan bir süreç.	`index =winlogs sourcetype =WinEventLog:Microsof EventCode =10 AND TargetImage ="C:\Windows\System3`
Defense Evasion (Savunma Atlatma)	`rundll32.exe` ile olağan dışı bir DLL'in çalıştırılması.	`index =winlogs sourcetype =WinEventLog:Security EventCode =4688 AND ParentProcess ="C:\Windows\Syste AND CommandLine!="*C:\Windows\System`

8.3 Threat Hunting Methodologies ve Techniques

Tehdit avcılığı (threat hunting), bir kuruluşun ağında ve sistemlerinde, geleneksel güvenlik araçları tarafından tespit edilemeyen gizli tehditleri proaktif olarak arama sürecidir. Tehdit avcıları, bir ihlalin zaten gerçekleştiği varsayımıyla hareket ederler ve saldırganların izlerini bulmak için çeşitli analitik teknikler ve araçlar kullanırlar. Bu süreç, reaktif bir olay müdahale yaklaşımından daha proaktif bir savunma duruşuna geçişi temsil eder.

8.3.1 Hypothesis-driven Threat Hunting Approaches

Hipotez tabanlı tehdit avcılığı, rastgele veri arama süreçleri yerine, belirli bir teori veya varsayıma dayalı olarak tehdit arama sürecini yönlendirir. Bu yaklaşım, tehdit avcılarının çabalarını daha verimli ve amaç odaklı hale getirir.

Hipotez Oluşturma Adımları:

1. **Tehdit İstihbaratına Dayalı Hipotez:** Güvenilir bir tehdit beslemesinden, sektörünüzü hedefleyen yeni bir fidye yazılımı grubunun kullandığı TTP'lerle ilgili bilgi edindiğinizi varsayın. Saldırganlar, kalıcılık için Windows Kayıt Defteri (Registry) çalıştırma anahtarlarını değiştirmeye çalışıyor olabilir.

2. **Senaryo Geliştirme:** "Bir saldırgan, kimlik avı yoluyla elde ettiği kimlik bilgilerini kullanarak ağımızda yanal hareket ediyor ve kalıcılık için kayıt defterini manipüle ediyor olabilir".

3. **Veri Kapsamını ve Sorgu Planını Belirleme:** Bu hipotezi test etmek için, uç nokta (endpoint) günlüklerinden Windows Kayıt Defteri değişikliklerini izleyen verileri toplayın ve yetki yükseltme olaylarına odaklanın.

Pratik Senaryo ve Örnek (Splunk/SIEM Sorgusu):

- **Hipotez:** "Bir saldırgan, yetki yükseltme taktiği olarak `whoami.exe` gibi sistem araçlarını kullanıyor."
- **Sorgu Planı:** Olağandışı bir sürecin `whoami.exe` komutunu çalıştırmasını arayın.
- **Örnek Splunk Sorgusu:**

```
index =winlogs sourcetype ="WinEventLog:Security" EventCode =4688
NOT (ParentProcess ="C:\\Windows\\System32\\cmd.exe"
```

```
OR ParentProcess ="C:\\Windows\\System32\\powershell.exe")
AND process_name ="whoami.exe" | stats count by user, host, ParentProcess | where count > 10
```

Bu sorgu, bir komut satırı veya PowerShell süreci tarafından başlatılmayan, anormal bir şekilde çok sayıda `whoami.exe` çalıştırma olayını arar. Bu, bir saldırganın savunma mekanizmalarını atlatmak için sıradışı bir süreçten bilgi toplama (discovery) tekniği kullandığını gösterebilir.

Hipotez tabanlı avcılık, bir SOC'u reaktif bir uyarı işleme biriminden, proaktif bir soruşturma gücüne dönüştürür. Bu, güvenlik ekiplerinin sürekli olarak gelen alarmları incelemek yerine, bir saldırının muhtemel yolunu varsayarak, otomatik sistemlerin gözden kaçırdığı davranışları aktif olarak aramalarını sağlar. Bu yaklaşım, güvenlik operasyonlarının stratejik olgunluğunu artırır ve analistlerin daha verimli çalışmasına olanak tanır.

8.3.2 Data-driven Hunting ve Statistical Analysis

Veri odaklı avcılık, belirli bir hipotez olmadan büyük veri kümelerindeki anormallikleri veya aykırı değerleri (outliers) bulmak için istatistiksel ve makine öğrenimi yöntemlerini kullanır. Bu yaklaşım, "bilinmeyen-bilinmeyenleri" (unknown-unknowns) bulmaya odaklanır.

Teknikler:

- **Yığınlama (Stack Counting):** Belirli bir veri alanındaki değerlerin frekansını hesaplar ve bu sayede nadir veya yaygın görülen davranışları belirler. Örneğin, bir organizasyon içindeki tüm uç noktalardan belirli bir komutun kaç kez çalıştırıldığını saymak, nadir kullanılan ikili dosyaları ortaya çıkarabilir.
- **Kümeleme (Clustering):** İlgili veri noktalarını belirli kriterlere göre gruplara ayırır. Bu, normal davranışa ait kümelerden önemli ölçüde sapan, anormal davranışları belirlemeye yardımcı olur. Bu teknik, saldırganların "yaşadığı yeri kullanma" (Living-off-the-Land) tekniklerini veya bilinmeyen bir kötü amaçlı yazılımı gizlemek için kullandığı nadir komutları tespit etmek için etkilidir.
- **Korelasyon:** İki veya daha fazla veri seti arasındaki ilişkileri arar. Örneğin, bir kullanıcının olağan dışı saatlerde oturum açmasıyla aynı anda gerçekleşen bir veri alma hacmindeki ani artışı korele edebilir.

Pratik Senaryo ve Örnek (Elastic/SIEM Sorgusu):

- **Senaryo:** Organizasyondaki tüm uç noktalarda yürütülen en nadir süreçleri analiz etmek.
- **Sorgu (Elastic Query Language - EQL):**

```
process where event.action: "process_started" | rare process.executable.name top =100
```

Bu sorgu, ağdaki en nadir yürütülebilir dosyaları belirler. Normalde bir sunucu grubunda görülmemesi gereken bir `.exe` dosyasının çalıştırılması gibi anormal süreçleri ortaya çıkarabilir. Bu, bir saldırganın gizlenmek için kullandığı sıradışı araçları veya yöntemleri tespit etmeye yardımcı olur.

Veri odaklı avcılık, yeni saldırı teknikleri veya daha önce görülmemiş kötü amaçlı yazılımlar gibi, önceden tanımlanmış bir hipotez oluşturmanın zor olduğu durumlerde devreye girer. Bu yöntem, analistlerin daha sonra bir hipotez geliştirebileceği şüpheli aktivitelere odaklanmasını sağlar.

8.3.3 Behavioral Analytics ve Anomaly Detection

Davranışsal analiz, kullanıcı ve sistem davranışlarını zaman içinde izleyerek kötü niyetli faaliyetleri gösterebilecek anlamlı kalıpları belirlemeyi amaçlar. İmza tabanlı güvenlik çözümlerinin kaçırdığı tehditleri (örneğin, sıfır gün zafiyetleri ve dosyasız kötü amaçlı yazılımlar) tespit etmede önemli bir boşluğu doldurur.

Nasıl Çalışır:

1. **Davranışsal Taban Çizgisi (Baseline) Oluşturma:** Bir kullanıcının veya sistemin normal davranışını (örneğin, bir kullanıcının mesai saatlerinde ve coğrafi konumundan oturum açması) belirlemek için geçmiş verileri kullanır.

2. **Anomali Tespiti:** Gerçek zamanlı aktiviteleri bu taban çizgisiyle karşılaştırır. Normalden sapmalar anomali olarak işaretlenir (örneğin, bir hesabın gece 3'te alışılmadık bir ülkeden oturum açması).

3. **Risk Puanlaması:** Anormalliklere, potansiyel tehlike düzeyini belirlemek için dinamik risk puanları atanır.

Uygulama Alanları:

- **İç Tehditlerin Tespiti:** Bir çalışanın normalde erişmediği sistemlere veya verilere erişmeye çalışması gibi olağan dışı davranışları tespit eder.
- **Gelişmiş Kalıcı Tehdit (APT) Tespiti:** APT'ler yavaş hareket ettiğinden ve normal davranışlarla gizlendiğinden, davranışsal analiz bu uzun vadeli ve ince kalıpları ortaya çıkarabilir.
- **Kimlik Bilgisi Hırsızlığı:** Güvenlik altyapısına yetkili bir kullanıcı gibi göründükleri için mevcut araçları atlatabilen, çalınmış kimlik bilgilerini kullanan saldırganları yakalar.

Davranışsal anomali tespiti, yanlış pozitifler (false positives) gibi zorluklar taşısa da, bu yaklaşım dinamik olarak normal faaliyetin tanımını ayarlayarak ve risk puanlarını kullanarak gereksiz uyarıları azaltmaya çalışır. Bu sayede, analistlerin gerçek tehditlere odaklanmasını sağlar ve analist yorgunluğunu azaltır.

Anomali Tespiti Türleri ve Örnekleri

Anomali Türü	Tanım	Siber Güvenlik Örneği
Nokta (Point) Anomalisi	Veri setindeki tek bir noktanın normalden büyük ölçüde sapması.	Bir kullanıcının birden fazla başarısız oturum açma denemesinden sonra aniden başarılı olması.
Bağlamsal (Contextual) Anomali	Bir verinin, belirli bir bağlam veya durum altında anormal olması.	Bir kullanıcının normal mesai saatlerinde sık sık eriştiği bir sunucuya gece yarısı erişmesi.
Kolektif (Collective) Anomali	Bireysel olarak normal görünen bir dizi verinin, birlikte anormal bir desen oluşturması.	Bir DDoS saldırısında olduğu gibi, normal görünen küçük veri akışlarının birleşerek anormal bir trafiği oluşturması.

8.3.4 Hunt Team Organization ve Skill Development

Etkili bir tehdit avcılığı programı, sadece doğru araçlardan ibaret değildir; aynı zamanda doğru becerilere sahip, iyi organize olmuş ve proaktif bir güvenlik ekibini gerektirir. Tehdit avcılığı, bir organizasyonun güvenlik duruşunun olgunluğunu gösterir ve onu reaktif bir yaklaşımdan, aktif ve insan liderliğindeki bir savunma stratejisine taşır.

- **Gerekli Beceriler:**
  - **Saldırgan Zihniyeti:** Tehdit avcıları, potansiyel saldırı yollarını ve zafiyetleri anlamak için bir saldırganın bakış açısını benimsemelidir.
  - **Derin Veri Analizi:** Büyük veri kümelerini analiz etme, korelasyonları bulma ve istatistiksel yöntemleri uygulama yeteneği.
  - **Teknik Uzmanlık:** İşletim sistemleri, ağ protokolleri, SIEM, EDR ve saldırı teknikleri hakkında derinlemesine bilgi.
  - **Programlama Bilgisi:** Otomasyon, özel sorgular oluşturma ve veri işleme için Python veya PowerShell gibi betik (scripting) dillerini kullanma becerisi.

Tehdit avcılığı, genellikle SOC (Güvenlik Operasyon Merkezi) içinde ayrı bir fonksiyon olarak konumlandırılır veya olay müdahale ekibiyle yakın işbirliği içinde çalışır. Düzenli kırmızı takım tatbikatları ve adli analiz çalışmaları, tehdit avcılarının becerilerini geliştirmelerine yardımcı olur.

8.3.5 Threat Hunting Metrics ve Success Measurement

Bir tehdit avcılığı programının değerini kanıtlamak için, başarısını somut ve ölçülebilir metriklerle desteklemek gereklidir. Bu metrikler, güvenlik liderliğine programın yatırım getirisini (ROI) göstermeyi ve sürekli iyileştirmeyi sağlar.

- **Önemli Metrikler:**
  - **Proaktif Olarak Tespit Edilen Olay Sayısı:** Otomatik sistemler tarafından değil, avcılık faaliyetleri sonucunda bulunan olayların sayısıdır. Bu, programın değerini doğrudan gösterir.
  - **İkamet Süresi (Dwell Time) Azaltılması:** Saldırının başlangıcından tespit edilmesine kadar geçen sürenin kısaltılmasıdır. Düşük ikamet süresi, saldırganın sistemde hasar verme şansını azaltır ve ihlal maliyetini doğrudan düşürür.
  - **Ortalama Tespit Süresi (MTTD) ve Ortalama Yanıt Süresi (MTTR):** Bir tehdidin ne kadar sürede tespit edildiğini ve ne kadar sürede tamamen çözüldüğünü gösteren anahtar performans göstergeleridir.
  - **Yanlış Pozitif Oranı:** Avcılık faaliyetleri sonucunda üretilen alarmların ne kadarının sahte olduğunu gösterir. Düşük bir yanlış pozitif oranı, analist yorgunluğunu azaltır ve SOC'un verimliliğini artırır.
  - **Avlanma Başarısı Oranı:** Gerçek bir tehdit tespitiyle sonuçlanan hipotezlerin yüzdesidir. Bu, hipotez oluşturma sürecinin kalitesini gösterir.

Bu metrikler, tehdit avcılığının sadece teknik bir faaliyet olmadığını, aynı zamanda finansal riski ve iş sürekliliğini doğrudan etkileyen stratejik bir iş yatırımı olduğunu kanıtlar.

Tehdit Avcılığı Başarı Metrikleri Tablosu

Metrik	Tanım	Ölçüm Formülü	İş Değeri Katkısı
MTTD Azaltma	Bir tehdidi tespit etmek için gereken ortalama süredeki düşüş.	(Önceki MTTD - Mevcut MTTD) / Önceki MTTD	Saldırganın ağda kalma süresini ve potansiyel hasarı azaltır.
Yanlış Pozitif Azaltma	Yanlış alarmlar nedeniyle harcanan analist süresindeki düşüş.	(Toplam Yanlış Pozitif Sayısı / Toplam Alarm Sayısı) * 100	Analist yorgunluğunu önler ve SOC verimliliğini artırır.
İkamet Süresi	Saldırının başlangıcından tam olarak çözülmesine kadar geçen süre.	(Olayın Çözülme Tarihi - Saldırının Başlangıç Tarihi)	Doğrudan bir ihlalin finansal ve itibar maliyetini düşürür.
Proaktif Olay Sayısı	Otomatik sistemler tarafından değil, avcılık faaliyetleriyle bulunan tehdit sayısı.	Proaktif Tespitle Bulunan Olay Sayısı	Kuruluşun güvenlik duruşunun olgunluğunu gösterir.

8.4 Advanced Persistent Threat (APT) Detection

Gelişmiş Kalıcı Tehditler (APT'ler), belirli bir hedefi uzun süre boyunca gizlice gözetlemek ve veri çalmak amacıyla tasarlanmış, sofistike ve hedefli siber saldırılardır. APT'ler, genellikle devlet destekli veya iyi finanse edilen suç grupları tarafından yürütülür. Bu saldırılar, geleneksel güvenlik önlemlerini atlatmak için özel olarak tasarlanmış kötü amaçlı yazılımlar ve karmaşık taktikler kullanır.

8.4.1 APT Lifecycle ve Long-term Persistence Techniques

APT saldırıları, genellikle uzun bir zaman dilimine yayılan ve hedeflerine ulaşmak için birden fazla aşamayı içeren karmaşık operasyonlardır. APT'lerin temel amacı, hızlı bir etki yaratmak yerine, veri sızdırma veya casusluk gibi uzun vadeli hedefler için sistemde kalıcılık sağlamaktır.

APT Yaşam Döngüsü Aşamaları:

1. **Sızma (Infiltration):** Saldırgan, genellikle oltalama (phishing), sosyal mühendislik veya sıfır gün (zero-day) zafiyetlerini kullanarak ağa ilk erişimi kazanır.
2. **Keşif ve Genişleme (Exploration and Expansion):** İlk erişimden sonra, saldırgan yanal hareket (lateral movement) ile ağın haritasını çıkarır, yetki yükseltir ve hassas verilere erişim arar. Bu aşamada, birden fazla giriş noktası sağlamak için arka kapılar kurulur.
3. **Veri Sızdırma (Exfiltration):** Toplanan veriler, tespit edilmeden ağ dışına, C2 sunucularına aktarılır. Saldırganlar bu aşamada, güvenlik personelini oyalamak için DDoS gibi "beyaz gürültü" (white noise) olayları sahneleyebilir.
4. **Kalıcılık (Maintenance):** Saldırganın ağdaki varlığını uzun süre boyunca gizli tutmasıdır. APT'lerin en ayırt edici özelliklerinden biridir ve bu aşamada kullanılan teknikler tespiti son derece zorlaştırır.

Uzun Süreli Kalıcılık Teknikleri:

- **Rootkit'ler:** Sistemin çekirdek seviyesine yerleşerek kötü amaçlı faaliyetleri ve dosyaları gizler.
- **Kayıt Defteri (Registry) Anahtarları:** Kötü amaçlı yazılımın sistem açılışında otomatik olarak başlatılması için kayıt defterindeki anahtarları değiştirir.
- **Zamanlanmış Görevler (Scheduled Tasks):** Zararlı bir betiği veya programı düzenli aralıklarla çalıştırmak için meşru zamanlanmış görevleri kötüye kullanır.
- **Dosyasız Kötü Amaçlı Yazılımlar (Fileless Malware):** Fiziksel bir dosyayı diske yazmadan doğrudan bellekte çalışan kötü amaçlı yazılımlardır.

APT'ler "düşük ve yavaş" (low and slow) tekniklerini kullandığı için, savunmanın IOC'leri aramaktan ziyade davranışsal anomalilere odaklanması gerekir. Bu, her bir eylemin bireysel olarak fark edilmeyecek kadar küçük olmasını sağlamayı hedefler.

APT Yaşam Döngüsü ve İlişkili TTP'ler Tablosu

APT Yaşam Döngüsü Aşaması	Açıklama	İlişkili TTP'ler ve Teknikler
Sızma (Infiltration)	Ağa ilk erişimin kazanılması.	Oltalama (Phishing), Sosyal Mühendislik, Sıfır Gün İstismarı.
Keşif ve Genişleme	Ağın haritalanması ve yetki yükseltilmesi.	Yanal Hareket, Kimlik Bilgisi Dökümü (Credential Dumping), Ağ Servis Tarama.
Veri Sızdırma (Exfiltration)	Hedeflenen verilerin ağ dışına çıkarılması.	DNS Tünelleme, Bulut Depolama Hizmetleri Kullanımı.
Kalıcılık (Maintenance)	Saldırganın ağdaki varlığının sürdürülmesi.	Rootkit'ler, Kayıt Defteri Değişiklikleri, Zamanlanmış Görevler.

8.4.2 Lateral Movement Detection ve Analysis

Yanal hareket, bir saldırganın ağ içinde ilk erişim noktasından hassas verilere veya sistemlere doğru hareket etme sürecidir. Bu, bir saldırının APT yaşam döngüsündeki en kritik aşamalarından biridir, çünkü saldırgan bu sayede ağ içindeki en değerli varlıkları bulur.

Tespit Yöntemleri ve Analiz (Pratik Örnekler):
Yanal hareket, genellikle normal ağ trafiği veya yönetici faaliyetleri gibi görünmek için meşru araçları kötüye kullandığından, tespiti anomali analizine dayanır.

- **Sysmon ve Windows Olay Günlükleri Analizi:**
  - **Olay Kimliği (Event ID) 4624:** Başarılı ağ oturum açma olaylarını kaydeder. Olağandışı kaynak IP'lerden veya normalde o hesaba ait olmayan ana bilgisayarlardan gelen oturum açma girişimleri yanal hareketin bir göstergesi olabilir.
  - **Olay Kimliği (Event ID) 4688:** Yeni bir sürecin (process) oluşturulduğu zaman kaydedilir. Saldırganların kullandığı `psexec` veya `wmiexec` gibi araçların süreç yaratma olaylarını takip etmek önemlidir.
  - **Olay Kimliği (Event ID) 5140:** Bir ağ paylaşımına erişim sağlandığında tetiklenir. Özellikle hassas paylaşımlara (örneğin, yönetici paylaşımı) yapılan olağan dışı erişimleri izlemek hayati öneme sahiptir.

Pratik Senaryo ve Sorgu: Bir saldırganın, yetki yükseltme elde ettikten sonra, `psexec` kullanarak ağda yanal hareket etmeye çalıştığını varsayalım. Bu, genellikle standart bir hizmet olarak görünse de, yanal hareketin bir göstergesidir.
Sorgu (Splunk/SIEM):

```
index =winlogs sourcetype ="WinEventLog:Security" EventCode =4688
process_name ="psexec.exe" | table _time, host, process_name, command_line
```

Bu sorgu, `psexec.exe` çalıştıran tüm süreçleri arar. Çıktı, hangi ana bilgisayarda, ne zaman ve hangi komutlarla çalıştırıldığını göstererek, saldırganın yanal hareketini haritalamanıza olanak tanır. Bir saldırının tespiti, tek bir olaydan ziyade, bir dizi anormal olayın korelasyonuna dayanır. Örneğin, farklı makinelere art arda yapılan oturum açma girişimleri ve yetki yükseltme denemeleri, bir araya geldiğinde yüksek riskli bir kalıbı ortaya çıkarır.

8.4.3 Living-off-the-Land Techniques Identification

Living-off-the-Land (LotL) saldırıları, saldırganların hedef sistemdeki meşru araç ve işlevleri (LOLBins - Living-Off-the-Land Binaries) kötü niyetli amaçlar için kullanmasıdır. Bu saldırılar, zararlı bir dosya indirmek yerine, sistemde zaten var olan `PowerShell.exe` veya `wmic.exe` gibi yerleşik araçları kullanır, bu da geleneksel antivirüs ve imza tabanlı sistemleri atlatmada son derece etkili olmalarını sağlar.

- **Yaygın LOLBins ve LOLScripts (Kötüye Kullanılan Meşru Araçlar):**
  - **PowerShell:** Windows'un güçlü bir komut satırı ve betik dilidir. Saldırganlar, kötü amaçlı kod indirmek, yetki yükseltmek ve ağda yanal hareket etmek için kullanır.
  - **WMI (Windows Management Instrumentation):** Sistem yönetim aracıdır. Saldırganlar, uzaktan komut çalıştırmak, bilgi toplamak ve kalıcılık sağlamak için kullanır.
  - **Certutil:** Dosyaları indirmek ve kodlamak için kullanılır. Saldırganlar, kötü amaçlı yazılımları indirmek veya sızdırılan verileri (exfiltrated data) kodlamak için kullanabilir.

Tespit ve Örnek (PowerShell Analizi): Dosyasız (fileless) bir saldırıda, kötü amaçlı kod bellekte çalıştırıldığı için fiziksel bir dosya bırakmaz. Bu tür bir saldırıyı tespit etmek için PowerShell betik blok günlüğünü (Script Block Logging) etkinleştirmek ve bu günlükleri SIEM platformunda analiz etmek esastır.
SIEM Kuralı Örneği:

```
index =winlogs sourcetype ="WinEventLog:Microsoft-Windows-PowerShell/Operational"
AND (host.name =target_server AND
(PowerShell_Script_Content ="*IEX*" OR PowerShell_Script_Content ="*Invoke-Expression*"))
```

Bu kural, `Invoke-Expression` veya `IEX` gibi, uzaktan indirilen kodları bellekte çalıştırmak için sıkça kullanılan PowerShell komutlarını arar. Bu, bir saldırganın savunma mekanizmalarını atlatmaya çalıştığını gösterebilir. LotL teknikleri, saldırı ve savunma arasındaki "silahlanma yarışı"nı yeni bir seviyeye taşır. Savunucular, bu meşru araçların anormal bir şey yapıp yapmadığını belirlemek zorundadır.

8.4.4 Command and Control (C2) Communication Analysis

Komuta ve Kontrol (C2), bir saldırganın ele geçirdiği sistemlerle uzaktan iletişim kurarak komut göndermesi ve veri alması sürecidir. Modern C2 iletişimleri, tespitten kaçınmak için meşru trafik içinde gizlenir ve genellikle şifreleme, protokol taklidi (HTTP, HTTPS), bulut hizmetleri veya DNS tünelleme gibi teknikleri kullanır.

DNS Tünelleme Analizi (Örnek Senaryo): DNS tünelleme, DNS istekleri ve yanıtları içinde kötü amaçlı verileri gizleme tekniğidir. Güvenlik duvarları genellikle DNS trafiğine güvendiği için etkili bir yöntemdir.

Adım-Adım İşleyiş:

1. **Saldırgan Alan Adını Kaydeder:** Saldırgan, `kotusite.com` gibi bir alan adı kaydeder ve bu alan adını kendi kontrolündeki bir DNS sunucusuna yönlendirir.

2. **Veriyi Kodlar ve DNS Sorgusu Gönderir:** Kurban makinesindeki kötü amaçlı yazılım, sızdırılacak veriyi (örneğin, çalınan kimlik bilgileri) bir alt alan adına (subdomain) kodlar. Örnek: `gizli.veriler.kotusite.com`.

3. **Sorgu Saldırganın Sunucusuna Ulaşır:** Bu istek, standart DNS çözücülerinden geçer ve sonunda saldırganın sunucusuna ulaşır.

4. **Yanıt Verisi Gönderir:** Saldırganın sunucusu, komutları kurban makinesine geri göndermek için DNS yanıtını kullanır.

Tespit Teknikleri:

- **Büyük Sorgu/Yanıt Boyutları:** Olağan dışı büyük boyutlar, veri sızdırmanın bir göstergesi olabilir.

- **Olağandışı DNS Kayıt Türleri:** Normalde kullanılmayan `TXT` veya `NULL` gibi kayıt türlerinin kötüye kullanımı.

- **Anormal İstek Sıklığı:** Bir ana bilgisayardan belirli bir alana çok sayıda, düzenli aralıklarla yapılan istekler.

Modern C2 teknikleri, geleneksel ağ savunmalarını atlatmak için "güvenilir" trafik türlerini (DNS, HTTPS) kullanır. Bu, ağ güvenliğinin sadece trafiği engellemekle kalmayıp, aynı zamanda trafiğin içeriğini ve davranışını da analiz etmesi gerektiğini vurgular.

8.4.5 APT Attribution ve Threat Group Tracking

Bir APT saldırısını belirli bir tehdit grubuna atfetmek, nadiren mümkün olan karmaşık ve kaynak yoğun bir süreçtir.

Atıf Süreci:

- **Gözlemlenebilir Kanıtları Analiz Etme:** Süreç, saldırganın altyapısını, kurbanlarını ve TTP'lerini analiz ederek başlar.

- **Uzman Yorumu ve Güven Seviyeleri:** Analistler, ellerindeki verilere dayanarak bir atıf değerlendirmesi yapar ve bu değerlendirmeye bir güven seviyesi (örneğin, düşük, orta, yüksek) atarlar.

- **Çelişen Atıfları Yönetme:** Farklı güvenlik firmalarının aynı saldırıya farklı isimler vermesi, tutarlı bir atıf standardı oluşturmayı zorlaştırır.

Atıf, bir "evet/hayır" sorusu değildir, daha çok bir olasılık değerlendirmesi ve dinamik bir süreçtir. Saldırganlar taktiklerini değiştirdikçe veya yeni kanıtlar ortaya çıktıkça, bir atıf değerlendirmesi sürekli olarak yeniden gözden geçirilmelidir.

APT Atıf Zorlukları Tablosu

Zorluk Türü	Açıklama	Pratik Örnek
Teknik Zorluklar	Saldırganların izlerini gizlemek için kullandığı yöntemler.	Saldırganların saldırı için botnetler, kiralanmış altyapı veya farklı vekil sunucular kullanması.
Organizasyonel Zorluklar	Güvenlik firmaları arasındaki atıf isimlendirme tutarsızlıkları.	Aynı saldırıya farklı firmaların APT34 (PWC) veya Helix Kitten (Mandiant) gibi farklı isimler vermesi.
Operasyonel Zorluklar	TTP'lerin veya araçların farklı gruplar arasında paylaşılması.	Bir grubun belirli bir arka kapı veya exploit kodunu başka bir gruba satması veya takas etmesi.

8.5 Malware Analysis ve Reverse Engineering

Kötü amaçlı yazılım analizi (malware analysis), bir kötü amaçlı yazılım örneğinin işlevselliğini, kökenini ve potansiyel etkisini anlamak için yapılan bir incelemedir. Tersine mühendislik (reverse engineering) ise, bir yazılımın veya sistemin nasıl çalıştığını anlamak için onu parçalarına ayırma sürecidir. Bu iki disiplin, siber güvenlik uzmanlarının yeni tehditleri anlamasına, savunma mekanizmaları geliştirmesine ve saldırıların arkasındaki aktörleri belirlemesine yardımcı olur.

8.5.1 Static Malware Analysis Techniques ve Tools

Statik analiz, kötü amaçlı yazılımı çalıştırmadan, kodunu, yapısını ve içindeki verileri incelemeyi içerir. Bu yöntem, malware hakkında ilk bilgileri hızlı bir şekilde elde etmek için kullanılır.

Teknikler:

- **Dosya Parmak İzi (File Fingerprinting):** Dosyanın MD5 veya SHA256 gibi kriptografik bir karmasını (hash) oluşturur ve bu karmayı bilinen kötü amaçlı yazılım veritabanlarında sorgular.
- **Dizge (String) Analizi:** Bir ikili dosyadan okunabilir dizgeleri (IP adresleri, dosya yolları, komutlar) ayıklamak, saldırganın niyetine dair ipuçları verebilir.
- **İçe Aktarılan Fonksiyonları (Imports) İnceleme:** Dosyanın hangi Windows API'lerini çağırdığını kontrol etme, bu sayede dosyanın potansiyel işlevleri hakkında fikir edinme. Örneğin, `CreateRemoteThread` işlevi süreç enjeksiyonunu, `URLDownloadToFile` internetten dosya indirmeyi gösterebilir.
- **Decompiler ve Disassembler Kullanımı:** `Ghidra` gibi araçlar, makine kodunu daha okunabilir assembly diline veya C diline çevirerek analistlerin kodun işleyişini anlamasına yardımcı olur.

Araçlar ve Pratik Kılavuz (Ghidra):

- **Ghidra:** NSA tarafından geliştirilen ücretsiz bir tersine mühendislik aracıdır.
- **Adım-Adım Kullanım:**
  1. **Kurulum:** Java'nın kurulu olduğundan emin olun ve Ghidra'yı GitHub sayfasından indirin.
  2. **Proje Oluşturma:** Yeni bir proje oluşturun ve kötü amaçlı yazılım dosyasını içe aktarın.
  3. **Analiz:** Ghidra'nın dosyayı analiz etmesine izin verin.
  4. **İnceleme:** `İçe Aktarılanlar` (Imports) penceresinden API çağrılarını inceleyin. `Dizge` (Strings) penceresinden okunabilir dizgeleri arayın. `İşlev Grafiği` (Function Graph) ile kodun akışını görselleştirin.

Statik analiz, kötü amaçlı yazılımın potansiyel işlevleri hakkında hızlı bir genel bakış sağlar ancak, gizleme (obfuscation) ve paketleme (packing) teknikleri nedeniyle her zaman tam bir resim sunamaz.

Statik Analiz Araçları ve Kullanım Alanları

Araç Adı	Tipi	Kullanım Alanı
Ghidra	Tersine Mühendislik	Kod analizi, disassembly, decompilation, içe aktarılan fonksiyonların incelenmesi.
strings	Komut Satırı Aracı	Bir ikili dosyadan okunabilir ASCII ve Unicode dizgeleri ayıklama.
md5sum/sha256sum	Komut Satırı Aracı	Bir dosyanın kriptografik karmasını (hash) oluşturma ve bilinen veritabanlarında sorgulama.
PEStudio	Otomatik Statik Analiz	Bir PE (Portable Executable) dosyasının başlıklarını, içe aktarılanlarını ve dizgelerini hızlıca inceleme.

8.5.2 Dynamic Analysis ve Sandbox Environments

Dinamik analiz, kötü amaçlı yazılımı kontrollü ve izole bir ortamda (sandbox) çalıştırarak gerçek zamanlı davranışını gözlemleme sürecidir. Bu yöntem, statik analizin atlatıldığı durumlarda kritik öneme sahiptir.

- **Sandbox:** Bir kötü amaçlı yazılımı ana sistemden tamamen izole edilmiş bir sanal makinede (VM) veya konteynerde çalıştıran güvenli bir ortamdır. Amacı, kötü amaçlı yazılımın sistemde kalıcı değişiklikler yapmasını veya ağa yayılmasını engellemektir.
- **Cuckoo Sandbox (Pratik Kılavuz):** Cuckoo, açık kaynaklı, otomatik bir malware analiz platformudur.
  1. **Yapılandırma:** Cuckoo, bir ana makine (host) ve bir veya daha fazla sanal makine (guest) içerir. Sanal makineler, kötü amaçlı yazılımı çalıştırmak için kullanılır ve ağları ana makineden izole edilir.
  2. **İşleyiş:** Analist, şüpheli bir dosyayı Cuckoo'ya gönderir. Platform, bu dosyayı sanal makinede çalıştırır ve şunları izler:
     – Dosya sistemi değişiklikleri (yeni dosya oluşturma, silme).
     – Kayıt defteri (registry) değişiklikleri.
     – Ağ trafiği (C2 sunucularıyla iletişim, DNS istekleri).
     – Süreç enjeksiyonu ve yetki yükseltme girişimleri.
  3. **Raporlama:** Analiz tamamlandığında, Cuckoo, gözlemlenen davranışları ve toplanan IOC'leri (IP'ler, alan adları, dosya karmaları) içeren kapsamlı bir rapor oluşturur.

Dinamik analiz, imza tabanlı sistemlerin kaçırdığı sıfır gün (zero-day) tehditlerini ve polimorfik (polymorphic) kötü amaçlı yazılımları (imzalarını değiştiren) tespit etmede etkilidir, çünkü bir dosyanın içeriğinden ziyade yaptığı şeye odaklanır.

8.5.3 Behavioral Analysis ve Family Classification

Davranışsal analiz, kötü amaçlı yazılımları, gösterdikleri davranışsal kalıplara göre sınıflandırmanın temelidir. Bu sınıflandırma, yeni varyantlar ortaya çıktığında bile benzer davranışları paylaşan kötü amaçlı yazılımları bir araya getirmeyi sağlar.

Sınıflandırma Süreci:

1. **Davranış İzleme:** Kötü amaçlı yazılım örneği, bir sandbox ortamında çalıştırılırken, yaptığı tüm eylemler (API çağrıları, dosya değişiklikleri, ağ bağlantıları) kaydedilir.
2. **Özellik Çıkarma:** Bu eylemlerden, dosya açma, mutex'leri kilitleme veya belirli kayıt defteri anahtarlarını ayarlama gibi davranışsal özellikler çıkarılır.

3. **Makine Öğrenimi ile Kümeleme:** Makine öğrenimi algoritmaları, benzer davranışsal kalıpları paylaşan kötü amaçlı yazılımları aynı aileye gruplar. Bu kümeleme, bilinen bir ailenin yeni varyantlarını tanımlamaya yardımcı olur.

Bir kötü amaçlı yazılımın hangi aileye ait olduğunu anlamak, onun potansiyelini (ne yapabileceğini), hedeflerini ve kökenini anlamamıza yardımcı olur ve tehdit aktörü atfı için önemli bir köprü görevi görür.

8.5.4 Anti-analysis Evasion Techniques

Kötü amaçlı yazılım yazarları, analizden kaçınmak ve tespit edilme sürelerini uzatmak için çeşitli teknikler kullanır. Bu teknikler, statik ve dinamik analiz araçlarını atlatmayı amaçlar.

Kaçınma (Evasion) Teknikleri:

- **Sandbox Tespiti:** Kötü amaçlı yazılım, bir sanal ortamda (VM) çalışıp çalışmadığını kontrol eder. Örneğin, belirli sanallaştırma sürücülerinin varlığını kontrol edebilir veya sistem özelliklerini (bellek boyutu, ekran çözünürlüğü) test edebilir. Bir sandbox tespit edilirse, kötü amaçlı yazılım zararsız bir şekilde sonlanabilir veya zararsız bir davranış sergileyebilir.
- **Zamana Bağlı Gecikme:** Kötü amaçlı yazılım, kötü niyetli faaliyetlerini, ortamın bir insan kullanıcısının etkileşimiyle geçen bir zaman dilimi içinde olduğunu teyit edene kadar geciktirir.
- **Statik Analizden Kaçınma:** Kod karmaşıklaştırma (obfuscation), paketleme (packing) veya sahte komutlarla disassembler'ları yanıltma.

Karşı Önlemler:

- **Gelişmiş Sandbox'lar:** Sanal ortamların gerçek sistemler gibi görünmesini sağlayarak sandbox tespit tekniklerini atlatır.
- **Davranışsal Analiz:** Kötü amaçlı yazılımın statik imzasını değil, davranışını analiz eder ve zararsız görünen ancak aniden kendini sonlandıran bir dosyanın davranışını kaydeder.

Saldırgan ve savunma arasındaki bu "silahlanma yarışı" sürekli evrilir. Saldırganlar yeni kaçınma yolları buldukça, güvenlik firmaları da bu kaçınma yöntemlerini tespit etmek için yeni karşı önlemler geliştirmek zorundadır.

8.5.5 Automated Malware Analysis ve YARA Rule Development

Otomatik kötü amaçlı yazılım analizi, büyük hacimli şüpheli dosyaları hızlı bir şekilde işlemek için sandbox'ları ve makine öğrenimini kullanır. YARA kuralları, bu otomasyonun ve tehdit avcılığının temel bir bileşenidir.

YARA Kuralları:

- **Tanım:** Kötü amaçlı yazılım ailelerini veya belirli zararlı dosyaları metinsel veya ikili kalıplara göre tanımlamak için kullanılan, imza benzeri kurallardır.
- **Yapı:** Her kuralın bir adı, bir dizi metinsel veya hexadecimal dize (strings) ve bir koşul (condition) ifadesi bulunur.

YARA Kuralı Geliştirme (Adım-Adım Örnek): Bir saldırganın, dosya şifreleme için bir parola içeren bir fidye yazılımı kullandığını varsayalım. Bu parola, YARA kuralı ile tespit edilebilir.

Adım-Adım Uygulama:

1. **Kuralı Tanımlama:** Kurala açıklayıcı bir ad verilir. `rule Ransomware_Encrypter_Password : ransomware`
2. **Dizgeleri Tanımlama:** Kötü amaçlı yazılıma özgü benzersiz metinsel veya ikili kalıplar belirlenir. Bu durumda, şifreleme parolası: `strings: $a = "s3cr3t_p4ssw0rd_3ncr_k3y"`
3. **Koşulu Belirleme:** Bu dizgenin dosya içinde bulunması gerektiğini tanımlayan koşul belirlenir. `condition: $a`

Tam YARA Kuralı Örneği:

```
rule Ransomware_Encrypter_Password : ransomware
{
    meta:
        description = "Detects a known ransomware variant by its hardcoded encryption key."
        author = "Cybersecurity Analyst"
        date = "2024-10-27"
        tlp = "amber"

    strings:
        $a = "s3cr3t_p4ssw0rd_3ncr_k3y"
        $b = { 6A 40 68 00 30 00 00 6A 14 8D 91 }

    condition:
        ($a or $b) and filesize < 1MB
}
```

YARA kuralları, otomatik analiz ve manuel tehdit avcılığı arasında bir köprü görevi görür. Bir tehdit avcısı, yaptığı araştırmada yeni ve bilinmeyen bir kötü amaçlı yazılım varyantı keşfettiğinde, bu kalıplara dayalı olarak bir YARA kuralı yazarak, manuel keşfini otomatik bir tespit kuralına dönüştürebilir.

8.6 Threat Intelligence Integration ve Operationalization

Tehdit istihbaratının operasyonelleştirilmesi, toplanan istihbaratın bir kuruluşun güvenlik süreçlerine ve teknolojilerine entegre edilerek, savunma yeteneklerini aktif olarak güçlendirmesi anlamına gelir. Bu, istihbaratın sadece bir bilgi yığını olmaktan çıkıp, eyleme geçirilebilir bir güvenlik varlığına dönüşmesini sağlar. Entegrasyon, SIEM, güvenlik duvarları, IDS/IPS ve EDR gibi güvenlik araçlarının, en son tehdit verileriyle beslenerek daha akıllı ve etkili hale getirilmesini içerir.

8.6.1 SIEM ve SOAR Platform Entegrasyonu

SIEM ve SOAR platformları, tehdit istihbaratını operasyonel hale getiren birbirini tamamlayıcı araçlardır.

- **SIEM (Security Information and Event Management):** Çeşitli kaynaklardan (sunucular, güvenlik duvarları, uygulamalar) günlük ve olay verilerini toplar, korele eder ve analiz ederek güvenlik olaylarını tespit eder.

- **SOAR (Security Orchestration, Automation, and Response):** Güvenlik olaylarına yönelik yanıtı otomatikleştiren ve düzenleyen bir hizmetler setidir.

Entegrasyon Senaryosu: Bir SIEM, yeni bir kötü amaçlı IP'den gelen bir oturum açma girişimi hakkında bir uyarı oluşturduğunda, bu uyarı otomatik olarak SOAR platformuna iletilir. SOAR, bu IP için bir "playbook"u tetikler. Bu playbook, IP'yi VirusTotal gibi kaynaklarda sorgular, bu IP'den gelen ağ trafiğini SIEM'de arar ve ilgili uç noktanın karantinaya alınması için EDR çözümüne komut gönderir. Bu işlemler, analiste zenginleştirilmiş verilerle birlikte bir bilet (ticket) atanmadan önce saniyeler içinde gerçekleşir.

SOAR, SIEM'den gelen "gürültü"yü azaltır ve analistlerin verimliliğini artırarak güvenlik operasyonları için bir "güç çarpanı" (force multiplier) görevi görür. Bu otomasyon, bir saldırganın ağ içinde yetki yükseltme veya veri sızdırma gibi hedeflerine ulaşmasını engellemek için hayati önem taşır.

SIEM ve SOAR Karşılaştırma Tablosu

Özellik	SIEM	SOAR
Fonksiyon	Veri toplama ve olay tespiti	Olay yanıtını otomasyon ve orkestrasyonu
Odak Noktası	Gelen verileri korelasyon yoluyla analiz etme	Güvenlik operasyonlarını otomatikleştirme
Giriş Verisi	Ham log ve olay verileri	İşlenmiş güvenlik verileri, SIEM uyarıları
Otomasyon Seviyesi	Veri toplama ve analizde sınırlı	Tam veya yarı otomasyon

8.6.2 Automated Threat Response ve Playbook Development

Otomatik tehdit yanıtı, bir güvenlik olayına önceden tanımlanmış bir dizi eylemi tetikleyen bir süreçtir. Playbook'lar, bu eylemleri tanımlayan kılavuzlardır ve bir kuruluşun olay müdahale planının operasyonel bileşenini oluşturur.

Playbook'un Ana Bileşenleri:

1. **Tetikleme Koşulu (Initiating Condition):** Playbook'u neyin tetiklediğini belirler (örneğin, yüksek riskli bir uyarı, kullanıcı raporu).
2. **Süreç Adımları (Process Steps):** Olayı çözmek için atılacak tüm teknik ve teknik olmayan adımları içerir.
3. **Son Durum (End State):** Playbook'un istenen sonucunu (örneğin, olay çözüldü, hafifletildi, başka bir ekibe aktarıldı) tanımlar.

Playbook Geliştirme (Adım-Adım Kılavuz):

1. **Amacı ve Kapsamı Tanımlama:** Hangi tür olaylara (örneğin, kimlik avı, kötü amaçlı yazılım) yanıt vereceğini belirleyin.
2. **Olası Eylemleri ve Bağımlılıkları Belirleme:** Tüm olası teknik eylemleri (IP adresini engelle, uç noktayı karantinaya al) ve iletişim adımlarını (yöneticileri uyar) listeleyin.
3. **Temel İş Akışını Oluşturma:** Yalnızca kritik ve zorunlu adımları içeren temel bir iş akışı oluşturun.
4. **Otomasyonu Entegre Etme:** Manuel görevleri otomatikleştirmek için SOAR'ı kullanın.
5. **Son Durumları ve Yükseltmeleri Tanımlama:** Playbook'un ne zaman sona ereceğini ve ne zaman manuel müdahale için bir analiste yükseltilmesi gerektiğini belirleyin.

Otomasyon ve playbook'lar, bir kuruluşun siber güvenlik kapasitesini artırır ve analistlerin daha azıyla daha fazlasını yapmasını sağlar. Otomatikleştirilmiş playbook'lar, olaylara reaksiyon süresini saatlerden saniyelere düşürebilir, bu da bir saldırganın hedeflerine ulaşmasını engellemek için kritik öneme sahiptir.

8.6.3 Intelligence Sharing Communities ve Trust Groups

Bilgi paylaşımı, tehdit istihbaratı programlarının temel taşıdır. Kuruluşlar, tehdit verilerini paylaşarak kolektif savunma yeteneklerini güçlendirirler. Bu, bir saldırıdan elde edilen bilgilerin daha geniş bir topluluk tarafından kullanılmasına olanak tanır, böylece tehditler daha hızlı tespit edilebilir ve savunmalar güçlendirilebilir.

Paylaşım Türleri:

- **Tek Yönlü (Unidirectional):** Bir sağlayıcının abonelere bilgi aktardığı modeldir.
- **Çift Yönlü (Bidirectional):** Tarafların aktif olarak hem katkıda bulunduğu hem de bilgi aldığı modeldir.

Topluluklar:

- **Bilgi Paylaşım ve Analiz Merkezleri (ISAC'ler):** Belirli sektörlerdeki kuruluşlar için oluşturulmuş, endüstri odaklı paylaşım platformlarıdır.

Avantajlar:

- **Daha Hızlı Tehdit Tespiti:** Tehdit beslemeleri, bir organizasyonun kendi başına fark edemeyeceği tehditler hakkında erken uyarı sağlar.

- **Gelişmiş Savunma:** En son tehditler, zafiyetler ve saldırı vektörleri hakkında bilgi paylaşımı, organizasyonların savunmalarını proaktif olarak güçlendirmesine yardımcı olur.

Bilgi paylaşımının başarısı, paylaşılan verinin kalitesine, güvene ve gizliliğe bağlıdır. Kuruluşlar, itibar veya sorumluluk endişeleri nedeniyle güvenlik olaylarını ifşa etmekte tereddüt edebilirler. Ancak, etkin bir bilgi paylaşım topluluğu, üyelerin hassas bilgileri güvenli bir şekilde paylaşabilmesi için güçlü protokoller (şifreleme, anonimleştirme) ve karşılıklı güvene dayalı bir ortam gerektirir.

8.6.4 Custom Threat Intelligence Development

Özel tehdit istihbaratı, organizasyonun kendi benzersiz risk profiline, varlıklarına ve tehdit ortamına göre geliştirilen istihbarattır. Ticari beslemeler, genel bir tehdit görünümü sağlasa da, özel istihbarat, kaynakların en kritik ve kişiselleştirilmiş risklere odaklanmasını sağlar.

Geliştirme Adımları:

1. **"Crown Jewels" (En Değerli Varlıklar) Analizi:** Saldırganların en çok hedefleyeceği kritik sistemleri ve verileri belirleyin.

2. **Özel PIR'ler Oluşturma:** Kuruluşun benzersiz ihtiyaçlarına odaklanan sorular geliştirin.

3. **Dahili Veri Madenciliği:** İç logları ve telemetri verilerini analiz ederek, harici beslemelerde bulunmayan tehditleri (örneğin, içeriden gelen tehditler veya bilinmeyen kötü amaçlı yazılım varyantları) bulun.

4. **Dış Kaynaklarla Zenginleştirme:** Dahili bulguları, ticari beslemeler ve OSINT ile zenginleştirin.

Özel tehdit istihbaratı, genel ticari beslemelerin ötesine geçerek bir kuruluşun savunma stratejisini saldırganın niyetleriyle hizalar. Bu, kaynakların en kritik risklere odaklanmasını sağlar ve ticari bir beslemeden elde edilemeyecek derinlikte içgörüler sunar.

8.6.5 Threat Intelligence ROI Measurement ve Effectiveness

Bir tehdit istihbaratı programının değerini kanıtlamak için, somut iş sonuçlarıyla ilişkilendiren metrikler kullanılmalıdır. Üst yönetim, güvenlik yatırımları için somut kanıtlar ister.

ROI Ölçüm Metrikleri:

- **Azaltılan Ortalama Tespit Süresi (MTTD) ve Azaltılan Ortalama Yanıt Süresi (MTTR):** Tehdit istihbaratının, saldırıları daha hızlı tespit etmeye ve müdahale etmeye ne kadar yardımcı olduğunu gösterir.

- **Yanlış Pozitif Oranı:** Tehdit istihbaratından gelen uyarıların ne kadarının sahte olduğunu gösterir. Düşük oran, daha verimli bir SOC anlamına gelir.

- **Önlenen Olay Sayısı:** Tehdit istihbaratıyla proaktif olarak durdurulan kimlik avı, fidye yazılımı veya diğer saldırıların sayısı.

Tehdit istihbaratı, güvenlikteki bir "bileşen"den ziyade, finansal riski ve iş sürekliliğini doğrudan etkileyen stratejik bir iş yatırımıdır. MTTD ve MTTR gibi metrikler, tehdit istihbaratının bir ihlal maliyetini nasıl düşürdüğünü doğrudan gösterir, bu da güvenlik ekiplerinin bütçe taleplerini destekler ve siber güvenliğin iş hedefleriyle uyumunu sağlar.

Tehdit İstihbaratının ROI Metrikleri Tablosu

Metrik	Tanım	İş Değerine Katkısı
MTTD Azaltma	Bir tehdidi tespit etmek için gereken süredeki düşüş.	Saldırganın ağda kalma süresini ve potansiyel hasarı azaltır.
MTTR Azaltma	Bir tehdidin çözümlenmesi için gereken süredeki düşüş.	Olay müdahale maliyetlerini düşürür ve iş sürekliliğini sağlar.
Yanlış Pozitif Azaltma	Hatalı güvenlik uyarılarının azalması.	Analist yorgunluğunu önler ve operasyonel verimliliği artırır.
Önlenen Olay Sayısı	Tehdit istihbaratı ile proaktif olarak engellenen saldırı sayısı.	İhlalden kaynaklanan maliyetleri ve itibar kaybını önler.

Bölüm 9

OLAY MÜDAHALE VE ADLİ BİLİŞİM

Giriş

Olay müdahale ve adli bilişim (DFIR - Digital Forensics and Incident Response), siber güvenlik olaylarının tespiti, analizi, müdahalesi ve sonrasında delil toplama süreçlerini kapsayan kritik bir disiplindir. Bu bölümde, modern DFIR metodolojileri, araçları ve teknikleri detaylı olarak ele alınacaktır.

9.1 Olay Müdahale Temelleri ve Metodolojiler

9.1.1 Gelişmiş Saldırı Göstergeleri ve Tehdit Avlama

9.1.2 DFIR Metodolojileri: SANS ve NIST Karşılaştırması

Olay müdahale ve dijital adli bilişim (DFIR) alanında en yaygın kullanılan iki metodoloji, SANS ve NIST'in geliştirdiği çerçevelerdir. Bu metodolojiler, organizasyonların siber olaylara karşı hazırlıklı olmasını ve etkili müdahale etmesini sağlar.

SANS DFIR Metodolojisi

SANS, dünya genelinde en çok kullanılan altı adımlı Olay Müdahale (Incident Response) sürecini önerir:

1. **Hazırlık (Preparation):**
    - Politika, olay müdahale planı ve araçların oluşturulması
    - İletişim kanallarının belirlenmesi
    - Personel eğitimlerinin tamamlanması
2. **Tespit (Identification):**
    - Olası güvenlik olaylarının belirlenmesi
    - Uyarılar ve logların analizi
    - Tehdit istihbaratı ve anomali tespiti
3. **Sınırlama (Containment):**
    - Kısa vadeli: Etkilenen sistemlerin izolasyonu
    - Uzun vadeli: Yamalar, güvenlik duvarı kuralları ve ağ segmentasyonu
4. **Ortadan Kaldırma (Eradication):**
    - Zararlı yazılımın temizlenmesi

- Açıklıkların kapatılması
- Saldırganın kalıcılığının yok edilmesi

5. **Kurtarma (Recovery):**
    - Sistemlerin geri yüklenmesi
    - Bütünlüğün doğrulanması
    - Normal işleyişe dönüş
6. **Dersler (Lessons Learned):**
    - Olay sonrası değerlendirme ve raporlama
    - Kök neden analizi
    - Savunma mekanizmalarının geliştirilmesi

NIST DFIR Metodolojisi

NIST SP 800-61 (Bilgisayar Güvenliği Olay Müdahale Rehberi) dokümanında tanımlanan süreç dört ana aşamadan oluşur:

1. **Hazırlık (Preparation):**
    - Politika belirleme ve personel eğitimi
    - İletişim kanallarının tanımlanması
    - Tespit araçlarının kurulması
2. **Tespit ve Analiz (Detection & Analysis):**
    - Güvenlik uyarıları ve logların izlenmesi
    - Olayın kapsamı ve türünün belirlenmesi
    - Ciddiyet sınıflandırması ve önceliklendirme
3. **Sınırlama, Ortadan Kaldırma ve Kurtarma:**
    - Olayın etkisinin durdurulması
    - Kök nedenin ortadan kaldırılması
    - Sistemlerin geri yüklenmesi ve izlenmesi
4. **Olay Sonrası Faaliyetler:**
    - Alınan derslerin dokümantasyonu
    - Olay detaylarının belgelenmesi
    - IR planı ve güvenlik mimarisinin güncellenmesi

SANS ve NIST Karşılaştırması

Kurumlar genellikle bu iki metodolojinin birleşimini kullanır:

- NIST → Politika ve standartlar için
- SANS → Operasyonel uygulama için
- **Tehdit Avlama Metodolojileri:**

Kriter	SANS	NIST
Köken	Eğitim & sertifikasyon kurumu	ABD federal standart kurumu
Odak	Pratik, uygulayıcı odaklı	Stratejik, politika & uyumluluk odaklı
DFIR Modeli	6 adım	4 adım
Güçlü Yanı	Operasyonel, sahada uygulanabilir	Standartlaştırılmış, politika uyumlu
Kullanım	SOC/IR ekipleri	ABD kamu sektörü & uyumluluk

Tablo 9.1: SANS ve NIST DFIR Metodolojileri Karşılaştırması

  - **Hipotez Bazlı Avlama:** Örnek: ”Sistemde PowerShell Empire kullanıldığına dair göstergeler var mı?”
  - **İstihbarat Bazlı Avlama:** Bilinen APT gruplarların TTPs'lerine dayalı arama
  - **Anomali Bazlı Avlama:** Normal sistem davranışından sapmaların tespiti

- **Gelişmiş IOC'ler:**

  - **YARA Kuralları:** Zararlı yazılım ailelerini tespit etmek için özel imza tanımları
  - **SIGMA Kuralları:** SIEM sistemleri için standartlaştırılmış log analiz kuralları
  - **Snort/Suricata Kuralları:** Ağ trafiğinde zararlı aktivite tespiti için kullanılan kurallar

- **Hazırlık (Preparation):** Olayların meydana gelme olasılığını azaltmak için proaktif önlemlerin alındığı aşamadır. Bu aşama, güvenlik açıklarının belirlenmesini ve azaltılmasını, güvenlik politikaları ile prosedürlerinin tanımlanmasını ve bilişim varlıklarının risk analizine göre önceliklendirilmesini içerir. Etkili bir hazırlık için, bir Bilgisayar Güvenlik Olay Müdahale Ekibinin (CSIRT) oluşturulması, iç ve dış paydaşlarla iletişim kanallarının kurulması ve düzenli masa başı (tabletop) tatbikatlarının yapılması hayati önem taşır.

- **Pratik Senaryo - Hazırlık Aşaması Tatbikatı:**

  - Bir fidye yazılımı saldırısı için masa başı tatbikatı ele alınabilir. Senaryoda, bir çalışanın e-posta yoluyla fidye yazılımını sisteme bulaştırdığı varsayılır. Olay müdahale ekibi, bu simülasyon sırasında rolleri ve sorumlulukları üzerinden aşağıdaki adımları tartışır:
    * Hangi varlıkların (sunucular, veritabanları) öncelikli olarak izole edilmesi gerektiği
    * İş sürekliliğini sağlamak için hangi alternatif sistemlerin devreye alınacağı
    * Üst yönetim, hukuk departmanı ve halkla ilişkiler ile hangi iletişim kanallarının kullanılacağı
    * Verilerin geri yüklenmesi için yedekleme altyapısının test edilmesi ve doğrulanması

  - **Tespit ve Analiz (Detection & Analysis):** Bu aşama, güvenlik olaylarının sürekli izlenmesini ve toplanan verilerin analiz edilmesini içerir. Amaç, şüpheli etkinlikleri gerçek bir siber olaydan ayırt etmektir. Olayın doğası, kapsamı, kökeni ve potansiyel etkisi bu aşamada belirlenir.
  - **Sınırlama, Yok Etme ve Kurtarma (Containment, Eradication & Recovery):** Tespit ve analiz aşamasının ardından, saldırının yayılmasını durdurmak ve zararı en aza indirmek için harekete geçilir. Bu, tehdidin sistemden tamamen temizlenmesini ve iş operasyonlarının normal durumuna geri döndürülmesini kapsar.

9.1.3 MITRE ATT&CK Çerçevesi ve Saldırı Tespit Stratejileri

MITRE ATT&CK, saldırganların kullandığı taktik ve teknikleri kategorize eden kapsamlı bir bilgi tabanıdır. Bu çerçeve, olay müdahale ve adli bilişim süreçlerinde saldırıların analizi ve savunma stratejilerinin geliştirilmesi için kullanılır.

– **Olay Sonrası Faaliyetler (Post-Incident Activity):** Olay müdahalesinin tamamlanmasından sonra gerçekleştirilen bu aşama, yaşanan olaydan dersler çıkarmayı ve elde edilen bilgileri gelecekteki güvenlik tedbirlerini güçlendirmek için kullanmayı hedefler. Bu süreç, organizasyonun savunma duruşunu sürekli olarak iyileştirme amacını taşır.

9.1.4 ATT&CK Matrisi Kullanımı

– **Taktikler ve Teknikler:**
  * **Başlangıç Erişimi (Initial Access):** Oltalama e-postaları, güvenlik açıklarının istismarı
  * **Yetki Yükseltme (Privilege Escalation):** Sistem zafiyetleri, token çalma
  * **Yanal Hareket (Lateral Movement):** Pass-the-hash, uzak masaüstü protokolü
  * **Veri Hırsızlığı (Exfiltration):** Alternatif protokoller, steganografi

9.1.5 Bilgisayar Güvenlik Olay Müdahale Ekibi (CSIRT) Yapısı

CSIRT, bir organizasyondaki siber güvenlik olaylarını yönetmekle görevli, disiplinler arası ve özel bir ekiptir. Bu ekibin temel amacı, hasarı en aza indirmek ve iş sürekliliğini sağlamaktır. Etkili bir CSIRT operasyonu, sadece teknik uzmanlık değil, aynı zamanda idari ve yasal yetkinlikler de gerektirir.

* **Roller ve Sorumluluklar:** CSIRT, olay yönetimi sürecinin tüm yaşam döngüsünü (hazırlık, tespit, analiz, sınırlama, yok etme, kurtarma ve olay sonrası inceleme) yönetir. Ekip üyeleri, altyapı, işletim sistemleri ve uygulamalar hakkında derinlemesine bilgiye sahip olmalı; etik hacking, bulut güvenliği ve log analizi gibi alanlarda uzmanlık sergilemelidir.
* **Çapraz Fonksiyonel Yapı:** Bir CSIRT, yalnızca teknik uzmanlardan oluşmamalıdır. Hukuk, iletişim, insan kaynakları ve kritik iş birimi yöneticileri gibi farklı departmanlardan temsilcileri içermelidir. Bu çok disiplinli yapı, bir siber saldırının teknik boyutlarının yanı sıra yasal ve itibari sonuçlarının da bütüncül bir yaklaşımla ele alınmasını sağlar. Yönetimden destek ve onay almak, bu ekibin etkin bir şekilde çalışabilmesi için başlangıçta atılması gereken en önemli adımlardan biridir.

9.1.6 Zararlı Yazılım Analiz Teknikleri

Modern zararlı yazılımlar, tespit edilmemek için gelişmiş teknikler kullanır. Bu yazılımları analiz etmek için hem statik hem de dinamik analiz teknikleri kullanılır.

* **Statik Analiz:**
  · Dosya imzaları ve hash değerleri kontrolü
  · PE başlık analizi ve bağımlılık incelemesi
  · Dizi analizi ve şüpheli API çağrılarının tespiti
  · Yürütülebilir dosya yapısının incelenmesi

9.1.7 Olay Sınıflandırma, Önceliklendirme ve Ciddiyet Değerlendirmesi

Olay müdahale sürecinin ilk adımlarından biri, bir olayın ciddiyetini ve potansiyel etkisini değerlendiren ”triyaj” sürecidir. Bu aşama, kısıtlı kaynakların en kritik olaylara yönlendirilmesini sağlar ve yanlış pozitif alarmların filtrelenmesine yardımcı olur. Olaylar, saldırı türlerine göre farklı öncelik seviyelerine ayrılır. Örneğin:

* **Keşif ve Araştırma (Probe):** Saldırganın bilgi toplamaya çalıştığı düşük öncelikli olaylar.
* **İstismar ve Kurulum (Exploit & Installation):** Bir zafiyetin sömürülmesi veya kötü amaçlı yazılımın sisteme yerleştirilmesi girişimi. Bu orta/yüksek öncelikli bir olaydır.
* **Sistemin Ele Geçirilmesi (System Compromise):** Saldırganın sisteme tam erişim sağladığı, en yüksek öncelikli olaydır.

Pratik Senaryo - Alarm Triyajı:

Bir SIEM sistemi, aynı IP adresinden 30 saniye içinde 6 başarısız oturum açma girişimi için bir alarm üretir. Bir siber güvenlik analisti bu durumu şu adımlarla yönetir:

1. **Gözden Geçirme:** Analist, alarmı inceler ve benzer olayların geçmişte yaşanıp yaşanmadığını, bu IP adresinin bilinen bir zafiyet tarayıcısına ait olup olmadığını kontrol eder.
2. **Sınıflandırma:** Etkinlik, ”Brute Force Saldırı Girişimi” olarak sınıflandırılır.
3. **Öncelik Belirleme:** Bu bir girişimi temsil ettiği ve henüz sisteme sızma gerçekleşmediği için öncelik seviyesi ”Orta” olarak belirlenir.
4. **Eskalasyon ve Aksiyon:** Analist, eğer birden fazla IP’den benzer aktiviteler geliyorsa durumu daha ileri analiz için Seviye 2 analiste bildirir ve ilgili IP adresinin güvenlik duvarı üzerinden geçici olarak engellenmesi için bir kural başlatır.

* **Dinamik Analiz:**
  - Sanal makine veya sandbox ortamında çalıştırma
  - Sistem çağrıları ve ağ trafiğinin izlenmesi
  - Bellek değişikliklerinin analizi
  - Anti-analiz tekniklerinin tespiti ve bypass edilmesi
* **Davranışsal Analiz:**
  - Dosya sistemi aktivitelerinin izlenmesi
  - Registry değişikliklerinin takibi
  - Ağ bağlantılarının analizi
  - Süreç ve servis değişikliklerinin kontrolü

9.1.8 İletişim Planları ve Paydaş Yönetimi

Etkin bir olay müdahale planının en önemli bileşenlerinden biri, iç ve dış paydaşlarla nasıl iletişim kurulacağını belirleyen önceden hazırlanmış bir iletişim planıdır. Bu plan, olayın teknik yönlerinin yanı sıra organizasyonun itibarı ve paydaş güveni üzerindeki potansiyel etkilerini yönetmek için tasarlanmıştır.

* **Kritik Paydaşlar:**
  - **İç Paydaşlar:** Üst yönetim, hukuk departmanı, halkla ilişkiler, insan kaynakları ve kritik iş birimi yöneticileri.
  - **Dış Paydaşlar:** Yasal ve düzenleyici kurumlar (örneğin Kişisel Verileri Koruma Kurumu), kolluk kuvvetleri, müşteriler, tedarikçiler ve basın.

Pratik Senaryo - İletişim Akışı:

Bir veri sızıntısı durumunda iletişim süreci aşağıdaki gibi işleyebilir:

1. **Dahili Bildirim:** CSIRT lideri, veri sızıntısının tespitini hemen üst yönetime ve ilgili iç paydaşlara bildirir. Teknik ekip olayın kapsamını belirlerken, halkla ilişkiler ve hukuk departmanları potansiyel yasal ve kamuoyu yansımaları için hazırlık yapar.
2. **Hukuki Bildirim:** Hukuk departmanı, veri sızıntısının yürürlükteki yasalara (ör. KVKK) göre bildirim gerektirip gerektirmediğini değerlendirir. Yasal bildirim süresine uyarak, ilgili düzenleyici kurumlara ”derhal” bildirimde bulunulur.
3. **Dış Paydaş İletişimi:** İletişim planına uygun olarak, halkla ilişkiler ekibi, durumu net, şeffaf ve abartısız bir dille açıklayan bir basın açıklaması veya müşteri bildirimi hazırlar. Bu açıklama, olayın doğasını, alınan önlemleri ve gelecekteki adımları içerir ve organizasyonun itibarını korumaya yardımcı olur.

9.1.9 SOAR (Security Orchestration, Automation and Response)

SOAR sistemleri, olay müdahale süreçlerini otomatikleştirerek yanıt süresini kısaltır ve insan hatasını azaltır.

* **Otomatik Müdahale Playbook'ları:**
  - **Oltalama E-posta Müdahalesi:**
    1. E-postanın otomatik karantinaya alınması
    2. Benzer e-postaların tüm posta kutularında aranması
    3. Şüpheli URL'lerin otomatik engellenmesi
    4. Etkilenen kullanıcıların bilgilendirilmesi
  - **Zararlı Yazılım Müdahalesi:**
    1. Etkilenen sistemin otomatik izolasyonu
    2. Zararlı yazılımın karantinaya alınması
    3. Sistem geri yükleme noktası oluşturulması
    4. IOC'lerin SIEM'e otomatik eklenmesi
* **Kolluk Kuvvetleri ile İşbirliği:** Özellikle ciddi siber suçlarda, Türkiye'deki Ulusal Siber Olaylara Müdahale Merkezi (USOM) ve Siber Olaylara Müdahale Ekipleri (SOME) gibi ulusal otoritelerle koordinasyon sağlanması zorunludur.
* **Adli Geçerlilik:** Bir olay sırasında toplanan delillerin mahkemede geçerli sayılması için "gözetim zinciri" (Chain of Custody) olarak bilinen bir prosedürün titizlikle uygulanması gerekir. Bu prosedür, delilin toplanmasından mahkemeye sunulmasına kadar geçen her adımın belgelenmesini, delil bütünlüğünün (hash değerleri ile) korunmasını ve yetkisiz erişime karşı güvenli bir şekilde saklanmasını gerektirir.

Olay müdahale süreci, sadece teknik bir problem çözme döngüsü değil, aynı zamanda bütüncül bir kriz yönetimi sürecidir. Bu durum, BT ve güvenlik ekiplerinin, hukuk, iletişim ve üst yönetim gibi diğer departmanlarla sürekli ve entegre bir işbirliği içinde olması gerektiği anlamına gelir. Bu işbirliğinin eksikliği, teknik olarak başarılı bir müdahalenin bile yasal veya itibari bir başarısızlıkla sonuçlanmasına neden olabilir.

9.2 Dijital Adli Bilişim Süreçleri

9.2.1 Güvenlik Olayı İzleme ve Alarm Korelasyonu

Güvenlik Olayı ve Bilgi Yönetimi (SIEM) sistemleri, bir organizasyonun ağındaki çeşitli kaynaklardan gelen günlük (log) verilerini toplayarak şüpheli olayları tespit eder. Alarm korelasyonu, bu sistemlerin sağladığı en önemli işlevlerden biridir. Tek bir olaydan kaynaklanan yüzlerce gereksiz alarmın ("alarm gürültüsü") filtrelenmesini ve bu karmaşanın gerçek tehditlerin gizlenmesine yol açmasının önlenmesini sağlar.

* **API Entegrasyonları:**
  - Güvenlik duvarı kurallarının otomatik güncellenmesi
  - EDR sistemleri ile entegrasyon
  - Tehdit istihbaratı platformları ile veri alışverişi
  - İş emri sistemleri ile entegrasyon
* **Pratik Korelasyon Kural Örnekleri:**
  - **Brute Force Saldırı Tespiti:** Bir SIEM sistemi, "aynı IP adresinden 30 saniye içinde 6'dan fazla başarısız oturum açma girişimi olursa alarm üret" kuralı ile deneme yanılma saldırılarını otomatik olarak tespit edebilir.

- **Zararlı Yazılım Kontrolü:** ”Endpoint koruma sistemi tarafından zararlı olarak tespit edilen bir IP adresi aynı zamanda kritik bir sunucuya oturum başlatırsa alarm üret” kuralı, saldırganın yanal hareketlerini saptamaya yardımcı olur.
- **Log Kaynağı Davranışı Tespiti:** Saldırganlar, tespit edilmekten kaçınmak için log kaynaklarını devre dışı bırakabilirler. Bu duruma karşı ”bir uç sistem 1 saatten uzun süre log göndermemişse alarm üret” gibi kurallar yazılabilir.

9.2.2 Veri Bilimi ve Yapay Zeka Teknikleri

Modern DFIR süreçleri, büyük veri analizi ve yapay zeka tekniklerinden faydalanır.

* **Makine Öğrenmesi Uygulamaları:**
  - **Anormal Davranış Tespiti:** Kullanıcı davranış analizi (UBA), ağ trafiği anomali tespiti, dosya sistemi aktivite analizi
  - **Tehdit İstihbaratı Analizi:** IOC kümeleme ve sınıflandırma, saldırı kampanyası ilişkilendirme, tehdit aktörü profilleme
* **Derin Öğrenme Uygulamaları:** Zararlı yazılım sınıflandırma, şifrelenmiş trafikte anormal davranış tespiti, sıfırıncı gün saldırılarının tespiti

9.2.3 İlk Müdahale ve Triyaj Prosedürleri

Olay tespiti yapıldıktan sonraki ilk ve en kritik adım, olayın kapsamını ve önceliğini belirleyen triyaj prosedürleridir. Bu süreç, bir olayın yanlış pozitif mi, yoksa gerçek bir güvenlik ihlali mi olduğunu hızla belirlemeye ve doğru müdahale ekibini atamaya odaklanır.

* **Triyaj Süreci Adımları:**
  1. **Gözden Geçirme:** Seviye 1 analist, SIEM'den gelen alarmı inceler ve bilinen bir güvenlik tarama aracı veya yazılım güncellemesi gibi meşru bir aktivite olup olmadığını kontrol eder.
  2. **Doğrulama:** Eğer şüphe devam ederse, analist ilgili log kayıtlarını (güvenlik duvarı, uygulama, sistem logları) ve ağ trafiğini inceleyerek olayın gerçek bir saldırı olduğunu doğrular.
  3. **Sınıflandırma ve Önceliklendirme:** Olay, önceden tanımlanmış kategorilere (ör. oltalama, fidye yazılımı) ve risk matrisine göre (Etki x Ciddiyet) sınıflandırılır ve önceliklendirilir.
  4. **Eskalasyon:** Olayın ciddiyetine göre, sorumlu yöneticilere ve daha ileri düzeyde teknik yetkinliğe sahip Seviye 2 analistlere bildirimde bulunulur.

9.2.4 Delil Toplama ve Gözetim Zinciri Yönetimi

Dijital delillerin toplanması, adli bilişim sürecinin temelini oluşturur. Bu delillerin, hukuki süreçlerde geçerli sayılabilmesi için değiştirilmediğinden, silinmediğinden ve manipüle edilmediğinden emin olunması gereklidir. Delilin bütünlüğünü korumak için, toplama işlemi sırasında dosyanın ”hash” değeri (parmak izi) hesaplanır ve herhangi bir değişiklikte bu değerin değişeceği kabul edilir.

* **Gözetim Zinciri (Chain of Custody):** Bu kavram, delilin olay yerinden toplanmasından mahkemeye sunulmasına kadar geçen süredeki tüm adımların titizlikle belgelenmesini ifade eder. Bu zincir, delilin doğru bir şekilde işlendiğini ve manipüle edilmediğini kanıtlayarak hukuki geçerliliğini sağlar.

Delil Gözetim Zinciri Formu Örneği

Delil Numarası	Delil Tanımı	Toplama Tarihi/Saati	Toplama Yeri	Hash Değeri (SHA25
IR-2024-001-H1	Sunucu 1'in fiziksel disk imajı	2024-05-15, 10:00	Sunucu Odası A	`e4c8e75...`
IR-2024-001-R1	Sunucu 1'in RAM dökümü	2024-05-15, 10:05	Sunucu Odası A	`3a5a7d9...`

9.2.5 Kök Neden Analizi ve Saldırı Vektörü Belirleme

Kök Neden Analizi (RCA), bir siber saldırının yalnızca yüzeysel nedenlerini (ör. bir bilgisayarın enfekte olması) değil, aynı zamanda olayın temelinde yatan asıl nedeni bulmayı amaçlayan sistematik bir çözme yöntemidir. Bu analiz, "neden" sorusunu tekrar tekrar sorarak saldırının nasıl gerçekleştiğini ve hangi zafiyetin sömürüldüğünü detaylıca ortaya çıkarır.

- **Pratik Senaryo - RCA Uygulaması ("5 Neden" Tekniği):**
  - **Sorun:** Kurumun web sitesi, bir siber saldırı sonucu erişilemez hale geldi.
  - **1. Neden?** Web sitesi neden erişilemez hale geldi? Çünkü saldırgan SQL enjeksiyonuyla veritabanını sildi.
  - **2. Neden?** Saldırgan neden veritabanını silebildi? Çünkü web uygulamasının kullanıcı girdileri düzgün bir şekilde filtrelenmiyordu.
  - **3. Neden?** Girdiler neden filtrelenmiyordu? Çünkü geliştirme ekibi bu güvenlik zafiyetini göz ardı eden eski bir kütüphane kullanıyordu.
  - **4. Neden?** Bu zafiyet neden göz ardı edildi? Çünkü geliştirme yaşam döngüsüne (SDLC) düzenli güvenlik testleri ve kod incelemeleri entegre edilmemişti.
  - **Kök Neden:** Yazılım geliştirme süreçlerinde güvenli kodlama politikalarının ve testlerinin olmamasıdır. Bu analiz, yüzeysel bir teknik sorunun (SQL enjeksiyonu) aslında daha derin bir sistemsel politika eksikliğinden kaynaklandığını göstermektedir.

9.2.6 Zaman Çizelgesi Oluşturma ve Saldırı Rekonstrüksiyonu

Olayın her aşamasının kronolojik olarak sıralandığı bir zaman çizelgesi, saldırının nasıl başladığını, nasıl yayıldığını ve hangi sistemleri etkilediğini anlamak için hayati bir araçtır. Adli bilişim uzmanları, sistem, uygulama ve ağ log dosyalarını kullanarak saldırının tüm seyrini yeniden kurgular.

- **Pratik Yönergeler - Zaman Çizelgesi Oluşturma:**
  1. **Veri Toplama:** Etkilenen sistemlerden güvenlik logları (`security.evtx` - Windows) veya kimlik doğrulama logları (`/var/log/auth.log` - Linux) toplanır.
  2. **Normalizasyon:** Farklı formatlardaki loglar, ortak bir formata ve zaman dilimine dönüştürülür.
  3. **Olay Akışını Kurgulama:** Toplanan olaylar, tarih ve saat bilgisine göre sıralanır. Örnek:
     - `09:30: Yeni kullanıcı (attacker_user) oluşturuldu.`
     - `09:32: attacker_user RDP ile sisteme bağlandı.`
     - `09:35: attacker_user yetki yükseltme saldırısı gerçekleştirdi.`

  Bu adımlar, Microsoft Office SmartArt gibi araçlar kullanılarak görsel bir zaman çizelgesine dönüştürülebilir, bu da olay akışının daha kolay anlaşılmasını sağlar. Zaman çizelgesi, adli bilişimin temel amacı olan "ne oldu?" sorusuna yanıt verirken, Kök Neden Analizi "neden oldu?" sorusuna yanıt verir. Bu iki sürecin entegre çalışması, olay müdahale süreçlerinin sadece reaktif değil, aynı zamanda proaktif ve önleyici olmasını sağlar.

9.3 Uzmanlık Alanları

9.3.1 Kısa Vadeli ve Uzun Vadeli Sınırlama Stratejileri

Sınırlama, bir siber saldırının yayılmasını durdurarak etkisini en aza indirmeyi amaçlayan en kritik aşamadır. Bu aşama, hem anlık hem de kalıcı önlemleri içerir.

- **Kısa Vadeli Stratejiler:** Saldırı anında uygulanan acil müdahale adımlarıdır.

    - Etkilenen sistemlerin ağdan izole edilmesi veya karantinaya alınması.
    - Saldırganın kullandığı hesapların veya erişim yetkilerinin askıya alınması.
    - Saldırganın bilinen IP adreslerinin veya alan adlarının güvenlik duvarında engellenmesi.
- **Uzun Vadeli Stratejiler:** Gelecekte benzer saldırıların önlenmesi için kalıcı iyileştirmelerdir.
    - Çok Faktörlü Kimlik Doğrulama (MFA) ve geçiş anahtarı (passkeys) gibi güçlü kimlik doğrulama mekanizmalarının uygulanması.
    - Düzenli güvenlik denetimleri ve sızma testleri gerçekleştirilmesi.
    - Eski sistem ve yazılımların güncel yamalarla korunması veya değiştirilmesi.
    - Ağın Sıfır Güven (Zero Trust) prensipleriyle yeniden yapılandırılması.

9.3.2 Ağ İzolasyonu ve Sistem Karantina Teknikleri

Ağ izolasyonu, potansiyel tehditlerin yayılmasını önlemek için sistemleri mantıksal veya fiziksel olarak ayırma işlemidir. Bu, olay müdahale süreçlerini kolaylaştırır ve saldırı etkisinin lokalize edilmesini sağlar.

- **İzolasyon Türleri:**
    - **Fiziksel İzolasyon:** Cihazların tamamen ayrı fiziksel ağ altyapılarına bağlanması. Yüksek güvenlik gerektiren ortamlar için kullanılır.
    - **Mantıksal İzolasyon:** Aynı fiziksel ağ üzerinde Sanal Yerel Ağlar (VLAN) veya mikrosegmentasyon kullanılarak farklı mantıksal alt gruplar oluşturulur.

Pratik Yönergeler - Komut Satırı ile İzolasyon:

- **Windows Ortamı:**
    - Bir IP adresinden gelen trafiği engellemek için:
      ```
      netsh advfirewall firewall add rule name ="BlokIP" dir =in action =block remoteip =192.1
      ```
    - Belirli bir programın gelen ve giden tüm bağlantılarını engellemek için:
      ```
      netsh advfirewall firewall add rule name ="BlokErisim" dir =in action =block program ="C
      ```
- **Linux Ortamı:**
    - Bir IP adresini engellemek için:
      ```
      iptables -A INPUT -s 192.168.1.10 -j DROP
      ```
    - Belirli bir porta gelen trafiği engellemek için:
      ```
      iptables -A INPUT -p tcp --dport 22 -j DROP
      ```

PowerShell betikleri, Azure gibi bulut ortamlarında sanal makineleri (VM) yeni bir alt ağa taşıyarak veya bir ağ güvenlik grubu (NSG) atayarak karantinaya almak için de kullanılabilir.

9.3.3 Kötü Amaçlı Yazılım Kaldırma ve Sistem İyileştirme Prosedürleri

Saldırgan ve kötü amaçlı yazılım kontrol altına alındıktan sonra, sistemden tamamen temizlenmesi gerekir. Bu süreç, otomatik veya manüel yöntemlerle gerçekleştirilebilir.

- **Otomatik Yöntemler:** Microsoft Kötü Amaçlı Yazılımları Temizleme Aracı (MSRT) veya Sophos Scan & Clean gibi virüs temizleme araçları, yaygın olarak bilinen tehditleri kaldırmak için kullanılır.

- **Manüel Yöntemler:** Daha karmaşık tehditler (rootkit'ler veya gelişmiş kalıcı tehditler) için manüel müdahale gerekebilir.
  1. **Sistem Hizmetlerini Durdurma:** Kötü amaçlı bir hizmetin kalıcı olarak kaldırılması için yönetici komut isteminde `sc delete "SERVICE NAME"` komutu kullanılır.
  2. **Kayıt Defteri Temizliği:** Zararlı yazılımlar genellikle sistem başlangıcında çalışmak için kayıt defterine girdi ekler. `regedit` kullanılarak `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\` gibi anahtarlardaki şüpheli girdiler silinir. Bu işlemden önce kayıt defterinin yedeklenmesi kritik öneme sahiptir.
  3. **Sistem Dosyalarını Onarma:** Zarar görmüş sistem dosyalarını onarmak için `sfc /scannow` ve `DISM /Online /Cleanup-Image /RestoreHealth` komutları kullanılır.

9.3.4 İş Sürekliliği ve Hizmet Restorasyonu

Sistemler temizlendikten sonra, iş operasyonlarının normale döndürülmesi sürecine geçilir. Bu süreç, önceden hazırlanmış bir İş Sürekliliği Planı (BCP) ve Felaket Kurtarma (DR) planı ile yönetilir. Felaket Kurtarma, kritik iş uygulamalarının kesintisiz çalışması veya en kısa sürede geri döndürülebilmesi amacıyla tüm sistem ve verilerin farklı bir lokasyonda kopyalanması hizmetidir. Kurtarma sürecinde, verilerin güvenli ve kötü amaçlı yazılımdan arınmış yedeklerden geri yüklendiğinden emin olunmalıdır.

9.3.5 Kurtarma Doğrulama ve Güvenlik Duruşu Değerlendirmesi

Kurtarma işlemi tamamlandıktan sonra, sistemlerin tamamen güvenli olduğunun doğrulanması hayati önem taşır. Bu doğrulama, saldırganın geri dönmesini engelleyecek şekilde güvenlik duruşunun güçlendirildiğini teyit eder. Bu süreç, tüm yamaların ve güncellemelerin uygulandığını, kötü amaçlı yazılım izlerinin kalmadığını ve yeni bir tehdidin ortaya çıkmadığını doğrulamak için uç nokta tespit ve yanıtı (EDR) veya genişletilmiş tespit ve yanıt (XDR) gibi çözümlerle davranışsal analizler yapılmasını içerir.

Olay müdahale döngüsü, sadece geçmişe dönük hasar onarımı değil, aynı zamanda geleceğe dönük bir güvenlik duruşu inşa etmeyi hedefler. Kısa vadeli reaktif adımlar (izolasyon) ile uzun vadeli proaktif stratejiler (yama yönetimi, MFA) arasındaki doğru denge, bir organizasyonun dayanıklılığını belirler.

9.4 Olay Sonrası Süreçler

9.4.1 Modern DFIR Araçları ve Platformları

Dijital Adli Bilişim ve Olay Müdahale (DFIR) süreçlerinde kullanılan çeşitli araçlar, sürecin etkinliğini ve verimliliğini artırır. İşte en önemli DFIR araçları ve özellikleri:

- **Velociraptor:**
  - Açık kaynak kodlu uç nokta görünürlük ve toplama aracı
  - VQL (Velociraptor Query Language) ile güçlü sorgulama yetenekleri
  - Ölçeklenebilir mimari ve hızlı veri toplama
  - Özelleştirilebilir "hunt" (av) senaryoları oluşturma imkanı
- **GRR Rapid Response:**
  - Google tarafından geliştirilen açık kaynak incident response framework
  - Python tabanlı uzaktan adli analiz yetenekleri
  - Büyük ölçekli kurumsal ortamlarda etkili kullanım
  - Otomatik veri toplama ve analiz iş akışları

- **Autopsy/Sleuth Kit:**
  - Açık kaynak dijital adli bilişim platformu
  - Disk imajı analizi ve dosya kurtarma özellikleri
  - Timeline analizi ve hash karşılaştırma
  - Çoklu format desteği (E01, dd, raw vb.)
- **Binalyze IREC:**
  - Hızlı forensik veri toplama ve analiz platformu
  - Tam sistem görünürlüğü ve bellek analizi
  - Otomatik raporlama ve delil zinciri yönetimi
  - Cloud entegrasyonu ve uzaktan toplama özellikleri
- **Redline:**
  - FireEye tarafından geliştirilen ücretsiz uç nokta analiz aracı
  - Host-based olay müdahale ve tehdit avcılığı
  - Bellek ve sistem analizi özellikleri
  - IOC tarama ve tehdit göstergesi analizi
- **KAPE (Kroll Artifact Parser and Extractor):**
  - Hedefli delil toplama ve işleme aracı
  - Özelleştirilebilir modüller ve hedef profilleri
  - Hızlı veri toplama ve otomatik işleme
  - Paralel işleme ve çoklu format desteği
- **THOR (Lite):**
  - İleri düzey IOC tarama ve tehdit avcılığı aracı
  - YARA kuralları ve özel imza desteği
  - Düşük sistem etkisi ve hızlı tarama
  - Özelleştirilebilir tarama profilleri

Araç Seçim Kriterleri:

- **Kullanım Senaryosu:** Olay türüne ve kapsamına uygun araç seçimi
- **Ölçeklenebilirlik:** Büyük organizasyonlarda etkin kullanım
- **Otomatizasyon:** Tekrarlanan görevlerin otomatikleştirilmesi
- **Entegrasyon:** Mevcut güvenlik altyapısı ile uyum
- **Maliyet:** Açık kaynak vs ticari çözümler değerlendirmesi

9.4.2 Adli Bilişim Hazırlık ve Planlama

Adli bilişim, dijital ortamlarda işlenen suçları aydınlatmak için elektronik cihazlardan delil toplama, analiz etme ve hukuki raporlama sürecidir. Bir olay meydana gelmeden önce adli bilişim hazırlığının yapılması, müdahale süresini ve maliyetini azaltır. Bu hazırlık, olay müdahale planına adli bilişim süreçlerinin entegre edilmesini, personelin gerekli eğitimleri almasını ve yazma engelleyiciler, adli bilişim yazılımları (Autopsy, FTK Imager) gibi gerekli araçların hazır bulundurulmasını içerir.

9.4.3 Delil Elde Etme: Canlı Sistem ve Post-mortem Analiz

- **Canlı Sistem Analizi (Live Forensics):** Bir sistem çalışırken, sistem kapandığında kaybolacak olan geçici (volatile) verilerin (bellek içeriği, çalışan süreçler, ağ bağlantıları) toplanmasıdır. Bu veriler, saldırganın sisteme nasıl sızdığına ve ne tür işlemler yaptığına dair kritik ipuçları sağlar.
- **Post-mortem Analiz (Post-mortem Forensics):** Sistem kapalıyken, kalıcı veri depolama birimlerinin (sabit disk, USB bellek) incelenmesidir. Bu işlem, bir delil diskinin bit-bit kopyasının (imajının) alınmasıyla gerçekleştirilir ve bu imaj üzerinde analiz yapılır.

9.4.4 Bilgisayar Adli Bilişimi: Windows, Linux ve macOS İncelemesi

Her işletim sisteminin kendine özgü yapısı, adli inceleme yöntemlerini de farklılaştırır.

- **Windows:** Kayıt defteri (Registry) ve merkezi log yönetimi sayesinde analiz daha çok bu verilere odaklanır. Kayıt defteri, kullanıcı aktiviteleri (çalıştırılan programlar, dosya geçmişi) ve sistem yapılandırmaları hakkında zengin bilgi içerir.
- **Pratik Komut Örnekleri (PowerShell):**

```
Get-EventLog -LogName Security -Newest 100
Get-ItemProperty -Path "HKCU:\Software\Microsoft\Windows\CurrentVersion\Run"
```

- **Linux:** Dağıtık günlükleme sistemi ve metin tabanlı yapılandırma dosyaları, daha çok manüel incelemeyi gerektirir.
- **Pratik Komut Örnekleri:**

```
cat /var/log/auth.log | grep "Failed password"
find / -name "backdoor.sh"
```

9.4.5 Ağ Adli Bilişimi ve Trafik Analizi

Ağ adli bilişimi, bir siber saldırının izini sürmek, saldırı vektörünü belirlemek ve zararlı aktiviteyi ortaya çıkarmak için ağ trafiğini analiz etmeye odaklanır. Trafik, paket yakalama dosyaları (`.pcap`) halinde kaydedilebilir ve daha sonra incelenebilir.

- **Araçlar:**
  - **Wireshark:** Ağ trafiğini yakalayan, analiz eden ve protokolleri ayrıştıran en yaygın araçlardan biridir.
  - **Snort:** Ağ Saldırı Tespit Sistemi (NIDS) olarak çalışarak önceden tanımlanmış kurallara uyan şüpheli paketler için alarm üretir.

Pratik Senaryo - Wireshark Analizi: Bir şüphelinin web sitesine yaptığı kullanıcı adı ve parola girişini incelemek için Wireshark kullanılabilir.

1. **Yakalama:** Wireshark ile ağ arayüzü seçilerek paket yakalama başlatılır.
2. **Filtreleme:** Yakalama tamamlandıktan sonra, trafiği daraltmak için `http.request.method == "POST"` gibi filtreler uygulanır.
3. **Paket İncelemesi:** İlgili pakete tıklanarak TCP akışı takip edilir ve bu sayede oturum açma formuna girilen kullanıcı adı ve parola gibi hassas bilgiler incelenir.

9.4.6 Mobil Cihaz Adli Bilişimi ve Bulut Adli Bilişimi Zorlukları

Bu alanlar, adli bilişimin en zorlu alanlarından biridir. Mobil cihazların cihaz çeşitliliği, işletim sistemi farklılıkları ve güçlü şifreleme mekanizmaları, delil elde etmeyi karmaşık hale getirir. Bulut adli bilişimi ise verilerin coğrafi olarak dağıtık olması, yasal yetki alanlarının belirlenmesi ve delilin mülkiyeti gibi sorunları beraberinde getirir.

- **Pratik Yönergeler - Mobil Cihaz Delil Toplama:**
  1. **İzolasyon:** Cihaz, uzaktan silme (remote wipe) veya veri manipülasyonu tehdidine karşı Faraday çantası gibi ekipmanlarla ağdan fiziksel olarak izole edilir.
  2. **Delil Elde Etme:** Cellebrite UFED veya Oxygen Forensic Detective gibi özel araçlarla cihazın fiziksel veya mantıksal imajı alınır.

Bu alandaki adli bilişim uzmanları, suçluların kullandığı teknolojinin gerisinde kalmamak için sürekli öğrenme ve araç kitlerini güncelleme baskısı altındadır. Mobil ve bulut teknolojilerindeki hızlı gelişmeler, geleneksel adli bilişim metodolojilerinin bu yeni dinamiklere adapte edilmesi gerektiğini göstermektedir.

9.5 Uzmanlık Gerektiren Adli Bilişim ve İleri Düzey Analiz

9.5.1 Bellek Adli Bilişimi ve Uçucu Veri Analizi

Bellek (RAM), bir sistem çalışırken oluşan ve sistem kapatıldığında kaybolan uçucu verileri (işlem parolaları, çalışan süreçler, ağ bağlantıları) içerir. Bu veriler, saldırının kapsamını ve amacını belirlemek için son derece değerli kanıtlar barındırır.

- **Araçlar ve Teknikler: Volatility Framework** gibi bellek analizi araçları, RAM dökümünden bu uçucu verileri çıkarmak için kullanılır.
- **Pratik Yönergeler - Volatility ile Analiz:**
  1. **Bellek Dökümü Alma:** Linux'ta `dd` komutu veya Windows için geliştirilmiş özel araçlar kullanılarak sistemin belleğinin ham kopyası (dökümü) alınır.
  2. **Analiz:** Volatility, bu döküm dosyası üzerinde bir dizi komut çalıştırarak analiz yapar:
     * `python3 vol.py -f <filename> windows.pslist`: Çalışan tüm süreçleri ve ilişkili PID'lerini listeler. Birçok kötü amaçlı yazılım, gizli süreçler oluşturur ve bunlar bu listelerde tespit edilebilir.
     * `python3 vol.py -f <filename> windows.netscan`: Bellek dökümü anındaki tüm ağ bağlantılarını, ilişkili süreçleri ve port bilgilerini listeler. Bu komut, bir komuta-kontrol (C2) sunucusuyla olan iletişimi ortaya çıkarabilir.

9.5.2 Veritabanı Adli Bilişimi ve Uygulama Log Analizi

Veritabanları ve uygulamalar, kullanıcı aktiviteleri, hatalar ve kritik işlemler hakkında detaylı kayıtlar (loglar) tutar. Bu loglar, siber saldırıların tespiti, saldırganın hareketlerinin izlenmesi ve adli süreçlerde delil olarak kullanılması açısından hayati öneme sahiptir.

- **Veritabanı Logları (SQL Server Örneği):**
  * SQL Server, tüm işlemleri bir işlem günlüğünde (`.ldf` dosyası) kaydeder.
  * Adli amaçlar için, `fn_dblog()` gibi belgelenmemiş fonksiyonlar, aktif işlem günlüğünü sorgulamaya ve "kimin ne zaman, hangi veriyi değiştirdiği" gibi sorulara yanıt bulmaya olanak tanır.
- **Uygulama Logları:**

    - **Linux Ortamında:** `grep`, `awk` ve `sed` gibi komut satırı araçları, `syslog` veya `auth.log` gibi log dosyalarını manüel olarak incelemek için kullanılır.
    - **Büyük Ölçekli Sistemlerde:** Logların merkezi olarak toplanması, işlenmesi ve görselleştirilmesi için Elasticsearch, Logstash, Kibana (ELK Stack) gibi kurumsal çözümler kullanılır.

9.5.3 Endüstriyel Kontrol Sistemi (ICS/SCADA) Adli Bilişimi

ICS ve SCADA sistemleri, imalat, enerji ve su arıtma gibi kritik endüstriyel süreçleri yöneten sistemlerdir. Bu sistemler, eski teknolojiler, fiziksel etki potansiyeli ve özel ağ protokolleri nedeniyle benzersiz güvenlik ve adli bilişim zorluklarına sahiptir.

- **Adli Bilişim Yaklaşımı:**
    - **Ağ Segmentasyonu:** Saldırının operasyonel teknoloji (OT) ağından bilgi teknolojileri (IT) ağına yayılmasını önlemek için iki ağ fiziksel veya mantıksal olarak ayrılmalıdır.
    - **Protokol Analizi:** Modbus, DNP3 gibi endüstriyel protokollere yönelik uzmanlık gerektiren paket analizi yapılır. Bu, saldırganın fiziksel süreçleri manipüle etmeye yönelik komutlarını ortaya çıkarabilir.
    - **Sistem Log Analizi:** SCADA master ünitesindeki ve İnsan-Makine Arayüzü (HMI) cihazlarındaki loglar, saldırganın eylemlerini yeniden kurgulamak için incelenir.

9.5.4 Sanal Makine ve Konteyner Adli Bilişimi

Sanallaştırma teknolojileri, adli bilişim süreçlerini karmaşıklaştırır.

- **Sanal Makineler (VM):** Bir fiziksel makinenin dijital kopyasıdır ve kendi işletim sistemine sahiptir. VM üzerinde adli bilişim, fiziksel bir makinedekiyle benzer adımlar içerir, ancak analiz sanal disk imajları (`.vmdk`, `.vdi`) üzerinde yapılır.
- **Konteynerler (Docker):** İşletim sistemini sanallaştıran ve uygulamayı platformdan bağımsız çalıştıran hafif ortamlardır. Konteynerler geçici ve dinamik olduğu için adli incelemesi zordur. Şüpheli bir konteyner tespit edildiğinde, `docker commit` komutuyla mevcut durumu yeni bir imaj olarak kaydedilebilir ve bu imaj üzerinde dosya sistemi, bellek ve log analizi yapılabilir.

9.5.5 Kripto Para ve Blockchain Adli Bilişimi

Kripto paralar, merkeziyetsiz bir defter teknolojisi olan blockchain üzerinde çalışır. Bu teknoloji, kullanıcıları takma adlar (pseudonymous) kullanarak anonimleştirdiği için finansal suç soruşturmaları için bir zorluk oluşturur. Ancak, blockchain'in değişmez bir kayıt defteri olması, bir kez kaydedilen işlemin manipüle edilememesi anlamına gelir ve bu durum, suçla ilgili kanıtın güvenli bir şekilde saklanmasını sağlar. Adli bilişim uzmanları, kripto para birimlerine ait işlem kayıtlarını takip ederek para akışını izlemeye ve bu akışları gerçek dünya kimlikleriyle ilişkilendirmeye çalışır.

9.6 Olay Sonrası Faaliyetler ve Kazanılan Dersler

9.6.1 Olay Dökümantasyonu ve Raporlama Gereksinimleri

Olay müdahalesinin her aşaması, gelecekteki analizler ve hukuki süreçler için detaylı bir şekilde belgelenmelidir. Olay sonrası rapor (post-mortem), organizasyonun süreç iyileştirmesi için temel bir kaynaktır ve olayın "kim, ne, nerede, ne zaman, neden ve nasıl" sorularına yanıt vermelidir.

– **Olay Sonrası Rapor Şablonu**

Bölüm	İçerik
Yönetici Özeti	Olayın kısa özeti ve iş üzerindeki etkisi.
Olay Zaman Çizelgesi	Olayın kronolojik olarak sıralanmış akışı.
Kök Neden Analizi	Olayın temelinde yatan zafiyetler ve nedenler.
Alınan Aksiyonlar	Sınırlama, yok etme ve kurtarma aşamalarında ya
Kazanılan Dersler ve Öneriler	Gelecekte benzer olayların önlenmesi için iyileşti

9.6.2 Kazanılan Dersler Analizi ve Süreç İyileştirme

Olaydan çıkarılan dersler, organizasyonel hafızayı güçlendirir ve aynı hataların tekrarlanmasını önler. Bu süreç, sürekli iyileştirme için bir geri bildirim döngüsü sağlar ve dört temel adımdan oluşur:

1. **Belirleme:** Olaydan elde edilecek önemli kazanımların ve başarıların belirlenmesi.
2. **Belgeleme:** Kazanılan derslerin, ilgili herkesin katkıda bulunabileceği bir formatta kaydedilmesi.
3. **Analiz Etme:** Elde edilen verilerin incelenerek anlamlı sonuçlar çıkarılması.
4. **Saklama:** Raporların, ekiplerin kolayca erişebileceği ortak bir dijital platformda arşivlenmesi.

9.6.3 Olaylardan Tehdit İstihbaratı Geliştirme

Bir siber saldırı, organizasyon için değerli bir tehdit istihbaratı (CTI) kaynağıdır. CTI, ham veriyi (ör. zararlı IP adresleri) işleyerek anlamlı ve eyleme dönüştürülebilir bilgilere dönüştürme sürecidir.

– **İki Önemli Kavram:**
  * **Tehlike Göstergeleri (IOCs):** Kötü amaçlı IP adresleri, dosya hash'leri veya zararlı URL'ler gibi somut, teknik verilerdir. IOC'ler taktiksel düzeyde savunma için kullanılır.
  * **Taktik, Teknik ve Prosedürler (TTPs):** Saldırganların amaçlarına ulaşmak için kullandığı yöntemlerdir. TTP'ler, IOC'lere göre daha kalıcıdır ve operasyonel düzeyde savunma için kritik öneme sahiptir.

İstihbarat Üretimi: Bir oltalama saldırısı sonrası tespit edilen zararlı dosya hash'i ve komuta-kontrol sunucusu IP adresi gibi IOC'ler toplanır. Bu veriler, saldırganın kullandığı genel yöntemlerle (ör. belirli bir zararlı yazılım ailesi) ilişkilendirilerek istihbarat haline getirilir ve gelecekteki saldırılara karşı otomatik koruma sağlamak için SIEM ve EDR sistemlerine entegre edilir.

9.6.4 Eğitim ve Farkındalık Programı Güncellemeleri

Bir siber saldırıdan elde edilen dersler, çalışan farkındalık eğitimlerini ve güvenlik prosedürlerini güncellemek için kullanılmalıdır. Olay sonrası yapılan analizler, çalışanların zafiyetli davranışlarını (ör. oltalama e-postalarına tıklama) ortaya çıkarır. Bu geri bildirimler, eğitim materyallerine dahil edilerek simülasyon bazlı testlerle çalışanların farkındalığı artırılır. Siber güvenlik eğitimlerinin de tıpkı diğer eğitim programları gibi, çağın gereklerine göre sürekli revize edilmesi, bu alanda durağanlığın mümkün olmadığının bir göstergesidir.

9.6.5 Hukuki Süreç Desteği ve Uzman Tanık İfadeleri

Siber güvenlik uzmanları, bir adli soruşturma veya mahkeme sürecinde "uzman tanık" olarak görev alabilirler. Hazırladıkları bilimsel mütalaalar, hukuka uygun, somut ve bilimsel verilere dayanmalıdır. Uzman, bulgularını ve vardığı sonuçları mahkemede net, anlaşılır ve basit bir dille açıklamalıdır. Uzmanın sunduğu rapor, hâkimin maddi gerçeği aydınlatmasına ve delilleri doğru yorumlamasına yardımcı olan bir araç olarak kullanılır. Bu

durum, teknik uzmanlığın nihayetinde hukuki bir amaca hizmet ettiğini ve uzmanların raporlama ile iletişim becerilerinin de teknik yetkinlikleri kadar kritik olduğunu ortaya koymaktadır.

Bölüm 10

YÖNETİŞİM, RİSK YÖNETİMİ VE UYUMLULUK (GRC)

Giriş

Bu bölüm, bir kuruluşun siber güvenlik duruşunun temelini oluşturan, stratejik ve operasyonel GRC (Yönetişim, Risk ve Uyum) mekanizmalarını derinlemesine incelemektedir. Odak noktası, yalnızca teknik kontrollere odaklanmak yerine, güvenliği bir iş önceliği haline getiren ve kuruluşun hedefleriyle hizalayan entegre bir yaklaşım sunmaktır. GRC, organizasyonun güvenli ve doğru yolda kalmasını sağlayan strateji ve yapı olarak işlev görür, bir tripod gibi yönetim, risk yönetimi ve uyumluluk ayaklarını birleştirir.

10.1 Bilgi Güvenliği Yönetişim Çerçevesi

Bilgi güvenliği yönetişimi, liderlik, organizasyonel yapılar ve süreçlerden oluşan, bilgi varlıklarını korumayı amaçlayan yapılandırılmış bir çerçevedir. Bu çerçeve, güvenliğin ”nedenini”, ”ne olduğunu” ve ”kimin sorumlu olduğunu” ele alarak, taktiksel güvenlik tedbirlerinin ötesine geçer ve stratejik bir iş fonksiyonu olarak konumlanmasını sağlar. Bu, yalnızca teknoloji odaklı bir yaklaşımdan ziyade, organizasyonel hedefler ve risk toleransı ile uyumlu güvenlik uygulamalarını garanti eder.

10.1.1 Board-level Security Governance ve Oversight (Yönetim Kurulu Seviyesi Siber Güvenlik Gözetimi)

Yönetim kurulu seviyesinde siber güvenlik gözetimi, siber risklerin kurumsal risk yönetimi (ERM) stratejisinin ayrılmaz bir parçası olarak kabul edilmesini sağlar. Yönetim kurulunun rolü, detaylı teknik operasyonlara müdahale etmekten ziyade, stratejik yönü belirlemek ve kaynakların etkin bir şekilde ayrılmasını sağlamaktır.

Bu katmanda güvenlik, teknik bir konudan stratejik bir iş konusuna dönüşür. Bir siber olayın yol açabileceği itibar kaybı, finansal cezalar ve iş sürekliliği kesintileri gibi doğrudan iş sonuçları, bu risklerin yalnızca teknik değil, aynı zamanda finansal ve operasyonel riskler olarak algılanmasını zorunlu kılar. Bu geniş etki alanı, yönetim kurulunun bütçe onayı, strateji belirleme ve genel gözetim kapsamında bu konuyu aktif olarak ele almasını gerektirir. Bu dönüşüm, güvenlik liderlerinin (örneğin CISO), teknik jargon yerine iş diliyle (risk, yatırım getirisi (ROI), iş etkisi gibi) konuşma becerisine sahip olmasının önemini ortaya koymaktadır. Yönetim kurulu desteği, kaynak tahsisi ve politika onayı için hayati öneme sahiptir.

Yönetim kurulu düzeyinde etkili gözetim mekanizmalarının temel taşları şunlardır:

- **Siber Okuryazarlığın Artırılması:** Yönetim kurulu üyelerinin, genel siber riskler ve şirketi etkileyen özel siber riskler hakkında bilgi edinmesi esastır. Bu, özel eğitimler ve uzmanlarla yapılan görüşmelerle sağlanabilir.
- **Düzenli Raporlama:** Yönetim kurulu, siber güvenlikten sorumlu yöneticilerle düzenli olarak görüşmeli ve şirketin siber olgunluğunu ve dayanıklılığını anlamalarını sağlayan uygun metrikleri talep etmelidir. Bu metrikler, kolayca toplanabilen verilerden ziyade, en riskli alanlara odaklanmalıdır.
- **Görevlerin Ayrılması:** Siber güvenlik stratejisinin belirlenmesi ve bu stratejinin uygulanması arasında görev ayrımının sağlanması kritik öneme sahiptir. Bu, hesap verebilirliği ve etkinliği artırır.

10.1.2 Information Security Strategy ve Policy Development (Bilgi Güvenliği Stratejisi ve Politika Geliştirme)

Kapsamlı bir bilgi güvenliği stratejisi ve politikası, sürekli gelişen tehditlere ve karmaşık uyumluluk gereksinimlerine karşı koordineli ve uygulanabilir bir program oluşturur. Bu strateji, organizasyonun tüm güvenlik süreçlerini uçtan uca kapsayan, uygulanabilir ve iş hedeflerine odaklanmış olmalıdır.

Politika geliştirme süreci, tek seferlik bir olaydan ziyade döngüsel bir yapıya sahiptir. Süreç, risk değerlendirmesiyle başlar, ancak politikanın kendisi de risk yönetimi süreçlerinin etkinliğini ölçmek için kullanılır. Bu yaklaşım, güvenlik duruşunun dinamik ve sürekli değişen bir ortamda güncel kalmasını garanti eder.

Bir bilgi güvenliği politikasının adım adım geliştirilmesi için şu adımlar izlenmelidir:

1. **Mevcut Durum ve Risk Değerlendirmesi:** Sürecin ilk adımı, kuruluşun mevcut güvenlik durumunu, hassas verilerini (PII, finansal belgeler vb.), sistemlerini ve potansiyel zayıflıklarını belirleyen kapsamlı bir risk değerlendirmesi yapmaktır. Bu aşama, aynı zamanda organizasyon için kabul edilebilir risk seviyelerini de tanımlar.
2. **Yasal ve Sektörel Gereksinimlerin Belirlenmesi:** İlgili tüm yerel, ulusal ve sektörel yasa ve standartlar (KVKK, GDPR, HIPAA, ISO 27001 vb.) incelenir. Bu, politikanın yasal yükümlülüklerle uyumlu olmasını sağlar.
3. **Strateji ve Politika Geliştirme:** Değerlendirme sonuçlarına dayanarak, iş hedefleriyle uyumlu bir bilgi güvenliği stratejisi oluşturulur. Bu stratejinin temelini oluşturan yüksek seviyeli ilkeler, ardından detaylı ve konu bazlı politikalara (örneğin, kabul edilebilir kullanım, erişim kontrolü, parola politikası, şifreleme politikası) dönüştürülür. Politika, gizlilik, bütünlük ve erişilebilirlik gibi temel güvenlik hedeflerine odaklanmalıdır.
4. **Onay ve İletişim:** Geliştirilen politikalar üst yönetimden onay almalı ve tüm paydaşlara şeffaf bir şekilde iletilmelidir. Politikanın etkinliğini ölçmek için düzenli olarak risk değerlendirmeleri yapılmalı ve bu değerlendirmeler sonucunda belirlenen iyileştirme alanlarına göre politika güncellenmelidir.

10.1.3 Security Organization Structure ve Roles/Responsibilities (Güvenlik Organizasyon Yapısı ve Görev/Sorumluluklar)

Etkili bir yönetişim için net bir organizasyon yapısı ve tanımlanmış roller hayati öneme sahiptir. Siber güvenlik yönetişimi, görev ve sorumlulukların katmanlı bir yapıda olduğunu gösterir.

- **Yönetim Kurulu:** En üst düzeyde stratejik gözetim sağlar ve siber güvenliğin iş hedefleriyle uyumunu denetler. Bu, en üst düzeyde sorumluluk ve hesap verebilirliği tesis eder.
- **Bilgi Güvenliği Komitesi:** Stratejik düzeyde karar alıcı organdır. Kaynak ayrılması, kabul edilebilir risk seviyelerinin belirlenmesi ve risk işleme aksiyonlarının onaylanması gibi görevleri üstlenir. Bu katman, stratejiyi somut eylemlere dönüştüren bağlayıcı noktadır.

- **Bilgi Güvenliği Koordinasyon Ekibi:** Stratejik hedeflerin operasyonel olarak uygulanmasından sorumludur. Risk değerlendirmesi, varlık envanteri oluşturma ve acil durumlarda hasar tespiti gibi çalışmaları yürütür.
- **Bilgi Güvenliği Yöneticisi / Siber Güvenlikten Sorumlu Yetkili Yönetici (SGSYY):** Yönetim sistemi içindeki rol ve sorumlulukları denetler, güvenlik sorunlarına müdahale eder ve gerekli koordinasyonu sağlar.

Bu katmanlı yapı, her seviyede şeffaflık ve hesap verebilirlik sağlayarak, güvenliğin tek bir kişiye veya departmana yüklenmesini engeller. Bu, GRC'nin üçayağının (Yönetişim-Risk-Uyum) işlevsel olarak birleştirilmesini de destekler.

10.1.4 Security Culture Development ve Awareness Programs (Güvenlik Kültürü Geliştirme ve Farkındalık Programları)

Güvenlik kültürü, çalışanların güvenliği önceliklendirdiği ve ortak bir sorumluluk olarak gördüğü bir ortamdır. Zira birçok güvenlik ihlali, oltalama saldırıları veya kullanıcı hatalarından kaynaklanır.

Güvenlik kültürü oluşturmak, sadece eğitimlerle sınırlı kalmamalı, sürekli ve katılımcı bir süreç olmalıdır. Bu süreç, üst yönetimin desteğiyle başlar; yönetim kademesi, güvenlik standartlarını belirleyerek ve güvenliği teşvik ederek örnek olmalıdır. Ardından, çalışanlara düzenli olarak siber güvenlik eğitimleri verilmelidir. Bu eğitimler, kimlik avı saldırılarını tespit etme, parola yönetimi, fiziksel güvenlik ve çıkarılabilir medya kullanımı gibi konuları içermelidir.

Pratik uygulamalar ve simülasyonlar, teorik bilgilerin pekiştirilmesi için kritik öneme sahiptir. Oltalama (phishing) simülasyonları gibi pratik testler, çalışanların riskli davranışlarını değerlendirmek ve güvenli alışkanlıklar geliştirmelerine yardımcı olmak için kullanılmalıdır.

Adım Adım Oltalama (Phishing) Simülasyonu Yürütme Kılavuzu:

1. **Hedef Kitleyi Belirleyin:** Simülasyonun hangi çalışan gruplarını veya departmanlarını hedefleyeceğini netleştirin.
2. **Senaryo Oluşturun:** Mevcut şablonlardan veya sıfırdan bir senaryo seçin. Senaryolar genellikle gerçek dünyadaki oltalama tekniklerini taklit eder. Popüler senaryo örnekleri şunlardır:
   - **CEO Dolandırıcılığı:** Üst düzey bir yöneticinin acil para transferi talep ettiği bir e-posta, duygusal ve hiyerarşik bir tetikleyici kullanır.
   - **Sahte Şifre Sıfırlama:** LinkedIn gibi güvenilir bir platformdan geldiği iddia edilen, acil şifre sıfırlama uyarısı içeren bir e-posta, aciliyet hissi yaratır.
   - **İnsan Kaynakları Politikası Güncellemesi:** Çalışanlara iç ve rutin görünen, yeni bir sağlık hizmeti planı veya ödeme politikası hakkında bilgi veren bir e-posta.
3. **Simülasyonu Yapılandırın:** Oltalama e-postasının içeriğini ve yönlendireceği sahte web sayfasını hazırlayın. Bu aşamada, sahte bir giriş sayfası veya zararlı bir dosya indirme bağlantısı içeren bir "yük" (payload) yapılandırılır.
4. **Kampanyayı Başlatın:** Simülasyonu başlatma ve bitiş tarihlerini belirleyin. Hedef kitleye otomatik olarak e-postalar gönderilecektir.
5. **Sonuçları İzleyin ve Raporlayın:** Kampanya sonuçları (tıklama oranları, veri girişi, raporlama sayısı) izlenerek, riskli kullanıcılar ve güvenlik açıkları belirlenir.
6. **Eğitim Ataması:** Başarısız olan çalışanlara otomatik olarak düzeltici eğitimler atanır. Bu, sadece bir test değil, aynı zamanda sürekli bir öğrenme sürecidir.

10.1.5 Third-party Risk Governance ve Vendor Management (Üçüncü Taraf Risk Yönetimi ve Tedarikçi Yönetimi)

Üçüncü taraflara (tedarikçiler, iş ortakları, bulut servis sağlayıcıları) olan bağımlılık arttıkça, bu tarafların yarattığı riskleri yönetmek hayati hale gelmektedir. Kaynaklar, tek bir tedarikçiye aşırı bağımlılığın (single-source) maliyeti optimize edebilse de, operasyonel zafiyetler yarattığını ve iş sürekliliğini tehdit ettiğini belirtmektedir. Uzun süreli ve başarılı vendor ilişkileri, risklerin ortadan kalktığına dair yanlış bir güven duygusu yaratabilir. Bu durum, izleme ve acil durum planlaması çabalarını azaltmaya yol açabilir. Coğrafi olarak aynı bölgede bulunan birden fazla tedarikçi de, doğal afetler gibi lokal olayların tedarik zincirini aynı anda kesintiye uğratmasına neden olabilir.

Bu riskleri yönetmek için kapsamlı bir Üçüncü Taraf Risk Yönetimi (TPRM) programı kurulmalıdır:

1. **Kapsam ve Envanter:** Tedarik zinciri haritasını çıkararak tüm tedarikçileri, hizmet sağlayıcıları ve yüklenicileri kapsamlı bir şekilde listelemekle başlanır.
2. **Riske Dayalı Değerlendirme:** Tüm üçüncü taraflara aynı risk değerlendirme süreci uygulanmamalıdır. Bunun yerine, riske orantılı bir yaklaşım benimsenerek, hassas verilere erişenler gibi kritik tedarikçilere daha derinlemesine incelemeler yapılmalıdır.
3. **Durum Tespiti (Due Diligence):** Yeni bir tedarikçi ile çalışmaya başlamadan önce, siber güvenlik duruşu, finansal istikrar ve uyumluluk geçmişi gibi konularda kapsamlı bir durum tespiti yapılmalıdır. Bu, kabul edilemez risk seviyelerine sahip tedarikçilerle çalışmaktan kaçınmayı sağlar.
4. **Sözleşmeler ve SLA'lar:** Tedarikçilerle imzalanan sözleşmeler ve hizmet seviyesi anlaşmaları (SLA), beklentileri açıkça belirlemeli ve olası kesintilere karşı bir koruma sağlamalıdır.
5. **Sürekli İzleme ve Denetim:** Tedarikçi risk profillerindeki değişiklikler düzenli olarak değerlendirilmeli ve gerektiğinde denetimler yapılmalıdır. Bu, risklerin dinamik doğasına uyum sağlamak için esastır.

10.2 Kurumsal Risk Yönetimi ve Siber Risk

Risk yönetimi, işletmenin hedeflerine ulaşmasını engelleyebilecek olumsuz sonuçlara yol açabilecek her şeyi yönetme sürecidir. Bu süreç, belirsizliği ortadan kaldırmayı ve uyumluluk gerekliliklerini karşılamayı amaçlar.

10.2.1 ISO 31000 Risk Management Framework (ISO 31000 Risk Yönetim Çerçevesi)

ISO 31000:2018, bir yönetim sistemi standardı olmamasına rağmen, organizasyonların riskleri tanımlaması, analiz etmesi ve yönetmesi için sistematik bir rehber sunar. Bu çerçeve, kuruluşların risk bilincini artırarak karar alma süreçlerini destekler ve sürdürülebilir başarıyı teşvik eder.

Standardın temel ilkeleri şunları içerir:

- **Entegre Yaklaşım:** Risk yönetimi, tüm iş süreçlerine ve karar alma mekanizmalarına entegre edilmelidir.
- **Özelleştirilmiş ve Esnek Olmak:** Çerçeve, her organizasyonun kendine özgü ihtiyaçlarına göre uyarlanabilmelidir.
- **İnsan ve Kültür Odaklı Olmak:** İnsan faktörü ve risk bilinci, sürecin kritik bir parçasıdır. Herkesin sürece dahil edilmesi, risklerin daha iyi anlaşılmasını sağlar.
- **Dinamik ve Sürekli Güncellenen Yapı:** Risk yönetimi, sürekli iyileştirilebilir ve proaktif olmalıdır. Organizasyonlar, potansiyel tehditleri önceden belirleyerek beklenmedik durumlarla karşılaşma olasılığını azaltabilir.

10.2.2 Cyber Risk Quantification ve Business Impact Analysis (Siber Riskin Nicelleştirilmesi ve İş Etki Analizi)

Siber riskin nicelleştirilmesi, riskleri soyut terimler (”yüksek/orta/düşük”) yerine somut finansal değerlerle ifade etmeyi amaçlayan bir yaklaşımdır. Bu, karar alma süreçlerini daha objektif hale getirir ve güvenlik yatırımlarının iş üzerindeki etkisini net bir şekilde gösterir.

İş Etki Analizi (BIA) Adım Adım Yönergesi: BIA, bir siber olayın kritik iş operasyonları üzerindeki potansiyel etkilerini sistematik olarak değerlendiren bir süreçtir. BIA'nın çıktıları, Felaket Kurtarma Planı (DRP) için gerekli olan Kurtarma Süresi Hedefi (RTO) ve Kurtarma Noktası Hedefi (RPO) gibi hedeflerin belirlenmesinde temel veri kaynağını oluşturur. Bir BIA, kritik iş süreçlerinin ne kadar süre durabileceğini ve ne kadar veri kaybına dayanabileceğini değerlendirerek, teknik ekiplerin Felaket Kurtarma Planı (DRP) için net RTO ve RPO değerleri belirlemesini sağlar.

BIA'nın temel adımları şunlardır:

1. **Kapsamı Tanımlayın:** Hangi iş birimlerinin, süreçlerinin ve sistemlerinin analize dahil edileceğini netleştirin.
2. **Kritik İş Fonksiyonlarını Belirleyin:** Kuruluşun temel operasyonlarını ve hizmetlerini listeleyin. Her bir fonksiyon için potansiyel finansal, operasyonel ve itibar kayıplarını belirleyin.
3. **Bağımlılıkları Haritalandırın:** Kritik fonksiyonların hangi IT sistemlerine, verilere, personele ve üçüncü taraf hizmetlerine bağımlı olduğunu çıkarın.
4. **Etkileri Değerlendirin:** Her bir kesintinin zaman içindeki (örneğin, ilk birkaç saat, ilk 24 saat, ilk hafta) potansiyel etkilerini (gelir kaybı, yasal cezalar, itibar zedelenmesi) finansal olarak ölçün.

10.2.3 FAIR (Factor Analysis of Information Risk) Methodology (FAIR Metodolojisi)

FAIR, siber riskleri parasal terimlerle nicelleştiren ve subjektif (”yüksek/orta/düşük”) değerlendirmelerden objektif, veriye dayalı analize geçişi sağlayan uluslararası bir standarttır. Bu model, güvenlik yatırımlarının işe yaradığını göstermek ve bütçe kararlarını desteklemek için stratejik bir dilde iletişim kurmaya olanak tanır.

FAIR metodolojisi, riski iki ana bileşene ayırır:

– **Kayıp Olayı Frekansı (Loss Event Frequency - LEF):** Belirli bir süre zarfında bir kayıp olayının ne sıklıkta meydana gelme olasılığını ölçer.
– **Kayıp Büyüklüğü (Loss Magnitude - LM):** Bir kayıp olayının potansiyel etkisini finansal olarak ifade eder. Bu, birincil (üretkenlik kaybı, müdahale maliyeti) ve ikincil (itibar kaybı, yasal cezalar) kayıpları içerir.

Pratik FAIR Analiz Senaryosu: Bir Çalışanın Dizüstü Bilgisayarının Çalınması

– **Senaryo:** ”Ayrıcalıklı bir çalışanın, hassas müşteri verilerini içeren dizüstü bilgisayarının çalınmasıyla ilişkili riski analiz edin”.
– **Adım 1: Değerleme (Varlık Değeri):** Varlığın değeri sadece fiziksel cihaz değil, asıl olarak içerdiği hassas veridir (PII, entelektüel mülkiyet, vb.).
– **Adım 2: Tek Kayıp Beklentisi (Single Loss Expectancy - SLE) Hesaplaması:** Bu, tek bir olaydan kaynaklanan maliyettir.
  ∗ **Formül:** SLE = Varlık Değeri × Maruz Kalma Faktörü
  ∗ **Örnek Uygulama:**
    · Varlık Değeri: $2.500 (laptop) + $22.500 (hassas PII verisi, önceki olaylara göre belirlenen yasal maliyetler, itibar kaybı vb.) = **$25.000**.
    · Maruz Kalma Faktörü: Şifrelenmemiş veriye sahip bir laptop çalındığında bu %100'dür.

· SLE: $25.000 × %100 = $25.000

– **Adım 3: Yıllık Meydana Gelme Oranı (Annual Rate of Occurrence - ARO) Hesaplaması:** Bu, yılda kaç kez bu tür bir olayın meydana gelmesinin beklendiğidir.
  * **Örnek Uygulama:** Geçmiş olaylara bakılarak, yılda ortalama 11 laptop hırsızlığı yaşanmışsa, ARO değeri **11**'dir.
– **Adım 4: Yıllık Kayıp Beklentisi (Annual Loss Expectancy - ALE) Hesaplaması:** Bu, bir riskten kaynaklanan yıllık maliyet beklentisidir.
  * **Formül:** ALE = SLE × ARO
  * **Örnek Uygulama:** $25.000 × 11 = $275.000

Aşağıdaki tablo, bu FAIR analiz senaryosunu özetlemektedir:

Bileşen	Tanım	Değer
Varlık	Çalınan dizüstü bilgisayarın değeri, içerdiği hassas PII verisi dahil	25.000$
Tek Kayıp Beklentisi (SLE)	Tek bir olayın maliyeti ($25.000 × %100 maruz kalma faktörü)	25.000$
Yıllık Meydana Gelme Oranı (ARO)	Yılda beklenen olay sayısı	11
Yıllık Kayıp Beklentisi (ALE)	Yıllık toplam beklenen maliyet ($25.000 × 11)	275.000$

Bu tür nicel veriler, soyut bir tehdidi finansal terimlere dökerek, güvenlik yöneticilerinin üst yönetimle aynı dilde konuşmasını sağlar ve bütçe gerekçelendirmesini kolaylaştırır.

10.2.4 Risk Appetite ve Tolerance Definition (Risk İştahı ve Tolerans Tanımı)

Risk iştahı, organizasyonun hedeflerini takip ederken kabul etmeye istekli olduğu genel risk seviyesidir. Risk toleransı ise, belirli bir risk türü için kabul edilebilir varyasyonun daha granüler bir yansımasıdır. Risk toleransı, risk iştahı doğrultusunda, belirli sınırlar ve eşikler belirler.

10.2.5 Risk Treatment Strategies ve Control Selection (Risk Ele Alma Stratejileri ve Kontrol Seçimi)

Risk değerlendirmesi tamamlandıktan sonra, kuruluşlar riskleri yönetmek için stratejiler belirler. Başlıca stratejiler şunlardır:

– **Kaçınma (Avoidance):** Riski tamamen ortadan kaldırmak için ilgili aktiviteyi veya sistemi terk etme. Örneğin, hassas verileri güvensiz bir bulut hizmetinde depolamaktan kaçınmak.
– **Azaltma (Loss Prevention and Reduction):** Kontroller uygulayarak bir olayın oluşma sıklığını veya etkisini azaltma. Bu, en yaygın siber güvenlik stratejisidir. Örneğin, güvenlik duvarları ve şifreleme kullanmak.
– **Transfer Etme (Transfer):** Riski finansal olarak başka bir tarafa devretme. En yaygın örneği siber sigorta almaktır. Bu, riskin finansal yükünü sigorta şirketine aktarır.
– **Kabul Etme (Retention):** Riski bilerek ve isteyerek kabul etme ve sonuçlarına katlanma kararı. Bu, riski yönetmenin maliyetinin, riskin gerçekleşme olasılığı veya etkisinden daha yüksek olduğu durumlarda tercih edilebilir.

10.3 Uyum Yönetimi ve Düzenleyici Çerçeveler

Uyum (compliance), bir organizasyonun yasalara, düzenlemelere ve standartlara uyma eylemidir. Yönetişim içsel hedeflerle ilgiliyken, uyumluluk dışsal gereksinimlere odaklanır.

10.3.1 Regulatory Compliance Assessment ve Gap Analysis (Düzenleyici Uyum Değerlendirmesi ve Açık Analizi)

Açık analizi (gap analysis), bir kuruluşun mevcut politika, prosedür ve uygulamalarını belirli bir düzenleyici çerçeveye (örneğin, GDPR, HIPAA) göre değerlendirerek eksiklikleri belirleme sürecidir.

Adım Adım Gap Analizi:

1. **Kapsamı Tanımlayın:** Hangi düzenlemelere ve standartlara uyum sağlanması gerektiğini net bir şekilde belirleyin.
2. **Mevcut Durumu Gözden Geçirin:** Mevcut politikaları, kontrolleri ve prosedürleri kapsamlı bir şekilde inceleyin. Bu, belgeleme, uygulamalar ve teknolojileri içerir.
3. **Açıkları Belirleyin:** Mevcut durum ile uyum gereklilikleri arasındaki tutarsızlıkları listeleyin. Bu, eksik kontrolleri, güncel olmayan politikaları veya eksik belgeleri içerebilir.
4. **Açıkları Önceliklendirin:** Her açığı, potansiyel etkisi ve risk seviyesine göre sıralayın. Örneğin, müşteri verilerinin şifrelenmemesi, ağır finansal cezalara ve itibar kaybına yol açabileceği için yüksek öncelikli bir açık olabilir.

10.3.2 Internal Audit Programs ve Control Testing (Kurum İçi Denetim Programları ve Kontrol Testi)

Kurum içi denetim programları, siber güvenlik süreçlerinin, politikalarının ve araçlarının etkinliğini değerlendirmek ve potansiyel tehditlere karşı uygun kontrollerin mevcut olduğunu doğrulamak için kritik öneme sahiptir. Bu denetimler, riskleri belirlemeye ve gidermeye yardımcı olurken, aynı zamanda dış denetimlere hazırlık sağlar.

10.3.3 External Audit Coordination ve Remediation Management (Dış Denetim Koordinasyonu ve İyileştirme Yönetimi)

Dış denetimler, üçüncü bir tarafın uyumluluğu doğrulaması için yapılır. Başarılı bir dış denetim, titiz bir hazırlık ve denetim bulgularının etkin bir şekilde giderilmesini gerektirir. Uyum otomasyon araçları, denetim sürecini kolaylaştırarak kanıt toplama ve raporlama yükünü azaltır.

10.3.4 Compliance Automation Tools ve Continuous Monitoring (Uyum Otomasyon Araçları ve Sürekli İzleme)

Sürekli uyum, kuruluşun uyumluluk gerekliliklerini yıl boyunca sürekli olarak izlemesini ve uyumsuzluk sorunlarını ortaya çıktıkça gerçek zamanlı olarak gidermesini sağlayan bir süreçtir. Bu, uyumu tek bir yıllık denetim etkinliğinden sürekli bir sürece dönüştürür.

Bu otomasyon, geleneksel manuel süreçlerin zaman alıcı ve kaynak yoğun doğasını ortadan kaldırır. Otomasyon araçları (SIEM, SOAR), kanıt toplama ve kontrol testini otomatikleştirerek insan müdahalesini %80 oranında azaltabilir. Bu, kuruluşların denetime hazır olma süresini haftalara indirebilir ve proaktif bir duruşa geçişi temsil eder.

Sürekli İzleme İçin Temel Araçlar:

– **Güvenlik Bilgileri ve Olay Yönetimi (SIEM):** SIEM araçları (Splunk, IBM QRadar, SolarWinds Security Event Manager) çeşitli kaynaklardan (sunucular, uygulamalar, ağ cihazları) gelen logları ve olay verilerini toplar, normalleştirir ve ilişkilendirir. Bu, gerçek zamanlı olarak tehditleri ve güvenlik ihlallerini tespit etmeyi sağlar. Örneğin, bir sağlık kuruluşu, HIPAA uyumluluğunu sağlamak için bir SIEM kullanır. SIEM, elektronik sağlık kayıtlarına yetkisiz erişimi izler ve raporlar oluşturur.

- **Siber Güvenlik Orkestrasyonu, Otomasyonu ve Müdahalesi (SOAR):** SOAR platformları, SIEM'den gelen uyarıları alır ve önceden tanımlanmış "playbook"lar (oyun planları) aracılığıyla otomatik veya yarı otomatik müdahale eylemleri başlatır.

Örnek SOAR Playbook (Oltalama Saldırısı Müdahalesi):

1. **Algılama:** SIEM, bir e-posta güvenliği ağ geçidinden şüpheli bir e-posta uyarısı alır.
2. **Sınıflandırma ve Analiz:** SOAR, e-postadaki göstergeleri (URL, IP, dosya hash'leri) otomatik olarak çıkarır ve üçüncü taraf tehdit istihbaratı araçlarıyla karşılaştırır. Bu analiz, e-postanın kötü amaçlı olup olmadığını doğrular.
3. **İçerme (Containment):** Eğer e-posta kötü amaçlı olarak doğrulanırsa, SOAR tüm kullanıcıların gelen kutularındaki şüpheli e-postaları otomatik olarak karantinaya alır ve ilgili etki alanlarını güvenlik duvarı seviyesinde engeller.
4. **Kaldırma (Eradication):** Analiz tamamlandıktan sonra, SOAR tehdidin etkisiz hale getirildiğini doğrular ve ilgili güvenlik olayını kapatır.

10.3.5 Cross-border Compliance ve Data Sovereignty (Sınır Ötesi Uyum ve Veri Egemenliği)

Veri egemenliği, dijital verilerin üretildiği ülkenin yasalarına ve yönetişim yapılarına tabi olduğu ilkesidir. Bu durum, özellikle çok uluslu şirketler için sınır ötesi veri transferlerini karmaşık bir hale getirir.

Yeterlilik Kararları (Adequacy Decisions): Avrupa Komisyonu tarafından verilen bir yeterlilik kararı, AB dışındaki bir ülkenin kişisel veriler için yeterli düzeyde koruma sağladığını resmen onaylar. Bir ülke "yeterlilik" statüsü kazandığında, kişisel veriler AB'den bu ülkeye ek koruyucu önlemler (örneğin, Standart Sözleşme Maddeleri - SCC'ler) olmaksızın serbestçe akabilir. Bu, veri transfer süreçlerini teknik ve operasyonel olarak basitleştirir.

Yeterlilik kararı olmadığında, kuruluşlar veri transferlerini sağlamak için "uygun güvenceler" (appropriate safeguards) kullanmak zorundadır. Bu güvenceler şunları içerir:

- **Standart Sözleşme Maddeleri (SCC'ler):** Bunlar, veri koruma yetkilileri tarafından onaylanmış ve veri transferinde kullanılması gereken yasal metinlerdir. Bu maddeler, transferin teknik olarak nasıl gerçekleştirileceğini (şifreleme, erişim kontrolü vb.) dolaylı olarak etkileyebilir.
- **Bağlayıcı Şirket Kuralları (BCR'ler):** Çok uluslu şirketler için geçerli olan, dahili veri transferlerini yöneten, denetim makamları tarafından onaylanması gereken bağlayıcı kurallardır.

Bu durum, gizlilik mühendisliği kontrollerinin (örneğin, şifreleme) yalnızca iyi bir uygulama değil, aynı zamanda belirli yasal durumlarda zorunlu bir gereklilik haline geldiğini göstermektedir.

10.4 Güvenlik Metrikleri, KPIs ve Performans Ölçümü

Güvenlik metrikleri ve temel performans göstergeleri (KPI'lar), bir siber güvenlik programının etkinliğini ölçmek, tehdit eğilimlerini anlamak ve yatırım kararlarını gerekçelendirmek için kullanılır. Bu veriler, güvenlik durumunun tarihsel bir perspektifini sunarak, zaman içindeki eğilimleri ve değişiklikleri görmeyi sağlar.

10.4.1 Security Performance Indicator Development (Güvenlik Performans Göstergesi Geliştirme)

Güvenlik metrikleri, yatırımları izlemenin ötesine geçerek tehdit modelleri, olay müdahale verimliliği ve sistem zafiyetleri hakkında içgörüler sunar. Bu göstergeler, güvenlik stratejilerinin ne kadar etkili olduğunu anlamak için kritik öneme sahiptir.

- **KPI Örnekleri:**
  - Algılama Süresi (Mean Time to Detect - MTTD) ve Müdahale Süresi (Mean Time to Respond - MTTR).
  - Yamalı ve güncel cihazların yüzdesi.
  - Siber güvenlik farkındalık eğitimi tamamlama oranı.
  - Saldırı girişimlerinin sayısı.

10.4.2 Risk Metrics ve Trend Analysis (Risk Metrikleri ve Trend Analizi)

Risk metrikleri, riskin parasal terimlerle nicelleştirilmesine olanak tanır ve zaman içindeki eğilimleri analiz etmeye yardımcı olur. Bir statik sayıdan ziyade, zaman içindeki ilerlemeyi gösteren bir trend çizgisi, üst yönetim için daha anlamlıdır.

- **Kullanım Alanları:**
  - **Nicel Risk Değerleri:** FAIR metodolojisinden elde edilen Yıllık Kayıp Beklentisi (ALE) gibi metrikler, riski objektif olarak değerlendirmeyi sağlar.
  - **Trend Analizi:** Güvenlik olaylarının sayısı veya bir riskin parasal değeri, düzenli aralıklarla izlenerek bir trend grafiği oluşturulabilir.

10.4.3 Executive Dashboard Design ve Reporting (Yönetici Kontrol Paneli Tasarımı ve Raporlama)

Yöneticilere yönelik bir kontrol paneli, kapsamlı ve teknik verilerde kaybolmadan yüksek seviyeli risk, uyumluluk ve stratejik hedeflere doğru ilerleme hakkında genel bir bakış sunar.

Tasarım İlkeleri:

1. **Hedef Kitleyi Tanıyın:** Yönetim kurulu üyeleri stratejik kararlar alırlar, bu nedenle raporlar finansal etki, risk azaltma ve iş değeri gibi konulara odaklanmalıdır.
2. **Sadelik:** Dashboard, beş ila altı temel "kart" veya bileşenle sınırlı olmalı ve trafik ışığı protokolü (kırmızı, sarı, yeşil) gibi basit görsel ipuçları kullanmalıdır.
3. **Hikaye Anlatma:** Statik veriler yerine, zaman içindeki ilerlemeyi gösteren trend çizgileri kullanılmalıdır. Bu, güvenlik yatırımlarının işe yaradığını gösteren bir hikaye anlatmaya yardımcı olur.

Örnek Dashboard Metrikleri:

- Genel Risk Skoru (Nicel bir değer)
- Politikalara Uyum Seviyesi (Yüksek, Orta, Düşük)
- MTTD ve MTTR Trendleri
- Saldırı Girişimleri Sayısı ve Kaynağı

10.4.4 Security Investment ROI Calculation (Güvenlik Yatırımı ROI Hesaplaması)

Güvenlik yatırımlarının geri dönüşünü (ROSI), geleneksel finansal ROI formüllerinin ötesinde, azaltılan potansiyel kayıplar üzerinden hesaplamak gerekir.

$$ROSI = \frac{(ALE \times \text{Azaltma Oranı}) - \text{Çözüm Maliyeti}}{\text{Çözüm Maliyeti}} \tag{10.1}$$

Örnek Senaryo:

- **Senaryo:** Fidye yazılımı ve DDoS saldırılarına karşı bir Yönetilen Güvenlik Hizmeti Sağlayıcısı (MSSP) çözümü almayı değerlendiren bir kuruluş.
- **Veriler:** Fidye yazılımı ve DDoS için beklenen toplam yıllık kayıp beklentisi (ALE) 41.500.000$. Çözümün bu tehditleri önlemedeki etkinliği (azaltma oranı) %80 ve yıllık maliyeti 120.000$.
- **Hesaplama:**

$$ROSI = \frac{(41,500,000 \times 0.8) - 120,000}{120,000} = 275.67 = 27,567\%$$

Güvenlik yatırımları doğrudan gelir getirmediği için finans ekipleri için değeri genellikle belirsizdir. ROSI formülü, bu soyut yararı (azaltılan risk) somut, parasal bir değere dönüştürür. Yapılan hesaplama, her bir dolarlık güvenlik yatırımının 275 dolar potansiyel kaybı önlediğini gösterir. Bu nicel veri, yöneticilerin bütçe kararlarını gerekçelendirmesine yardımcı olur.

10.4.5 Benchmark Analysis ve Peer Comparison (Kıyaslama Analizi ve Akran Karşılaştırması)

Kıyaslama analizi, bir organizasyonun güvenlik duruşunu sektördeki veya benzer büyüklükteki akranlarına göre değerlendirme sürecidir. Bu, kuruluşun kendi performans hedeflerini belirlemesine ve yatırım yapılacak alanlara odaklanmasına yardımcı olur. Etkin kıyaslama için, yüksek kaliteli veri, ortak bir terminoloji ve karşılaştırılabilir metriklerin kullanımı esastır.

10.5 İş Sürekliliği ve Felaket Kurtarma Planlaması

İş sürekliliği planlaması (BCP), bir kriz anında temel iş süreçlerinin devamlılığını sağlamayı amaçlayan geniş kapsamlı bir yaklaşımdır. Felaket Kurtarma Planı (DRP) ise, BCP'nin teknolojiye odaklanan, veri kurtarma ve IT sistemlerini yeniden çalışır hale getirme üzerine yoğunlaşan bir alt kümesidir.

10.5.1 Business Impact Analysis (BIA) ve Criticality Assessment (İş Etki Analizi ve Kritiklik Değerlendirmesi)

BIA, bir olayın en önemli iş süreçlerini nasıl etkileyeceğini anlamak için kritik bir ilk adımdır. Bu analiz, varlıkların önceliklendirilmesine temel oluşturur ve kaynakların en kritik alanlara ayrılmasını sağlar. Adımlar, bir varlık envanteri oluşturmayı, varlıkları iş fonksiyonlarına göre kritikliklerine göre sıralamayı ve potansiyel etkilerini değerlendirmeyi içerir.

10.5.2 Recovery Time Objective (RTO) ve Recovery Point Objective (RPO) (Kurtarma Süresi ve Kurtarma Noktası Hedefi)

Bu hedefler, bir felaket durumunda kurtarmanın nasıl önceliklendirileceğini ve ne kadar hızlı gerçekleşeceğini belirler.

- **RTO (Recovery Time Objective):** Bir uygulamanın, işi olumsuz etkilemeye başlamadan önce maksimum kapalı kalabileceği süredir.
- **RPO (Recovery Point Objective):** Bir kesinti anında kabul edilebilir maksimum veri kaybı miktarıdır. Bu, veri yedekleme sıklığını belirler.

Aşağıdaki tablo, farklı iş süreçleri için RTO ve RPO hedeflerini kritiklik seviyelerine göre sınıflandırmaktadır:

İş Kritikliği Seviyesi	Örnek İş Süreci	RTO (Kurtarma Süresi)	RPO (Kurtarı
Görev-Kritik (Mission-Critical)	Müşteri Sipariş Sistemi, Finansal İşlemler	Saniyelerden Dakikalara	0-15 dakika
Temel (Essential)	E-posta, İK Uygulamaları	4 ila 24 saat	1-24 saat
Kritik Olmayan (Nonessential)	İç Raporlama Araçları, İdari Dosyalama	24 saatten Haftalara	24 saatten haf

Bu matris, IT ve iş ekiplerinin kurtarma öncelikleri üzerinde ortak bir anlayışa sahip olmasını sağlar, böylece kaynaklar en acil ihtiyaçlara yönlendirilebilir.

10.5.3 Disaster Recovery Planning ve Testing (Felaket Kurtarma Planlaması ve Testi)

DRP, siber saldırı, doğal afet veya sistem arızası gibi olaylara müdahale etmek için stratejiler ve protokoller sağlayan bir yol haritasıdır. Bir DRP'nin var olması yeterli değildir; planın gerçek bir kriz anında işe yaradığından emin olmak için düzenli olarak test edilmesi ve güncellenmesi gerekir.

Plan Oluşturma Adımları:

1. **Ekip Kurma:** Kriz anında müdahale çabalarına liderlik edecek bir afet müdahale ekibi oluşturun.
2. **Altyapı Planı:** Ağ altyapınızın ve sistem bağımlılıklarının detaylı bir planını çizin.
3. **Kurtarma Prosedürlerini Belgeleyin:** Hasarlı sistemleri, uygulamaları ve verileri kurtarmak için adım adım talimatları basit bir dille yazın. Bu plan ağdan uzak bir yerde veya değişmez (immutable) depolama alanında saklanmalıdır.

Test ve Tatbikat Türleri:

- **Masaüstü Tatbikatları (Tabletop Exercises):** Ekip, bir senaryoyu (örneğin, fidye yazılımı saldırısı) sözlü olarak tartışır, rolleri ve prosedürleri gözden geçirir.
- **Simülasyonlar:** Gerçek bir kesintiyi simüle ederek, sistemlerin kurtarma kabiliyetini test eder. Bu testler, planın zayıf yönlerini ortaya çıkarır ve bu eksiklikler, gerçek bir olaydan önce giderilebilir.

10.5.4 Crisis Management ve Emergency Response (Kriz Yönetimi ve Acil Durum Müdahale)

Bir siber saldırı, özellikle fidye yazılımı (ransomware), özel bir müdahale planı gerektirir. Aktörlerin sizi izleyebileceği düşünülerek, müdahale işlemleri koordineli bir şekilde ve bant dışı (out-of-band) iletişim kanalları (telefon görüşmeleri) kullanılarak yapılmalıdır.

Fidye Yazılımı Müdahale ve Kurtarma Planı (Teknik Playbook):

1. **İçerme (Containment):** Ağdaki enfekte sistemleri hemen belirleyin ve yalıtın.
   - **Teknik Adımlar:** Etkilenen sistemlerin ağ kablosunu çekin veya kablosuz bağlantısını kesin. Enfeksiyonun yayılmasını engellemek için anahtar (switch) seviyesinde ağı kapatmak en etkili yöntem olabilir.
2. **Delil Toplama:** Forensik inceleme için etkilenen sistemlerin imajını alın ve uçucu hafıza içeriğini (volatile memory) yakalayın.
3. **Temizleme (Eradication):** Tüm enfekte sistemleri silin veya dezenfekte edin.
4. **Kurtarma ve Yeniden İnşa:** Yedeklerden geri yüklemeye başlayın.

```
--Veritabanı Geri Yükleme (SQL Server Örneği):
RESTORE DATABASE [veritabani_adi] FROM DISK = '[yedek_dosyasi_yolu]' WITH RECOVERY

--Anlık Görüntü (Snapshot) Geri Yükleme (AWS RDS Örneği):
aws rds restore-db-cluster-from-snapshot --db-cluster-identifier my-db-cluster --snapshot-identi

--Dosya Geri Yükleme (Linux Örneği):
restore -xvqf /dev/rmt0 /home/mike/tools
```

10.5.5 Supply Chain Continuity ve Vendor Dependency Management (Tedarik Zinciri Sürekliliği ve Tedarikçi Bağımlılık Yönetimi)

İşletmelerin giderek daha karmaşık hale gelen tedarik zincirlerine bağımlı olması, bu zincirdeki kesintilere karşı dayanıklılık oluşturmayı zorunlu hale getirir. Bu dayanıklılığı artırmak için şu stratejiler benimsenmelidir:

- **Proaktif Tanımlama:** Kritik vendor bağımlılıklarını önceden belirleyin ve tek kaynaklı tedarik anlaşmalarının potansiyel tehlikelerini anlayın.
- **Alternatif Kaynak Geliştirme:** Kriz anında hızlı geçişi mümkün kılmak için önceden nitelikli, alternatif tedarikçilerle ilişkiler kurun.
- **SLA ve Sözleşme Güçlendirme:** Kesinti anında tedarikçi sorumluluklarını netleştiren güçlü sözleşmeler hazırlayın.

10.6 Gizlilik Yönetimi ve Veri Koruma Yönetişimi

Gizlilik yönetimi, özellikle kişisel verilerin korunmasına odaklanan GRC'nin kritik bir bileşenidir.

10.6.1 Privacy by Design ve Privacy Impact Assessments (Tasarım Yoluyla Gizlilik ve Gizlilik Etki Değerlendirmeleri)

Tasarım yoluyla gizlilik (Privacy by Design), gizliliği bir sistemin veya projenin başlangıç tasarımından itibaren entegre etme ilkesidir. Gizlilik Etki Değerlendirmesi (PIA), bu ilkenin pratik bir uygulamasıdır. PIA, özellikle kişisel verilerin işlenmesi yüksek risk taşıyorsa veya yeni teknolojiler kullanılıyorsa zorunludur.

10.6.2 Data Protection Officer (DPO) Role ve Responsibilities (Veri Koruma Görevlisi Rolü ve Sorumlulukları)

Veri Koruma Görevlisi (DPO), kuruluşun veri koruma kurallarına uyumunu sağlamaktan sorumlu kilit bir pozisyondur.

- **Temel Sorumlulukları:** Yönetime ve çalışanlara veri koruma yükümlülükleri hakkında danışmanlık yapmak, uyumluluğu izlemek ve veri sahiplerinden gelen talepler için iletişim noktası olmak.
- **DPO'nun Konumu:** DPO, görevlerini bağımsız bir şekilde yerine getirebilmeli ve yönetim çatışmasından kaçınmak için diğer görevlerle çelişmemelidir.

10.6.3 Data Subject Rights Management ve Breach Notification (Veri Sahibi Hakları Yönetimi ve İhlal Bildirimi)

Veri sahipleri, kişisel verileri üzerinde belirli haklara sahiptir. Kuruluşlar, bu hakları yönetmek için süreçler kurmalıdır. Kişisel veri ihlali durumunda, kontrolör, mümkünse ihlalin farkına vardıktan sonra en geç 72 saat içinde yetkili denetim makamına bildirimde bulunmalıdır. Eğer ihlal veri sahipleri için yüksek risk oluşturuyorsa, doğrudan bildirim zorunludur.

10.6.4 Cross-border Data Transfer ve Adequacy Decisions (Sınır Ötesi Veri Transferi ve Yeterlilik Kararları)

Bir yeterlilik kararı, veri egemenliği sorunlarını azaltarak teknik akışı kolaylaştıran bir yasal mekanizma sağlar. Yeterlilik kararı, veri transferi mekanizmaları üzerindeki teknik yükü doğrudan etkiler. Bu karar olmadığında,

kuruluşlar daha karmaşık mekanizmalar kullanmak zorundadır. Bu, Standart Sözleşme Maddeleri (SCC'ler) veya Bağlayıcı Şirket Kuralları (BCR'ler) gibi "uygun güvenceler"in kullanılmasını gerektirir. Bu durum, gizlilik mühendisliği kontrollerinin (örneğin, şifreleme) sadece iyi bir uygulama değil, aynı zamanda belirli yasal durumlarda zorunlu bir gereklilik haline geldiğini gösterir.

10.6.5 Privacy Engineering ve Technical Controls Implementation (Gizlilik Mühendisliği ve Teknik Kontroller Uygulaması)

Gizlilik mühendisliği, kişisel verileri işleyen sistemlerin ve hizmetlerin tasarımına gizlilik ilkelerini dahil eden bir disiplindir. Bu, veri anonimleştirme ve şifreleme gibi teknikleri içerir.

Veri Anonimleştirme ve Gizlilik Koruma Teknikleri:

- **K-Anonimlik:** Bir veri setinde, her bir birey için en az k-1 diğer bireyin verileriyle ayırt edilemez olmasını sağlayan bir özelliktir. Örneğin, bir hastane veritabanında, yaş ve cinsiyet gibi yarı tanımlayıcı (quasi-identifier) veriler genelleştirilerek (örneğin, 20-30 yaş arası erkek), saldırganın bir bireyi diğerlerinden ayırt etmesi engellenir. Bu tekniğin, saldırganın arka plan bilgisi varsa başarısız olabileceği bir sınırlılığı bulunmaktadır.
- **Diferansiyel Gizlilik (Differential Privacy):** Bir veri setine istatistiksel sorgular yapılırken, bireysel bir kaydın varlığının veya yokluğunun sorgu sonucunu önemli ölçüde etkilemesini önlemek için dikkatli bir şekilde gürültü (noise) ekleme tekniğidir. Bu teknik, özellikle büyük veri setlerinin analizi sırasında her bir bireyin gizliliğini korumak için kullanılır.

Aşağıdaki tablo, farklı veri anonimleştirme ve maskeleme tekniklerini karşılaştırmaktadır:

Teknik	Tanım
Maskeleme	Gerçek verilerin yerine farklı, ancak kullanışlı veriler koyma
Anonimleştirme	Verileri, başka verilerle eşleştirilerek dahi hiçbir şekilde belirli bir kişiyle ilişkilendirilemeyecek ha
K-Anonimlik	Bir veri setindeki her bir kaydın, en az k diğer kayıtla ayırt edilemez olmasını sağlama
Pseudonymization	Özel tanımlayıcıların yerine takma adlar veya sahte tanımlayıcılar kullanma
Hashing	Veriyi tek yönlü bir algoritmaya dönüştürerek orijinal veriye geri dönülmesini engelleme

Bölüm 11

SIZMA TESTİ VE ETİK HACKING

Giriş

Sızma testi ve etik hacking, organizasyonların güvenlik açıklarını proaktif olarak tespit etmek için kullanılan kritik metodolojilerdir. Bu bölümde sızma testi çerçeveleri, etik hacking teknikleri ve güvenlik değerlendirme süreçlerini detaylı olarak ele alacağız.

11.1 Sızma Testi Çerçeveleri ve Metodolojileri

Sızma testi (penetration testing), bir bilgisayar sisteminin, ağın veya web uygulamasının güvenlik açıklarını belirlemek ve değerlendirmek için yetkili bir simüle edilmiş siber saldırıdır. Bu süreç, bir saldırganın bakış açısını benimseyerek, potansiyel güvenlik zafiyetlerini istismar etmeye çalışır. Sızma testleri, bir kuruluşun güvenlik duruşunu proaktif bir şekilde değerlendirmesine ve savunma mekanizmalarını güçlendirmesine olanak tanır.

11.1.1 OWASP Testing Guide Uygulamaları ve Kapsamı

OWASP (Open Web Application Security Project), özellikle web uygulamaları ve hizmetlerinin güvenliği için dünya çapında kabul görmüş, kar amacı gütmeyen, açık kaynaklı bir organizasyondur. OWASP Testing Guide, web uygulamalarından mobil uygulamalara, API'lerden IoT cihazlarına kadar geniş bir yelpazedeki güvenlik testlerini kapsayan kapsamlı bir çerçeve sunar. Bu kılavuz, yalnızca teknik zafiyetleri değil, aynı zamanda güvenli olmayan geliştirme pratiklerinden kaynaklanan karmaşık mantık hatalarını da tespit etmeye odaklanır. OWASP metodolojisi, kullanıcıların kendi organizasyonlarında uygulayabilecekleri en iyi pratikleri içeren bir çerçeve sunar. Aynı zamanda, en yaygın web uygulama ve web hizmeti güvenlik sorunlarını test etmek için alt seviye, pratik teknikleri de detaylandırır. Bu, onu hem yüksek seviyeli bir yol haritası hem de düşük seviyeli bir test kılavuzu haline getirir.

11.1.2 PTES (Penetration Testing Execution Standard) Süreci

PTES, bilgi güvenliği uzmanlarından oluşan bir ekip tarafından, sızma testinin ilk iletişiminden test sonrası raporlamaya kadar olan her aşamayı kapsayan kapsamlı ve güncel bir standart olarak oluşturulmuştur. Bu standart, test uzmanlarına yol gösterirken, müşterilere de bir sızma testinden ne beklemesi gerektiği konusunda net bir çerçeve sunar. PTES, yedi ana aşamadan oluşur ve her bir aşama, bir sızma testinin başarısı için kritik öneme sahiptir.

1. **Pre-Engagement Interactions (Test Öncesi Etkileşimler):** Bu aşama, herhangi bir teknik test başlamadan önce gerçekleşir. Testin kapsamı, tahmini bütçe ve zaman çizelgesi gibi konular bu aşamada netleştirilir. Ayrıca, acil durum iletişim kanalları, delil toplama prosedürleri ve yasal izinler (*permission to test*) gibi kurallar (*Rules of Engagement*) belirlenir.
2. **Intelligence Gathering (Bilgi Toplama):** Bu aşama, hedef hakkında mümkün olduğunca fazla bilgi edinmeye odaklanır. Süreç, pasif (OSINT) ve aktif keşif yöntemlerini içerir. Pasif bilgi toplama, arama motorları ve halka açık veritabanları gibi üçüncü taraf kaynaklardan veri toplanmasını; aktif keşif ise doğrudan hedef sistemle etkileşim kurarak bilgi edinilmesini içerir.
3. **Threat Modeling (Tehdit Modelleme):** Bu aşama, iş varlıklarının ve süreçlerinin tanımlanmasıyla başlar. Potansiyel tehdit aktörleri (içeriden veya dışarıdan) ve bunların yetenekleri analiz edilerek, saldırganların sisteme nasıl sızabileceğine dair gerçekçi senaryolar oluşturulur.
4. **Vulnerability Analysis (Zafiyet Analizi):** Bu aşamada, hedef sistemlerdeki zayıflıklar ve güvenlik açıkları belirlenir. Bu analiz, otomatik zafiyet tarayıcılarının kullanımıyla aktif değerlendirme veya trafik izleme yoluyla pasif değerlendirme gibi çeşitli yöntemlerle gerçekleştirilir. Bu süreç, istismar edilebilecek potansiyel saldırı vektörlerinin belirlenmesiyle sonuçlanır.
5. **Exploitation (İstismar):** Bu, tespit edilen zafiyetlerin kullanılarak sisteme erişim sağlandığı aşamadır. Amaç, en az direnç yolunu bularak ve tespit edilmekten kaçınarak organizasyonun varlıklarına erişmektir.
6. **Post-Exploitation (İstismar Sonrası):** Bir sisteme ilk erişim sağlandıktan sonra, bu aşama daha derin bir kontrol elde etmeye odaklanır. Tester, sistemin değerini belirler, ayrıcalık yükseltme, yanal hareket ve veri sızdırma gibi eylemler gerçekleştirir. Bu aşamanın sonunda, elde edilen bulgular raporlama için belgelenir.
7. **Reporting (Raporlama):** Sızma testinin son aşamasıdır. Rapor, iki ana bölümden oluşur: Yöneticilere yönelik iş etkisini özetleyen bir Yönetici Özeti ve teknik personele yönelik detaylı bulgular, saldırı yolu ve iyileştirme önerilerini içeren bir Teknik Rapor.

11.1.3 OSSTMM (Open Source Security Testing Methodology Manual) Yaklaşımı

OSSTMM, operasyonel güvenliği bilimsel bir yaklaşımla ölçen, hakemli ve açık kaynaklı bir metodoloji kılavuzudur. Bu çerçeve, güvenliğin sadece teknolojik mekanizmalara bağlı olmadığını, aynı zamanda insan, fiziksel, telekomünikasyon ve süreç güvenliğini de kapsayan bütüncül bir yaklaşımı vurgular. OSSTMM, güvenlik kontrollerinin varlığını değil, bunların yokluğunu ölçerek istismar edilebilir zafiyetleri belirlemeye odaklanır. Metodoloji, güvenlik testini beş ana alanda ele alır: bilgi güvenliği, süreç güvenliği, internet teknolojisi güvenliği, iletişim güvenliği ve fiziksel güvenlik. Ayrıca, test sonuçlarını standartlaştırmak ve operasyonel güvenliği ölçmek için RAV (Risk Assessment Values) Hesaplayıcı ve STAR (Security Test Audit Report) gibi özel araçlar içerir.

11.1.4 NIST SP 800-115 Technical Guide to Information Security Testing

NIST SP 800-115, kuruluşlara teknik bilgi güvenliği test ve değerlendirmelerini planlama, yürütme ve bulguları analiz etme konusunda pratik tavsiyeler sunan bir rehberdir. Kılavuz, kapsamlı bir güvenlik programı olmaktan ziyade, belirli tekniklere, bunların faydalarına, sınırlılıklarına ve nasıl kullanılacağına odaklanır. Yapılandırılmış bir yaklaşımı vurgulayarak, planlama, yürütme ve yürütme sonrası analizden oluşan metodik bir süreci teşvik eder.

11.1.5 Sızma Testi Kapsam Belirleme (Scoping) ve Etkileşim Kuralları (Rules of Engagement)

Bir sızma testinin başarısı, net ve açık bir şekilde tanımlanmış bir kapsama bağlıdır. Kapsam, testin hangi sistemler, ağlar, uygulamalar ve bileşenler üzerinde gerçekleştirileceğini belirleyen bir yol haritasıdır. Bu belirleme, testin alakasız sistemlere kaynak israfını önler ve operasyonel kesinti riskini minimize eder.

- **Kapsam (Scoping):** Kapsam belgesi, *kapsam içi* ve *kapsam dışı* varlıkları net bir şekilde listeler. Kapsam içi varlıklar, test için açıkça yetkilendirilmiş sistemleri, IP adreslerini, alan adlarını ve hizmetleri içerirken, kapsam dışı varlıklar, potansiyel operasyonel etki veya yasal kısıtlamalar nedeniyle testin dışında bırakılan sistemlerdir (örneğin, üçüncü taraf platformlar veya yüksek çalışma süresi gerektiren üretim sistemleri). Kapsam ayrıca testin türünü de belirler:
  - **Black-Box:** Test uzmanı, hedef hakkında önceden bilgi sahibi değildir. Bu, dışarıdan bir saldırganı taklit eder.
  - **White-Box:** Test uzmanına kaynak kodları, ağ diyagramları ve kimlik bilgileri gibi tüm bilgiler verilir. Bu, derinlemesine kod analizi ve sistem mimarisi değerlendirmesi için idealdir.
  - **Gray-Box:** Test uzmanına sınırlı düzeyde dahili bilgi (örneğin, standart bir kullanıcı hesabı) sağlanır. Bu yaklaşım, içeriden bir tehdidi veya ele geçirilmiş kimlik bilgilerine sahip bir saldırganı simüle eder.
- **Etkileşim Kuralları (Rules of Engagement - ROE):** ROE, sızma testi projesinin "yapılacaklar ve yapılmayacaklar" listesini detaylandıran kritik bir belgedir. ROE, testin zaman çizelgesini, acil durum iletişim bilgilerini, hassas verilerin nasıl işleneceğini ve hangi tekniklerin (örneğin, Hizmet Reddi - DoS saldırıları) kullanılabileceğini belirler. Bu belge, hem müşteriyi hem de test uzmanını koruyan yasal bir temel sağlar.

Sızma testi metodolojileri arasındaki en önemli ilişki, her birinin farklı bir amaca hizmet etmesidir. PTES, sızma testinin operasyonel aşamalarını organize ederken, OWASP web uygulaması testine odaklanır, OSSTMM güvenliğin bütüncül bir ölçümünü sunar ve NIST uyumluluk odaklı bir rehberlik sağlar. Bu çerçevelerin birleştirilmesi, statik bir süreçten ziyade, amaca göre uyarlanabilen dinamik bir disiplinin temelini oluşturur. Örneğin, PTES'in genel aşamalarını izleyen bir proje, web uygulaması testi için OWASP'tan yararlanırken, operasyonel güvenlik ölçümü için OSSTMM'nin bilimsel yaklaşımını benimseyebilir. Bu yaklaşım, yalnızca teknik zafiyetleri bulmakla kalmaz, aynı zamanda iş süreçlerindeki, fiziksel çevredeki ve insan faktöründeki zayıflıkları da kapsar. Sızma testinin başarısı, teknik becerilerin yanı sıra, kapsam belirleme ve etkileşim kuralları gibi hukuki ve etik altyapının da ne kadar sağlam olduğuna bağlıdır. Sızma testinin bu yönleri, etik hackerı kötü niyetli bir hackerdan ayıran temel unsurlardır.

Metodoloji	Amacı	Kapsamı
OWASP	Web ve uygulama güvenliği açıklarını belirlemek.	Web ve mobil uygulamalar, API'ler, IoT.
PTES	Sızma testinin tüm aşamalarını standartlaştırmak.	Pre-engagement'tan raporlamaya tüm test süreci
OSSTMM	Operasyonel güvenliği bilimsel olarak ölçmek.	Bilgi, süreç, fiziksel, internet ve iletişim güvenl
NIST SP 800-115	Teknik güvenlik testleri için rehberlik sağlamak.	Planlama, yürütme ve analiz süreçleri.

11.2 Bilgi Toplama ve Keşif

Bilgi toplama ve keşif (reconnaissance), bir sızma testinin ilk ve en kritik aşamasıdır. Bu aşamada, hedef sistem hakkında mümkün olduğunca fazla bilgi toplanır. Bu bilgiler, hedef kuruluşun ağ altyapısı, çalışanları, kullandığı teknolojiler ve iş süreçleri hakkında olabilir. Bilgi toplama, pasif ve aktif olmak üzere iki ana kategoriye ayrılır.

11.2.1 Pasif Bilgi Toplama ve OSINT Teknikleri

Pasif bilgi toplama, hedefe doğrudan bir etkileşimde bulunmadan, halka açık kaynaklardan veri toplanmasıdır. Bu yöntem, saldırganın tespit edilme riskini en aza indirdiği için (*stealthy*) tercih edilir.

- **Google Dorking:** Arama motorlarının gelişmiş operatörlerinin (*dorks*) kullanılmasıyla, dizin listelemeleri, hassas dosyalar (`filetype:pdf`), giriş sayfaları (`inurl:login`) ve diğer hassas bilgiler gibi halka açık ancak kolayca bulunamayan veriler ortaya çıkarılabilir.
- **WHOIS Aramaları:** Bir alan adının sahiplik bilgilerini, kayıt tarihini ve iletişim detaylarını öğrenmek için WHOIS veritabanı sorgulanabilir.

- **Halka Açık Veritabanları ve Web Siteleri:** Hükümet kayıtları veya şirket web siteleri, bir kuruluşun geçmişi, finansal durumu, çalışan listeleri ve hatta kullanılan yazılımlar hakkında değerli bilgiler sağlayabilir.
- **OSINT Framework:** Bu, IP adresleri, kullanıcı adları, e-posta adresleri gibi çeşitli veri türlerine göre kategorize edilmiş, açık kaynaklı istihbarat araçlarının kapsamlı bir dizinidir.
- **Araçlar:**
  - **Maltego:** Farklı veri noktaları (kişiler, alan adları, web sayfaları) arasındaki karmaşık ilişkileri görsel olarak haritalayan ve analiz eden bir veri madenciliği aracıdır.
  - **Shodan:** İnternete bağlı cihazları, sunucuları ve diğer sistemleri tarayan bir arama motorudur. *Hackerlar için Google* olarak anılır ve yanlış yapılandırılmış veya güvensiz cihazları bulmada kullanılır.
  - **The Harvester ve Recon-ng:** Bu araçlar, e-posta adresleri, alt alan adları ve ana bilgisayar isimleri gibi bilgileri halka açık kaynaklardan otomatik olarak toplar.

11.2.2 Aktif Keşif ve Ağ Numaralandırma

Aktif keşif, hedefe doğrudan sorgu göndererek bilgi edinme yöntemidir ve bu, hedef sistemde uyarıları tetikleme riski taşır. Bu aşama, ağ yapısı ve kaynakları hakkında daha kesin bilgiler elde etmek için kullanılır.

- **Port ve Hizmet Taraması:** Bir sistemin hangi portlarının açık olduğunu ve bu portlarda hangi hizmetlerin çalıştığını belirlemek, potansiyel zafiyetleri ortaya çıkarır.
- **Banner Yakalama:** Ağ hizmetlerinin versiyon bilgileri gibi detayları gösteren *banner* mesajlarının toplanmasıdır.
- **Uygulama Parmak İzi (Fingerprinting):** Bir web uygulamasının kullandığı sunucu yazılımı, betik dili ve işletim sistemi gibi bilgilerin doğrudan sorgulanmasıdır.
- **Araçlar ve Komut Örnekleri:**
  - **Nmap (Network Mapper):** Ağ keşfi ve güvenlik denetimi için birincil araçtır.

    ```
    $ nmap -sn 192.168.1.0/24
    $ nmap -sV <hedef_IP>
    $ nmap -A <hedef_IP>
    ```

  - **Gobuster & Dirb:** Web sunucularındaki gizli dizinleri ve dosyaları kaba kuvvetle bulmak için kullanılır.

    ```
    $ gobuster dir -u http://<hedef_IP> -w /usr/share/wordlists/dirbuster/directory-list-2.3
    ```

11.2.3 Sosyal Medya İstihbaratı (SOCMINT) Toplama Yöntemleri

Sosyal medya istihbaratı (SOCMINT), açık kaynak istihbaratının (OSINT) bir alt dalıdır ve sosyal medya platformlarından bilgi toplamaya odaklanır. SOCMINT, hem bireyler hem de kuruluşlar hakkında değerli bilgiler sağlayarak, özellikle sosyal mühendislik saldırıları için temel oluşturur.

- **Profil ve Etkileşim Analizi:** Kullanıcıların halka açık profil bilgileri (iş unvanları, konumlar) ve platform içi etkileşimleri (yorumlar, beğeniler, paylaşımlar) incelenerek ilişkiler ve bağlantılar haritalandırılır.
- **Metadata Toplama:** Sosyal medya paylaşımlarındaki fotoğraflar ve videoların metadata'sı (EXIF verileri) incelenerek coğrafi konum bilgileri veya diğer hassas veriler elde edilebilir.
- **Gelişmiş Arama:** Hashtag (#), kullanıcı adı (@) ve belirli anahtar kelimelerle platform içi arama yapılarak, istenen konulardaki konuşmalar ve kullanıcılar hedeflenir.
- **Araçlar:** Sherlock, Maigret ve SpiderFoot gibi araçlar, sosyal medya profilleri ve ilgili verileri otomatize bir şekilde toplamada kullanılır.

11.2.4 DNS Numaralandırma ve Alt Alan Adı Keşfi

DNS numaralandırması, bir hedef alan adıyla ilişkili DNS kayıtlarını sistematik olarak toplayarak potansiyel saldırı vektörlerini belirleme sürecidir.

- **Alt Alan Adı Keşfi:** Bir kuruluşun genişletilmiş ağ yüzeyini anlamak için kritik bir adımdır. Bu, arama motoru operatörleri (`site:*.domain.com`), çevrimiçi hizmetler (DNSdumpster) veya araçlar (OWASP Amass, DNSRecon) kullanılarak gerçekleştirilebilir.
- **Ters DNS Araması (Reverse DNS Lookup):** Bir IP adresini, DNS'deki Pointer (PTR) kayıtlarını kullanarak alan adına geri çözümleme işlemidir. Bu, standart DNS numaralandırma teknikleriyle kolayca bulunamayan ana bilgisayar adlarını ortaya çıkarır.
- **Bölge Transferi İstismarı (Zone Transfer Exploitation):** DNS kayıtlarını sunucular arasında çoğaltmak için tasarlanan bölge transferi, yanlış yapılandırıldığında tüm bölge dosyasının sızdırılmasına neden olabilir. Saldırganlar, `dig` veya `nslookup` gibi araçlarla bu zafiyeti istismar ederek tüm ana bilgisayar adlarını ve IP adreslerini ele geçirebilir.
- **Araçlar ve Komut Örnekleri:**

```
$ dig example.com MX
$ nslookup -type =any example.com
$ dig @ns1.example.com example.com axfr
```

11.2.5 Arama Motoru ve Genel Veritabanı Madenciliği

Bu teknik, halka açık kaynaklardan bilgi toplamanın temelini oluşturur ve yalnızca web siteleriyle sınırlı değildir. Arama motorları, siber güvenlik bağlamında, hassas veritabanlarını veya güvenlik zafiyetleri olan sistemleri bulmak için bir araç olarak kullanılır. Örneğin, *Shodan* gibi özel arama motorları, internete bağlı cihazların açık portlarını, kullanılan hizmetleri ve coğrafi konumlarını listeler. Benzer şekilde, `grep.app` veya `SourceGraph` gibi kod arama motorları, açık kaynaklı kod depolarında hassas bilgileri (örneğin API anahtarları) aramak için kullanılabilir. Bilgi toplama süreci, doğrusal bir süreç değil, sürekli bir geri besleme döngüsüdür. Pasif keşif, saldırganın ilk temas noktalarını belirlerken, aktif keşif bu temas noktalarındaki zafiyetleri ve iç ağ yapısını daha derinlemesine anlamak için kullanılır. Deneyimli bir test uzmanı, tespitten kaçınmak için bu iki yaklaşım arasındaki dengeyi hassas bir şekilde yönetir. OSINT'ten elde edilen veriler (çalışanların isimleri, hobileri, konumu gibi), sosyal mühendislik saldırıları için gerçekçi senaryolar oluşturmanın temelini oluşturur. Bu, teknik zekâ ile sosyal zekânın birleştiği noktayı işaret eder. İstihbarat, tehdit modelleme ve istismar aşamaları için temel oluşturur; bu olmadan, sonraki adımlar verimsiz ve rastgele bir şekilde gerçekleştirilir.

11.3 Zafiyet Değerlendirmesi ve İstismar

Zafiyet değerlendirmesi (vulnerability assessment), bir sistemdeki veya uygulamadaki potansiyel güvenlik açıklarını belirlemek için otomatik araçlar ve manuel teknikler kullanılarak yapılan bir süreçtir. Bu aşamada, önceki bilgi toplama aşamasında elde edilen bilgiler kullanılarak, hedef sistemdeki zafiyetler taranır ve analiz edilir. İstismar (exploitation) ise, belirlenen zafiyetlerin kullanılarak sisteme yetkisiz erişim sağlanması veya kontrolün ele geçirilmesidir.

11.3.1 Otomatik Zafiyet Taraması ve Manuel Doğrulama

Zafiyet değerlendirmesi, iki ana yaklaşımı içerir: otomatik tarama ve manuel doğrulama. Her iki yöntemin de kendine özgü avantajları ve dezavantajları bulunur.

- **Otomatik Tarama:** Bu yaklaşım, yazılım araçlarını kullanarak bir uygulama veya ağdaki bilinen zafiyetleri hızlı bir şekilde tarar. Otomatik araçların en büyük avantajları hız ve geniş kapsamdır. Binlerce sunucuyu, web uygulamasını veya cihazı kısa sürede tarayabilirler. Ancak, bu araçlar bağlamsal anlayıştan yoksundur ve karmaşık iş mantığı zafiyetlerini veya çok adımlı saldırı zincirlerini genellikle gözden kaçırır.
- **Manuel Doğrulama:** Manuel test, yetenekli güvenlik uzmanları tarafından elle yürütülen, uygulamalı bir yaklaşımdır. Manuel testler, otomatik araçların kaçırdığı nüanslı yapılandırma sorunlarını, karmaşık iş mantığı hatalarını ve benzersiz güvenlik açıklarını tespit etmede hayati öneme sahiptir. Bu yaklaşım, bir saldırganın yaratıcılığını ve uyum sağlama yeteneğini taklit eder. Dezavantajı ise, zaman ve kaynak yoğun olmasıdır.

En etkili sızma testi, otomatik taramanın hızını ve geniş kapsamını, manuel doğrulamanın derinliği ve bağlamsal zekasıyla birleştiren hibrit bir model benimser. Otomatik tarayıcılar, tekrarlayan ve bilinen zafiyetleri bulmada son derece etkili olsa da, gerçek bir saldırganın yaratıcılığından yoksundurlar. En tehlikeli zafiyetler (iş mantığı hataları, zincirleme saldırılar) ancak insan zekası ve bağlamsal anlayış ile bulunabilir. Bu, siber güvenlik profesyonellerinin rolünün, sadece bir aracı çalıştırmaktan ziyade, bu araçların bulgularını yorumlamak ve daha derinlemesine analizler yapmak olduğunu gösterir.

Özellik	**Otomatik Zafiyet Taraması**	**Manuel Doğrulama**
Hız	Yüksek, dakikalar veya saatler içinde.	Düşük, günler veya haftalar sürebilir.
Kapsam	Geniş, birçok sistemi aynı anda tarar.	Dar, belirli sistemlere odaklanır.
Derinlik	Yüzeysel, bilinen zafiyetleri arar.	Derin, karmaşık zafiyetleri ve iş mantığı hatalarını
Maliyet	Genellikle lisans ücretleriyle daha düşüktür.	Uzman personelin emeği nedeniyle daha yüksektir.
Bulunan Zafiyetler	Standart teknik zafiyetler (SQLi, XSS).	İş mantığı hataları, zincirleme saldırılar, sıfır gün za
Gereken Beceri	Düşük, araç bilgisi yeterlidir.	Yüksek, derin teknik bilgi ve yaratıcılık gerektirir.

11.3.2 Exploit Geliştirme ve Proof-of-Concept (PoC) Oluşturma

Exploit geliştirme, bir yazılım zafiyetinden yararlanmak için özel kod (*exploit*) oluşturma sürecidir. Bu, güvenlik araştırmacıları ve etik hackerlar için zafiyetlerin nasıl ortaya çıktığını ve nasıl istismar edilebileceğini anlamak için kritik bir beceridir.

- **Temel Teknikler:**
  - **Buffer Overflow (Arabellek Taşması):** Verinin bir arabellek sınırını aşarak bitişik bellek konumlarını ezmesiyle oluşan, rastgele kod yürütülmesine yol açabilen eski ve yaygın bir tekniktir.
  - **Return-Oriented Programming (ROP):** Bellekteki mevcut kod parçalarını (*gadgets*) zincirleyerek, güvenlik korumalarını (*DEP*, *ASLR*) aşmaya ve rastgele kod yürütmeye olanak tanıyan sofistike bir tekniktir.
  - **Heap Exploitation (Yığın İstismarı):** Yığın (*heap*) bellek yapısını hedef alan, veri bozulmasına veya rastgele kod yürütülmesine yol açabilen bir tekniktir.
- **Proof-of-Concept (PoC) Oluşturma:** Bir PoC, bir zafiyetin gerçek ve istismar edilebilir olduğunu kanıtlayan, işlevsel ancak genellikle tam teşekküllü bir saldırı aracı olmayan bir kod veya senaryodur. PoC, bir zafiyet raporunun en önemli parçasıdır, çünkü teorik bir riskin pratik bir tehdit olduğunu kanıtlar ve iyileştirme sürecini kolaylaştırır. Exploit geliştirme, CVE (Common Vulnerabilities and Exposures) sistemiyle yakından ilişkilidir; CVE'ler, bilinen zafiyetler için evrensel bir referans sunarak, araştırmacılar, satıcılar ve savunmacılar arasında iletişimi ve önceliklendirmeyi kolaylaştırır.

11.3.3 Web Uygulaması Sızma Testi Teknikleri

Web uygulaması sızma testi, bir web uygulamasının güvenlik zayıflıklarını bulmak için metodolojik bir dizi adımı içerir. Süreç, bilgi toplama ile başlar, ardından araştırma ve istismar aşamalarına geçilir. En yaygın zafiyetler arasında SQL Enjeksiyonu, Cross-Site Scripting (XSS), kırık kimlik doğrulama ve güvenli olmayan dosya yükleme mekanizmaları yer alır.

- **Pratik Senaryo: Burp Suite ve SQLMap ile SQL Enjeksiyonu**
  - **Burp Suite ile Zafiyet Tespiti:** Burp Suite, web uygulaması güvenlik testleri için popüler bir araç setidir. İlk adım, Burp Suite'i tarayıcınız için bir vekil sunucu (*proxy*) olarak ayarlamaktır. Burp Suite'in `Intercept is on` özelliği açıkken, tarayıcı trafiği Burp üzerinden geçer ve yakalanır. Bir tester, URL'deki veya bir formdaki parametrelere enjeksiyon payloadları (`' OR 1 =1--`) girerek uygulamayı test edebilir. Burp Suite, trafiği manipüle etme ve kaydetme olanağı sunar, bu da manuel testler için kritik öneme sahiptir.
  - **SQLMap ile İstismar:** Burp Suite'den kaydedilen bir HTTP isteği (`.txt` dosyası olarak) SQLMap'e beslenerek, SQL Enjeksiyonu saldırısı otomatize edilebilir.
  - **Örnek Komut:**

    ```
    $ sqlmap -r saved_request.txt -p 'id'
    ```

    Bu komut, `saved_request.txt` dosyasındaki HTTP isteğini kullanarak `'id'` parametresinde bir SQL enjeksiyonu olup olmadığını kontrol eder. SQLMap, zafiyeti bulduğunda, veritabanı sürümü hakkında bilgi verebilir ve hatta bir SQL kabuğu (`--sql-shell`) açarak tester'ın veritabanına doğrudan komut göndermesini sağlayabilir.

11.3.4 Ağ Hizmeti İstismarı ve İstismar Sonrası

Ağ hizmeti istismarı, bir ağdaki zafiyetlerin kullanılarak ilk erişimin elde edilmesidir. İstismar sonrası (*post-exploitation*) aşaması ise, bu ilk erişim elde edildikten sonra gerçekleştirilen tüm işlemlerdir. Bu aşamanın temel amacı, sistem üzerindeki kontrolü artırmak ve kuruluşun iş süreçleri üzerindeki gerçek etkiyi göstermektir.

- **Ayrıcalık Yükseltme (Privilege Escalation):** İlk erişim genellikle düşük yetkili bir kullanıcı hesabı üzerinden elde edilir. Ayrıcalık yükseltme, bu yetkileri artırarak yönetici (*admin*) veya kök (*root*) düzeyinde tam kontrol kazanma sürecidir.
- **Kalıcılık Sağlama (Maintaining Persistence):** Saldırganın, sistem yeniden başlatılsa veya orijinal zafiyet giderilse bile ele geçirilen sisteme geri dönebilmesini sağlayan mekanizmaların (*arka kapılar*, *gizli hesaplar*, *zamanlanmış görevler*) oluşturulmasıdır.
- **Veri Toplama:** Ele geçirilen sistemde ve ağ ortamında değerli bilgilerin (*kimlik bilgileri*, *hassas dosyalar*, *ağ topolojisi*) toplanmasıdır.
- **Yanal Hareket (Lateral Movement):** İlk ele geçirilen makineyi bir köprü (*pivot*) olarak kullanarak, dışarıdan erişilemeyen ağ içindeki diğer sistemlere sızma tekniğidir.

Sızma testlerinin ve Red Team operasyonlarının gerçek değeri, genellikle *post-exploitation* aşamasında ortaya çıkar. Bu aşama, normal bir kullanıcı hesabının ele geçirilmesinin bir kuruluşa ne kadar zarar verebileceğini ve nasıl daha geniş bir ihlale yol açabileceğini göstererek yöneticilere somut bir risk profili sunar.

Amaç	Tanım	Teknik Örnekleri
Ayrıcalık Yükseltme	Düşük yetkili bir hesaptan daha yüksek yetkilere geçiş.	Zafiyetli yazılımları istis
Kalıcılık Sağlama	Saldırganın gelecekte sisteme yeniden erişimini güvence altına alma.	Gizli kullanıcı hesapları
Veri Toplama	Değerli bilgileri keşfetme ve toplama.	Sistem ve hesap bilgileri
Yanal Hareket	İlk ele geçirilen sistemden ağdaki diğer sistemlere yayılma.	Ele geçirilen kimlik bilgi

11.3.5 Kablosuz Ağ Sızma Testi Yöntemleri

Kablosuz ağ sızma testleri, kablosuz ağ güvenliğindeki zayıflıkları ortaya çıkarmayı amaçlar. Bu testler, WEP ve WPA/WPA2 gibi şifreleme protokollerini ve bu protokollere yönelik bilinen saldırıları hedefler.

- **Ağ Keşfi ve Paket Yakalama:** İlk adım, kablosuz ağ arayüzünü `monitor mode`'a almak için `airmon-ng` gibi bir araç kullanmaktır. Ardından, `airodump-ng` ile ağdaki erişim noktaları, bağlı istemciler ve ağ trafiği hakkında bilgi toplanır.
- **Deauthentication Saldırısı:** `aireplay-ng` aracı kullanılarak, bir istemci kablosuz ağdan zorla düşürülür. İstemci ağa yeniden bağlanırken, *handshake* paketleri yakalanır. Bu paketler, ağın şifresini kırmak için kullanılır.
- **Şifre Kırma:** Yakalanan *handshake* paketleri, `aircrack-ng` aracıyla sözlük saldırıları veya diğer teknikler kullanılarak şifrenin kırılabileceği kontrol edilir. Ayrıca, `KRACK` (Key Reinstallation Attack) gibi saldırılar, WPA2 protokolündeki zafiyetlerden yararlanarak veri akışını şifresiz hale getirebilir.
- **Sahte Erişim Noktası (*Evil Twin*) Oluşturma:** `airbase-ng` ile sahte bir erişim noktası kurulabilir ve bu nokta, meşru ağa bağlandığını düşünerek giriş yapan istemcileri kandırmak için kullanılır.

11.4 Sosyal Mühendislik Testi ve Fiziksel Güvenlik

Sosyal mühendislik, insan psikolojisini manipüle ederek, kişileri gizli bilgileri ifşa etmeye veya belirli eylemleri gerçekleştirmeye ikna etme sanatıdır. Sızma testlerinde sosyal mühendislik, bir kuruluşun insan faktörüne dayalı güvenlik zafiyetlerini değerlendirmek için kullanılır. Fiziksel güvenlik ise, bir kuruluşun tesislerine, veri merkezlerine ve diğer kritik altyapılarına yetkisiz fiziksel erişimi önlemeyi amaçlar.

11.4.1 Sosyal Mühendislik Kampanya Tasarımı ve Uygulaması

Sosyal mühendislik kampanyaları, belirli bir hedefe yönelik olarak tasarlanmış ve genellikle bir dizi aşamadan oluşan saldırılardır. Bu kampanyaların amacı, hedefin güvenlik farkındalığını test etmek ve insan faktöründen kaynaklanan zafiyetleri ortaya çıkarmaktır.

- **Spear Phishing:** Belirli bir birey veya organizasyona yönelik özelleştirilmiş oltalama saldırılarıdır. Genellikle, kurbanın ilgisini çekecek veya aciliyet hissi uyandıracak şekilde tasarlanır.
- **Pretexting:** Saldırganın, kendisini güvenilir bir kişi veya kurum olarak tanıtarak bilgi toplamasıdır. Bu, genellikle telefonla veya yüz yüze etkileşimle gerçekleştirilir.
- **Baiting:** Kurbanın merakını veya açgözlülüğünü kullanarak onu tuzağa düşürmeyi amaçlayan bir tekniktir. Örneğin, bir USB bellek üzerine kötü amaçlı yazılım yükleyerek, kurbanın bu belleği bilgisayarına takmasını sağlamak.

11.4.2 Phishing Simülasyonu ve Farkındalık Testi

Phishing simülasyonları, bir kuruluşun çalışanlarını gerçek saldırılara karşı hazırlamak için tasarlanmış kontrollü siber güvenlik tatbikatlarıdır. Bu testler, çalışanların bir phishing e-postasını tanıma ve uygun şekilde tepki verme yeteneklerini değerlendirir.

- **Kampanya Adımları:**
  1. **Planlama:** Kampanyanın hedefleri (örneğin, tıklama oranını düşürmek), kapsamı ve simülasyon türleri (e-posta, sesli oltalama (*vishing*), kısa mesaj oltalama (*smishing*)) belirlenir.
  2. **E-posta Oluşturma:** Hedef kitlenin inanabileceği, gerçekçi ve cazip phishing e-postaları tasarlanır.

3. **Yürütme ve İzleme:** E-postalar çalışanlara gönderilir ve tıklama, veri girişi veya raporlama gibi tepkileri izlenir.
4. **Raporlama ve Eğitim:** Simülasyon sonuçları analiz edilir. Zafiyet gösteren çalışanlara, davranışlarını düzeltmeleri için ek güvenlik farkındalığı eğitimleri verilir.

Phishing simülasyonlarının temel değeri, bir kuruluşun insan faktörü zafiyetlerini belirlemesine ve çalışanların güvenlik farkındalığını somut olarak artırmasına yardımcı olmasıdır.

11.4.3 Fiziksel Sızma Testi ve Tesis Değerlendirmesi

Fiziksel sızma testi, bir tesisin fiziksel güvenlik kontrollerini (*kilitler*, *kameralar*, *güvenlik görevlileri*) test etmek amacıyla, gerçek bir saldırı senaryosunun taklit edilmesidir. Bu testler, dijital sistemlerin güvenliğinin, fiziksel güvenliğin zayıf noktaları nedeniyle nasıl tehlikeye atılabileceğini gösterir.

- **Yöntemler:**
  - **Lock Picking (Kilit Açma):** Fiziksel kilitlerin ne kadar kolay aşıldığını test eder.
  - **Tailgating (Kapıdan Arkadan Girme):** Yetkili bir kişinin arkasından, kimlik doğrulaması olmadan binaya girme tekniğidir. Bu, sosyal mühendisliğin bir parçasıdır.
  - **Dumpster Diving (Çöp Karıştırma):** Atılan belgelerden veya fiziksel atıklardan hassas bilgileri (parolalar, ağ diyagramları, müşteri bilgileri) toplama yöntemidir.
  - **RFID Klonlama:** Yetkisiz erişim elde etmek amacıyla, bir erişim kartının veya RFID etiketinin kopyalanmasıdır.

11.4.4 OSINT Tabanlı Sosyal Mühendislik Saldırı Vektörleri

OSINT (*Open Source Intelligence*), halka açık kaynaklardan toplanan bilgilerin analizidir. Bu bilgiler, sosyal mühendislik saldırılarının temelini oluşturur. Örneğin, sosyal medya platformlarından toplanan çalışanların isimleri, unvanları, hobileri ve ilişkileri gibi veriler, bir *spear phishing* veya *CEO dolandırıcılığı* saldırısı için son derece gerçekçi ve inandırıcı senaryolar oluşturmakta kullanılır. Bu, dijital ve fiziksel dünyalar arasındaki sınırların ne kadar bulanıklaştığını ve bir saldırı zincirinin nasıl OSINT ile başlayıp, fiziksel bir erişimle devam edebileceğini gösterir.

11.4.5 İnsan Faktörü Güvenlik Değerlendirme Yöntemleri

Sızma testleri, sadece teknik zafiyetleri değil, aynı zamanda güvenlik zincirindeki en zayıf halka olan insan faktörünü de değerlendirmelidir. Phishing ve fiziksel güvenlik testleri, sadece bir zafiyet bulma aracı değil, aynı zamanda çalışanların farkındalığını ölçme ve iyileştirme yöntemleridir. Bir kuruluş, milyonlarca dolarlık teknik güvenlik önlemine yatırım yapsa bile, bir çalışanın tek bir dikkatsiz tıklaması veya kapıyı açık bırakması, tüm bu yatırımları boşa çıkarabilir. Bu nedenle, sürekli eğitim, düzenli testler ve pozitif geri bildirim yoluyla bir güvenlik kültürü oluşturmak, teknolojiden bağımsız olarak bir kuruluşun güvenliğini artırmak için hayati önem taşır.

11.5 Red Team Operasyonları ve Gelişmiş Kalıcı Tehdit Simülasyonu

Red Team operasyonları, bir kuruluşun güvenlik savunmalarını gerçekçi bir saldırı senaryosu altında test etmek için tasarlanmış, hedef odaklı bir sızma testidir. Geleneksel sızma testlerinden farklı olarak, Red Team operasyonları daha gizli ve uzun süreli olabilir. Bu operasyonlar, Gelişmiş Kalıcı Tehdit (APT) gruplarının taktiklerini, tekniklerini ve prosedürlerini (TTP'ler) taklit ederek, bir kuruluşun tespit ve müdahale yeteneklerini değerlendirir.

11.5.1 Red Team vs. Penetration Testing Farkları

Penetrasyon testi (*sızma testi*) ile Red Team operasyonları, her ikisi de bir kuruluşun güvenliğini test etse de, amaç, kapsam ve metodoloji açısından önemli farklılıklar gösterir.

- **Amaç:** Sızma testinin birincil amacı, belirlenmiş sistemler içindeki teknik zafiyetleri bulmak ve belgelemektir. Red Team operasyonunun amacı ise, gerçek bir saldırıyı taklit ederek kuruluşun tespit ve müdahale yeteneklerini değerlendirmektir.
- **Kapsam:** Sızma testi, genellikle belirli ağlar, sistemler veya uygulamalarla sınırlı, dar bir kapsama sahiptir. Red Team operasyonları ise daha geniştir ve sosyal mühendislik, fiziksel güvenlik ihlalleri gibi teknik olmayan vektörleri de kapsayabilir.
- **Gizlilik:** Sızma testleri genellikle *noisy* (*gürültülü*) bir yaklaşıma sahiptir ve savunma ekibi (*Blue Team*) testten haberdardır. Red Team operasyonları ise genellikle gizli (*stealthy*) yürütülür ve saldırganların uzun süre tespit edilmeden kalma çabalarını taklit eder.
- **Metodoloji:** Sızma testleri, OWASP veya PTES gibi yapılandırılmış ve tekrarlanabilir metodolojileri izler. Red Team operasyonları ise esnek ve yaratıcıdır, saldırganın hedefine ulaşmak için çeşitli taktikleri gerçek zamanlı olarak adapte etmesini gerektirir.

Özellik	**Sızma Testi (Penetration Testing)**	**Red Team Operasyonları (Red Teaming)**
Amaç	Teknik zafiyetleri belirlemek.	Savunma yeteneklerini ve olay müdahale süreçlerin
Kapsam	Belirlenmiş teknik sınırlar.	Kurumun tamamı (dijital, fiziksel, insan faktörü).
Odak	Mümkün olduğunca çok zafiyet bulmak.	Belirli hedeflere ulaşmak (örneğin, hassas verilere e
Metodoloji	Yapılandırılmış ve tekrarlanabilir.	Esnek, adapte edilebilir ve yaratıcı.
Gizlilik	Genellikle savunma ekibi testten haberdardır (*noisy*).	Genellikle gizli (*stealthy*), savunma ekibini şaşırtır.

11.5.2 Gelişmiş Kalıcı Tehdit (APT) Simülasyon Kampanyaları

Gelişmiş Kalıcı Tehdit (*Advanced Persistent Threat - APT*) simülasyonları, devlet destekli veya yüksek organize siber suç grupları gibi sofistike tehdit aktörlerinin kullandığı uzun vadeli ve hedef odaklı saldırıları taklit eder. Bu simülasyonlar, standart testlerle tespit edilemeyen gizli zafiyetleri ve süreçsel zayıflıkları ortaya çıkarmayı hedefler. Kampanyalar, planlama, senaryo geliştirme (MITRE ATT&CK gibi çerçevelerden yararlanılarak), yürütme ve raporlama gibi aşamalardan oluşur.

11.5.3 Komuta ve Kontrol (C2) Altyapısı Kurulumu

Komuta ve kontrol (*C2*), bir saldırganın ele geçirdiği sistemlerle gizli iletişimi sürdürmek ve onlara talimatlar göndermek için kullandığı araç ve teknikler bütünüdür. Bir sistemin ele geçirilmesi, bir C2 kanalı kurulmadıkça anlamsızdır, zira bu kanal, verilerin sızdırılması ve saldırının sonraki aşamaları için hayati öneme sahiptir.

- **C2 Mimari Türleri:**
  - **Merkezi Mimari:** En yaygın modeldir. Ele geçirilen her sistem (*zombie*), merkezi bir sunucuya bağlanarak komut bekler. Tespiti ve engellenmesi nispeten kolaydır.
  - **P2P (*Peer-to-Peer*) Mimari:** Merkezi bir sunucuya bağımlı değildir. Komutlar, botnet üyeleri arasında eşler arası olarak aktarılır. Bu, tespiti çok daha zorlaştırır.
  - **Rastgele Mimari:** Tespit edilmeyi engellemek için tasarlanmıştır. C2 iletişimi, sosyal medya yorumları, IRC odaları veya DNS sorguları gibi güvenilir ve yaygın kullanılan kaynaklar aracılığıyla iletilir.

11.5.4 Living-off-the-Land (LOTL) Teknikleri ve Kaçınma Yöntemleri

Living-off-the-Land (*LOTL*), bir saldırganın hedef sistemde zaten var olan meşru araçları ve ikili dosyaları (*binaries*) kullanarak tespit edilmekten kaçınma tekniğidir. Bu teknikler, geleneksel imza tabanlı güvenlik çözümlerini atlatmada son derece etkilidir, çünkü kötü amaçlı yazılım indirmek veya sisteme enjekte etmek yerine, zaten güvenilen araçları kötüye kullanırlar.

- **PowerShell:** Windows işletim sistemlerinde yerleşik olarak bulunan güçlü bir komut satırı aracıdır. Saldırganlar, PowerShell komut dosyalarını kötü amaçlı yükleri indirmek, sistem ayarlarını değiştirmek veya hassas verileri sızdırmak için kullanır.
- **LOLBins (*Living Off the Land Binaries*):** `rundll32.exe`, `mshta.exe` ve `certutil.exe` gibi yasal sistem ikilileridir. Bu dosyalar, güvenlik kontrollerini atlatmak ve kötü amaçlı kodları yürütmek için kötüye kullanılabilir. *LOLBAS* projesi, bu ikilileri belgeler.
- **Mimikatz:** Bellekte saklanan kimlik bilgilerini, açık metin parolaları ve *hash*'leri çıkarmak için kullanılan güçlü bir araçtır.

LOTL tekniklerinin yükselişi, siber güvenlikte imza tabanlı savunmaların artık yetersiz olduğunu ve davranışsal analiz ve olay yanıtı (*EDR*) gibi daha gelişmiş çözümlerin zorunluluğunu ortaya koyar. Saldırganlar, meşru araçları kullanarak *radar altı* kalmaya çalışırlar, bu da savunmacılar için tespiti son derece zor hale getirir.

11.5.5 Red Team Tatbikatı Planlama ve Yürütme

Bir Red Team operasyonunun başarısı, dikkatli planlama ve profesyonel yürütmeye bağlıdır. Süreç, PTES'e benzer ancak daha esnek ve gizlilik odaklıdır.

1. **Planlama:** Hedefler, kapsam, bütçe ve yasal izinler belirlenir. Bu, tüm paydaşların operasyonun doğası hakkında tam bilgi sahibi olmasını sağlar.
2. **Keşif:** Pasif ve aktif bilgi toplama teknikleri kullanılarak hedef hakkında mümkün olduğunca fazla veri toplanır.
3. **İlk Erişim ve Kalıcılık:** Sosyal mühendislik veya teknik zafiyetler kullanılarak ilk erişim elde edilir ve bu erişimi sürdürmek için kalıcılık mekanizmaları kurulur.
4. **Yanal Hareket:** Elde edilen ilk erişim kullanılarak ağ içinde daha kritik sistemlere doğru ilerleme kaydedilir.
5. **Veri Sızdırma (*Exfiltration*):** Saldırının nihai hedeflerinden biri olan hassas verilerin güvenli bir şekilde dışarıya aktarılmasıdır.
6. **Raporlama ve *Debriefing*:** Operasyonun tamamlanmasının ardından, savunma ekibiyle (*Blue Team*) bir *debriefing* toplantısı yapılır. Bu toplantıda, saldırı yolu, elde edilen bulgular ve savunma ekibinin tepkileri detaylı olarak gözden geçirilir. Bu, kuruluşun güvenlik duruşundaki ve süreçlerindeki zayıflıkları anlamasına yardımcı olur.

11.6 Bug Bounty Programs ve Responsible Disclosure

Bug bounty programları, bir kuruluşun ürünlerindeki veya hizmetlerindeki güvenlik açıklarını bulan ve bildiren araştırmacılara ödül verdiği bir sistemdir. Bu programlar, bir kuruluşun güvenlik duruşunu sürekli olarak test etmesine ve iyileştirmesine olanak tanır. Sorumlu ifşa (responsible disclosure), bir güvenlik açığı bulunduğunda, bu açığın kamuoyuna duyurulmadan önce ilgili kuruluşa bildirilmesi ve düzeltilmesi için zaman tanınması sürecidir.

11.6.1 Bug Bounty Program Yapısı ve Yönetimi

Bug bounty, bir kuruluşun varlıklarındaki güvenlik açıklarını bulmaları ve raporlamaları karşılığında etik hackerlara finansal ödül (*bounty*) sunan bir güvenlik girişimidir.

- **Program Türleri:**
  - **Herkese Açık (*Public*) Programlar:** Etik hacking topluluğunun tüm üyelerine açıktır. Geniş bir araştırmacı kitlesi tarafından test edilme fırsatı sunar, ancak gelen raporların (*gürültü* dahil) yönetimi daha fazla çaba gerektirebilir.
  - **Özel (*Private*) Programlar:** Sadece davetle girilebilen programlardır. Kuruluşlara, daha yüksek sinyal-gürültü oranı sunan, güvenilir ve seçilmiş bir araştırmacı grubuna erişim sağlar.
- **Yönetim Süreçleri:**
  - **Kapsam Tanımı:** Programın test edilecek ve edilmeyecek varlıklarını, hedefleri ve kuralları net bir şekilde tanımlamak kritik öneme sahiptir.
  - **Ödül Yapısı:** Zafiyetin ciddiyetine ve iş üzerindeki potansiyel etkisine göre belirlenen adil ve şeffaf bir ödül yapısı, yüksek kaliteli bulguları teşvik eder.
  - **Bug Triaging (Rapor Sıralama ve Önceliklendirme):** Gelen raporların geçerliliğini, alaka düzeyini ve kritiklik seviyesini değerlendiren bir süreçtir. Bu, gürültüyü eleyerek yalnızca yüksek etkili bulguların ilgili ekiplere ulaşmasını sağlar.

11.6.2 Araştırmacı İletişimi ve İlişki Yönetimi

Başarılı bir bug bounty programı, şeffaf ve zamanında iletişimle desteklenen güçlü bir ilişki yönetimi gerektirir. Araştırmacılara zamanında geri bildirim sağlamak ve adil ödüller sunmak, onların programa olan bağlılıklarını sürdürmek için kritik öneme sahiptir.

11.6.3 Zafiyet Doğrulama ve Ciddiyet Değerlendirmesi

Rapor edilen bir zafiyet, iyileştirme sürecine başlanmadan önce doğrulanmalıdır. Kaliteli bir rapor, zafiyetin açık bir açıklamasını, tekrarlama adımlarını, kanıtı ve potansiyel etkisinin değerlendirmesini içerir. Zafiyetin ciddiyetini objektif bir şekilde değerlendirmek için kullanılan en yaygın metodoloji, **CVSS (Common Vulnerability Scoring System)**'dir.

CVSS, bir zafiyetin temel özelliklerini, zaman içinde değişen faktörleri ve bir kullanıcının ortamına özgü nitelikleri değerlendiren üç grup metriğe sahiptir.

- **Temel (*Base*) Grup:** Zafiyetin doğal, kalıcı özelliklerini yansıtır. Saldırı vektörü, karmaşıklığı, gereken yetki ve kullanıcı etkileşimi gibi metrikleri içerir.
- **Zamansal (*Temporal*) Grup:** Zafiyetin zamanla değişen özelliklerini (*exploit* kodunun varlığı, düzeltmenin yayınlanması) yansıtır.
- **Çevresel (*Environmental*) Grup:** Zafiyetin belirli bir kuruluş ortamındaki etkisini (*gizlilik*, *bütünlük*, *kullanılabilirlik*) değerlendirir.

Metrik Grubu	Metrikler	Açıkl
Temel (*Base*)	Saldırı Vektörü, Saldırı Karmaşıklığı, Gereken Ayrıcalık, Kullanıcı Etkileşimi.	Zafiy
Zamansal (*Temporal*)	İstismar Edilebilirlik, İyileştirme Düzeyi, Güven Raporu.	Zafiy
Çevresel (*Environmental*)	Gizlilik Gereksinimleri, Bütünlük Gereksinimleri, Kullanılabilirlik Gereksinimleri.	Zafiy

CVSS gibi puanlama sistemleri, güvenlik uzmanlarının teknik bulguları iş liderlerine anlaşılır bir dille (*risk düzeyi*, *öncelik*) sunmasını sağlar.

11.6.4 İyileştirme Koordinasyonu ve Zaman Çizelgesi Yönetimi

Zafiyetlerin giderilmesi, sadece güvenlik ekibinin değil, aynı zamanda geliştirme ve operasyon ekiplerinin de dahil olduğu kolektif bir çabadır (*DevSecOps*). İyileştirme süreci, yama yönetimi, konfigürasyon güncellemeleri veya hatta bazı durumlarda riskin kabul edilmesi gibi adımları içerir.

11.6.5 Yasal Çerçeve ve Araştırmacı Koruması

Sorumlu Açıklama (*Responsible Disclosure*): Etik hackerların, bir zafiyeti kötü niyetli kişiler tarafından istismar edilmeden önce, etkilenen kuruluşa veya satıcıya gizlice raporlaması sürecidir. Bu süreç, kuruluşun bir yama veya düzeltme geliştirmesi için zaman tanır.

Yasal Koruma (*Safe Harbor*): Sorumlu açıklama politikaları, araştırmacılara, kurallara uydukları sürece yasal yükümlülüklerden (*yetkisiz erişim*) muafiyet sağlar. HackerOne'ın "Gold Standard Safe Harbor" bildirisi veya CISA'nın "Coordinated Vulnerability Disclosure" programı, araştırmacılar ve şirketler arasında güvene dayalı bir ilişki kurarak güvenlik açıklarının kitlesel olarak ifşa edilmesini önler ve siber ekosisteminin güvenliğini artırır.

Bug bounty programları, güvenliğin tek seferlik bir değerlendirme (*point-in-time assessment*) olmadığını, sürekli bir süreç olduğunu vurgular. Bir kuruluş, dinamik ve sürekli değişen tehdit ortamına karşı savunma yapmak için sürekli geri bildirim ve iyileştirme döngüsüne ihtiyaç duyar. Sorumlu açıklama ve bug bounty programları, siber güvenlik dünyasındaki açık kaynak ve işbirliği kültürünün bir yansımasıdır. Kuruluşlar, dışarıdan gelen geri bildirimi bir tehdit olarak değil, bir fırsat olarak görerek güvenlik duruşlarını radikal bir şekilde güçlendirebilirler. Bu yaklaşım, "kapalı sistemlerin daha güvenli olduğu" yönündeki eski güvenlik anlayışını yıkar.

Bölüm 12

MALWARE ANALİZİ VE TERSİNE MÜHENDİSLİK

Giriş

Malware analizi ve tersine mühendislik, siber güvenlik uzmanlarının kötücül yazılımları anlamak, davranışlarını incelemek ve etkili karşı önlemler geliştirmek için kullandıkları kritik disiplinlerdir. Bu bölümde malware türleri, analiz teknikleri ve tersine mühendislik metodolojilerini detaylı olarak inceleyeceğiz.

12.1 Malware Sınıflandırması ve Türleri

12.1.1 Malware Kavramı ve Temel Türleri

Malware, "*malicious software*" (kötü amaçlı yazılım) teriminin kısaltmasıdır ve bilgisayar sistemlerine zarar vermek, onları bozmak, veri çalmak veya genel olarak meşru olmayan eylemleri gerçekleştirmek için tasarlanmış yazılım veya kod anlamına gelir.[5] Malware, işlevine ve yayılma biçimine göre çeşitli kategorilere ayrılır.

- **Virüsler:** Kendini çoğaltan ve varlığını sürdürmek ve yayılmak için bir "*host*" (ana bilgisayar) uygulamasına ihtiyaç duyan kötü amaçlı kodlardır. Bir virüs, ana bilgisayar dosyası çalıştırıldığında aktif hale gelir ve kendini diğer uygulamaların koduna yerleştirerek çoğalır.[5] Virüsler, sistem kaynaklarını yavaşlatmaktan dosyaları bozmaya kadar değişen hasarlara neden olabilir. 1999'daki Melissa virüsü, kendini e-posta kişilerine göndererek sunucuları aşırı yüklemesi ve milyonlarca dolarlık hasara yol açmasıyla bu türün bilinen bir örneğidir.[6]
- **Solucanlar (Worms):** Kendi kendine çoğalabilen, bağımsız kötü amaçlı programlardır ve bir ana bilgisayar dosyasına ihtiyaç duymazlar.[5] Solucanlar, ağdaki güvenlik açıklarından yararlanarak kullanıcı etkileşimi olmadan cihazlar arasında hızla yayılabilirler.[5] Bu otonom yayılma yeteneği, solucanları özellikle tehlikeli kılar. 2012'de siber casusluk için tasarlanan Flame solucanı, karmaşıklığı ve gizliliğiyle bilinen bir örnektir.[6]
- **Truva Atları (Trojans):** Adını Antik Yunan mitolojisindeki tahta attan alan Truva atları, meşru bir yazılım gibi görünen, ancak kullanıcının kandırılarak sistemine yüklemesi ve çalıştırmasıyla aktifleşen zararlı programlardır.[5] Bir kez etkinleştirildiğinde, dosyaları silebilir, veri çalabilir veya sisteme arka kapılar (*backdoors*) oluşturabilir.[5] Virüsler ve solucanların aksine, Truva atları kendini çoğaltma yeteneğine sahip değildir.[5]
- **Fidye Yazılımları (Ransomware):** Bu malware türü, kullanıcının dosyasını veya sistemini şifreleyerek erişimi engeller ve şifre çözme anahtarı karşılığında genellikle kripto para birimiyle ödeme talep eder.[6]

WannaCry gibi fidye yazılımları, AES ve RSA gibi güçlü kriptografik algoritmalar kullanır.[7] Bu saldırılar, şifrelemenin geri döndürülemez doğası nedeniyle küresel çapta büyük hasarlara yol açmıştır.[7]

– **Gelişmiş Kalıcı Tehditler (APTs):** Gelişmiş Kalıcı Tehditler, bir ağa uzun bir süre boyunca gizlice erişim sağlayan ve burada kalıcı bir varlık oluşturan, sofistike ve hedefli saldırıları ifade eder.[6] APT'lerin ana amacı, genellikle hassas verileri çalmak veya kritik altyapıyı bozmaktır.[9] Bu saldırılar, keşif, sızma, ayrıcalık yükseltme ve veri sızdırma gibi çok aşamalı bir yaşam döngüsünü takip eder.[10]

12.1.2 Diğer Malware Kategorileri

Malware ekosistemi, yukarıda bahsedilen temel türlerin ötesine geçen çeşitli özel tehditleri de içerir:

– **Botnetler:** Siber suçluların komuta ve kontrol (C2) sunucusu üzerinden uzaktan yönettiği, ele geçirilmiş bilgisayarlardan oluşan ağlardır.[6]

– **Adware:** İstenmeyen reklamlar gösteren yazılımlardır.[6]

– **Spyware:** Kullanıcı etkinliğini gizlice izleyen ve hassas bilgileri (örneğin, parolalar, kişisel veriler) toplayan yazılımlardır.[6]

– **Rootkitler:** Kötü amaçlı etkinliği sistemden gizleyen, genellikle çekirdek (*kernel*) düzeyinde çalışan gizli yazılımlardır.[6]

– **Fileless Malware:** Bellekte çalışan ve diskte neredeyse hiçbir iz bırakmadan, geleneksel antivirüs çözümlerinden kaçan bir türdür.[6]

– **Keyloggerlar:** Klavyede yapılan tuş vuruşlarını kaydederek hassas bilgileri çalan yazılımlardır.[6]

12.1.3 Karşılaştırmalı Analiz

Farklı malware türleri arasındaki temel benzerlikleri ve farklılıkları anlamak, bir analistin tehditleri hızlı bir şekilde sınıflandırması ve doğru karşı önlemleri geliştirmesi için kritik öneme sahiptir. Aşağıdaki tablo, bu türlerin ana özelliklerini özetlemektedir.

Malware Türü	Tanım	Yayılma Mekanizması	Temel Amaç
Virüs	Kendi kendini çoğaltan kod	Bir ana bilgisayar dosyasına (örneğin, EXE, DOC) bulaşarak	Dosyaları bozma, sistemde hasar verme
Solucan	Kendi kendine çoğalan, bağımsız program	Ağ zafiyetlerini kullanarak, kullanıcı etkileşimi olmadan	Ağları tüketme, ek malware kurma, arka kapı oluşturma
Truva Atı	Meşru görünen, ancak kötü niyetli yazılım	Kullanıcı etkileşimiyle (örneğin, sahte indirme, e-posta eki)	Veri çalma, arka kapı oluşturma, ek malware indirme
Fidye Yazılımı	Dosyaları şifreleyen ve fidye talep eden yazılım	E-posta ekleri, zafiyetlerden yararlanma, diğer malware'ler tarafından	Şifrelenmiş veriler karşılığında finansal kazanç elde etme
Botnet	Saldırganın uzaktan kontrol ettiği ele geçirilmiş bilgisayarlar ağı	Diğer malware türleri aracılığıyla (örneğin, virüs, solucan)	DDoS saldırıları, spam yayma, kripto madenciliği
Adware	İstenmeyen reklamlar görüntüleyen yazılım	Yasal programlara bağlı olarak veya drive-by indirmelerle	Reklam geliri elde etme

Spyware	Kullanıcı aktivitesini izleyen yazılım	Kullanıcının bilgisi olmadan, genellikle gizlice	Veri çalma, finansal bilgileri ele geçirme
Rootkit	Kötü amaçlı etkinliği gizleyen yazılım	Sisteme sızma veya diğer malware'ler aracılığıyla	Kalıcılık sağlama, diğer kötü amaçlı eylemleri gizleme
Fileless Malware	Diskte iz bırakmayan kötü amaçlı kod	Kötü amaçlı betikler, meşru araçlar (PowerShell) aracılığıyla	Gizli kalma, sistem kaynaklarını kullanma

12.2 Malware Analiz Ortamının Kurulumu

12.2.1 Neden İzole Ortam? Güvenlik ve İzolasyonun Önemi

Malware analizi, doğası gereği yüksek riskli bir faaliyettir. Şüpheli bir dosyayı doğrudan birincil sistemde çalıştırmak, ağın tamamının veya kritik verilerin enfekte olmasına neden olabilir.[11] Bu nedenle, malware'i incelemek için ana ağdan tamamen izole edilmiş bir ortam oluşturmak mutlak bir gerekliliktir. İzole ortamlar, güvenlik uzmanlarının kötü niyetli kodun davranışını güvenli bir şekilde gözlemlemesini sağlar, böylece ağa yayılma riski olmadan saldırının etkileri incelenebilir ve buna karşı etkili önlemler geliştirilebilir.[11]

12.2.2 Sanal Makine Seçimi ve Kurulumu

Malware analizi için en yaygın yaklaşım, bir sanal makine (VM) kullanmaktır. VMware, VirtualBox, KVM, Hyper-V gibi sanallaştırma yazılımları, izole bir laboratuvar ortamı oluşturmak için idealdir.[11] Bu sanal ortamın gerçek bir kullanıcı sistemini taklit etmesi kritik önem taşır. Modern malware, sanal makineleri tespit etmek için tasarlanmış anti-sanallaştırma mekanizmalarına sahiptir. Örneğin, bir malware yeterli CPU çekirdeği veya RAM tahsis edilmediğini ya da sistemde yaygın uygulamaların (MS Word, Chrome) yüklü olmadığını tespit ettiğinde çalışmayı durdurabilir veya zararsız gibi davranabilir.[13] Bu durum, analiste yanlış sonuçlar vererek hatalı bir varsayımda bulunmasına yol açabilir. Bu nedenle, sanal makineye en az 4 GB RAM ve 2 CPU çekirdeği gibi gerçekçi kaynaklar tahsis etmek ve sahte ağ bağlantılarını taklit etmek, bu tür kaçınma taktiklerini atlatmak için hayati adımlardır.[13]

12.2.3 Sanal Sandbox Ortamları

Malware analizinde davranışsal gözlemi otomatikleştirmek için sandbox ortamları kullanılır. Bir sandbox, kötü niyetli kodu bir sanal makine içinde çalıştırır ve dosya sistemi, kayıt defteri, ağ iletişimleri ve işlem aktiviteleri gibi sistem üzerindeki tüm etkileşimlerini gerçek zamanlı olarak izler.[14] Bu yaklaşım, geleneksel imza tabanlı sistemlerin atladığı sıfır-gün tehditlerini ve gelişmiş gizlenme tekniklerini ortaya çıkarmak için son derece etkilidir.[16]

Hazır Sandbox Sistemleri: Cuckoo Sandbox ve Özelleştirilmesi

Cuckoo Sandbox, şüpheli dosyaların analizini otomatikleştiren açık kaynaklı ve popüler bir araçtır.[17] Cuckoo'nun kurulumu, Python kütüphaneleri, bir sanallaştırma yazılımı (örneğin VirtualBox), ağ trafiği yakalama aracı (tcpdump) ve Volatility gibi ek bağımlılıkların yüklenmesini gerektirir.[18] Cuckoo, modüler yapısı sayesinde analistin ihtiyaçlarına göre derinlemesine özelleştirilebilir. Bu özelleştirme, `cuckoo.conf`, `machinery.conf` ve `reporting.conf` gibi çeşitli yapılandırma dosyaları aracılığıyla yapılır.[17] Bir analist, ağ trafiğini belirli bir şekilde yönlendirebilir, sanal makineleri tanımlayabilir ve analiz raporlarının çıktısını özelleştirebilir. Bu esneklik, Cuckoo'yu hem temel hem de ileri düzey analizler için güçlü bir platform haline getirir.[19] Sandbox ortamı kurmak sadece bir güvenlik önlemi değil, aynı zamanda malware'in anti-analiz mekanizmalarına karşı stratejik

bir hamledir. Malware'in gerçek bir kullanıcı ortamı ile izole bir test ortamı arasındaki farkı anlama yeteneği, analistin başarısı için temel bir gerekliliktir.[13]

12.3 Statik Malware Analizi

12.3.1 Statik Analiz: Kavram ve Yöntemler

Statik malware analizi, kötü amaçlı yazılım kodunu çalıştırmadan inceleme sürecidir. Bu yöntem, bir dosyanın içeriğini, yapısını, gömülü dizeleri ve meta verilerini hızlı bir şekilde gözden geçirmek için güvenli bir yol sunar.[20] Statik analiz, dosya adları, hash değerleri, IP adresleri, alan adları ve dosya başlık verileri gibi IOC'leri çıkarmak için kullanılabilir.[1] Dinamik analizin aksine, statik analiz kodu etkinleştirmeden yapıldığı için sistemin enfeksiyon riski yoktur.

12.3.2 Dosya Formatı Analizi

Malware analizinde kritik bir ilk adım, dosyanın formatını anlamaktır. Analistlerin en çok karşılaştığı iki ana format, Windows için PE (Portable Executable) ve Linux için ELF (Executable and Linkable Format) formatlarıdır.

- **PE (Portable Executable) Dosya Yapısı:** Windows sistemlerinde çalışan neredeyse tüm yürütülebilir dosyalar (EXE, DLL) PE formatındadır.[22] PE dosyası, işletim sistemi yükleyicisinin yürütülebilir kodu yönetmesi için gerekli bilgileri içeren bir veri yapısıdır. Bu yapı, dosyanın mimarisi (32-bit/64-bit), gerekli kütüphaneler ve bölüm bilgileri gibi kritik meta verileri içeren bir başlıkla başlar.[23] PE dosya başlığının incelenmesi, bir analistin malware'in potansiyel yetenekleri hakkında bir fikir edinmesini sağlar. Özellikle, dosyanın ihtiyaç duyduğu harici işlevleri ve kütüphaneleri listeleyen İçe Aktarma (*Import*) Tablosu ve dosyanın diğer programların kullanımına sunduğu işlevleri listeleyen Dışa Aktarma (*Export*) Tablosu tersine mühendislik için hayati önem taşır.[23]
- **ELF (Executable and Linkable Format) Dosya Yapısı:** Linux ve diğer Unix benzeri sistemlerde yürütülebilir dosyalar, nesne kodu ve paylaşılan kütüphaneler için standart dosya formatıdır.[26] ELF formatı, farklı mimarileri ve byte sıralamalarını (*endianness*) desteklemesi nedeniyle esnek ve çapraz platform bir yapıya sahiptir. Her ELF dosyası, bir başlık ve bunu takip eden program ve bölüm başlık tablolarından oluşur.[26] Program başlıkları, dosyanın çalışma zamanı yürütülmesi için gereken bilgileri sağlarken, bölüm başlıkları bağlama ve yer değiştirme için önemli veriler içerir.[26]

Aşağıdaki tablo, PE ve ELF dosyalarının ana bölümlerini karşılaştırmalı olarak sunmaktadır:

Bölüm Adı	PE Dosyasındaki Amacı	ELF Dosyasındaki Amacı	Önemli Özellikler
`.text`	Yürütülebilir CPU talimatlarını içerir	Yürütülebilir kodu içerir	Okunabilir ve yürütülebilir izinlere sahiptir
`.data`	Başlatılmış global verileri içerir	Başlatılmış global verileri içerir	Okuma ve yazma izinlerine sahiptir
`.rdata`	Salt okunur verileri (örneğin, içe aktarma bilgisi) içerir	Genellikle `.rodata` olarak adlandırılır, salt okunur verileri içerir	Sadece okuma izinlerine sahiptir
`.rsrc`	İkonlar, menüler ve gömülü dosyalar gibi kaynakları saklar	Kaynakları saklamak için isteğe bağlı bir bölümdür	Genellikle okunabilir izinlere sahiptir

12.3.3 Dize Analizi ve Önemli Verilerin Çıkarılması

Bir malware ikilisinden insan tarafından okunabilir dizeleri çıkarmak, analiste dosyanın olası davranışları hakkında değerli ipuçları verebilir. `strings` komutu gibi araçlar, alan adları, IP adresleri, hata mesajları veya API işlev adları gibi statik verileri hızlıca ortaya çıkarabilir.[28] Örneğin, `URLDownloadToFile` gibi bir Windows API dizesinin varlığı, malware'in internetten dosya indirme yeteneğine sahip olduğunu düşündürebilir.[29] Ancak, bu yaklaşımın önemli bir sınırlaması vardır. Eğer bir dosya "paketlenmişse" veya şifrelenmişse, statik analizden elde edilen dizeler anlamsız karakterler yığını (*gibberish*) gibi görünebilir.[30] Bu, analiste malware'in anti-analiz teknikleri kullandığını ve davranışını anlamak için dinamik analize geçmesi gerektiğini işaret eder. Bu durum, statik analizin dinamik analize olan ihtiyacı nasıl doğurduğunun bir örneğidir.

12.3.4 İmza Oluşturma: YARA Kuralları

YARA (*Yet Another Recursive Acronym*), güvenlik araştırmacılarının kötü amaçlı yazılım ailelerini metinsel veya ikili kalıplara göre tanımlamasına ve sınıflandırmasına yardımcı olan bir araçtır.[32] YARA, geleneksel dosya `hash`'lerine dayalı imzalamadan daha güçlü bir yaklaşımdır çünkü malware yazarlarının küçük değişikliklerle oluşturduğu yeni varyantları da tespit edebilir. Bir YARA kuralı, `meta` (açıklama ve yazar bilgisi), `strings` (aranacak desenler) ve `condition` (mantıksal ifade) olmak üzere üç ana bölümden oluşur.[33]

– **İkili Kalıp Eşleştirme (Binary Pattern Matching):** YARA'nın en güçlü yönlerinden biri, sadece metinsel dizeleri değil, aynı zamanda dosyalardaki spesifik onaltılık (*hexadecimal*) değerleri ve byte dizilerini de aramasına olanak tanıyan ikili kurallardır. Bu kurallar, `wildcards (?)` , `jumps ([4-16])`, `alternatives ((DE|AD))` ve `offsets` gibi gelişmiş söz dizimlerini kullanarak son derece esnek ve dayanıklı hale getirilebilir.[35] Örneğin, bir dosyanın paketlenip paketlenmediğini anlamak için `math.entropy()` kullanarak dosyanın rastgelelik seviyesini kontrol eden bir kural yazılabilir. Bu yetenekler, analiste bir malware'i sadece bilinen bir örnek olarak değil, aynı zamanda belirli bir ailenin parçası olarak tanımlama yeteneği kazandırır.[33]

12.4 Dinamik Malware Analizi

12.4.1 Dinamik Analiz: Kavram ve Yöntemler

Dinamik malware analizi, şüpheli bir dosyayı izole edilmiş ve kontrollü bir ortamda (sandbox) çalıştırarak davranışını gözlemleme sürecidir.[14] Bu yöntem, statik analizde gizlenen veya şifrelenen eylemleri ortaya çıkarır ve malware'in gerçek dünyadaki etkileşimlerini gösterir. Dinamik analiz, sıfır-gün tehditleri, Gelişmiş Kalıcı Tehditler (APTs) ve polimorfik malware gibi geleneksel imza tabanlı sistemleri atlatabilen tehditleri tespit etmek için özellikle etkilidir.[16]

12.4.2 Dosya Sistemi ve Kayıt Defteri İzleme

Malware, genellikle kalıcılık sağlamak ve işlevlerini yerine getirmek için dosya sisteminde ve kayıt defterinde değişiklikler yapar. Process Monitor (Procmon)[37] ve Process Hacker[30] gibi araçlar, bu değişiklikleri gerçek zamanlı olarak izlemek için vazgeçilmezdir. Bu araçlar, yeni oluşturulan, silinen veya değiştirilen dosyaları, başlatılan yeni işlemleri ve kayıt defteri anahtarlarındaki değişiklikleri takip eder.[30] Ancak, malware yazarları da bu araçların farkındadır. [2]'daki bilgilere göre, bazı sofistike malware'ler kullanıcı modu araçlarını (örneğin Procmon'u) tespit edip analizden kaçabilir. Bu durum, analistleri, dosya izlemesini çekirdek (*kernel*) seviyesinde yapan ve malware tarafından kolayca tespit edilemeyen gelişmiş sandbox çözümlerini kullanmaya yöneltmiştir.

12.4.3 Ağ Trafiği Analizi

Malware, genellikle saldırganla iletişim kurmak için ağa bağlanır. Bu iletişim, komut ve kontrol (C2) sunucularından talimat almak, ek kötü amaçlı yükleri indirmek veya sisteme ait verileri sızdırmak için gerçekleşir.[39] Wireshark[41] gibi ağ analiz araçları, bu trafiği izlemek ve anormallikleri tespit etmek için kullanılır.[68]'de gösterildiği gibi, bazı malware'ler C2 iletişimi için kendinden imzalı (*self-signed*) sertifikalar kullanır. Bu sertifikaların benzersiz alan verileri veya ana bilgisayar adları (örneğin, `example.com` veya `localhost`) gibi özel desenleri, bir malware ailesini diğerlerinden ayırmak için güçlü birer davranışsal imza görevi görebilir. Bu durum, ağ trafiği analizinin sadece bir anomali tespit aracı olmaktan çıkıp, tehdit istihbaratını zenginleştiren bir köprü görevi gördüğünü gösterir.

12.4.4 API ve Sistem Çağrısı İzleme

Bir programın, işletim sistemiyle etkileşime girmek için kullandığı API (Uygulama Programlama Arayüzü) ve sistem çağrılarını izlemek, malware'in yetkisiz davranışlarını (örneğin, ayrıcalık yükseltme) anlamak için güçlü bir tekniktir.[42] API çağrılarının davranışsal modelleri, makine öğrenimi algoritmaları kullanılarak malware tespiti ve sınıflandırması için kullanılabilir.[43] Örneğin, bir fidye yazılımının dosya okuma, şifreleme ve silme işlemlerini içeren döngüsel bir sistem çağrısı dizisi, onun kötü niyetli davranışını belirlemekte kullanılabilir.[42]

Aşağıdaki tablo, statik ve dinamik analiz tekniklerinin temel özelliklerini, avantajlarını ve dezavantajlarını karşılaştırmaktadır.

Teknik	**Temel Kavram**	**Avantajlar**	**Dezavantajlar**	**Kullanılan Araçlar**
Statik Analiz	Kodu çalıştırmadan inceleme	Güvenli, hızlı, IOC'leri hızlıca çıkarır	Gizlenmiş veya paketlenmiş malware'i analiz edemez	PEStudio, `strings`, YARA, `objdump`, `readelf`
Dinamik Analiz	Kodu izole bir ortamda çalıştırma	Gizlenmiş, polimorfik ve sıfır-gün tehditlerini ortaya çıkarır	Yavaş, riskli, ek kurulum gerektirir	Process Monitor, Wireshark, Cuckoo Sandbox

12.5 Tersine Mühendislik Teknikleri

12.5.1 Sökme (Disassembly) ve Montaj Dili Analizi

Tersine mühendislik, bir programın kaynak koduna sahip olmadan iç işleyişini anlama sürecidir. Derlenmiş ikili dosyalar genellikle kaynak kodundan (örneğin C/C++) okunabilir işlev ve değişken adlarını barındırmaz.[45] Bu durumda, *disassembler* adı verilen araçlar, makine dilini insan tarafından okunabilir montaj diline (*assembly*) çevirir.[46] Montaj dili, bir programın gerçekte ne yaptığını gösteren ham, filtrelenmemiş talimatları içerir.[45] Analistin, `mov` (taşıma), `jmp` (atlama) ve `lea` (adres yükleme) gibi temel komutları ve işlemci bayraklarını anlaması, malware'in mantığını ve kontrol akışını çözmek için temel bir beceridir.[45]

12.5.2 Hata Ayıklayıcı (Debugger) Kullanımı

Hata ayıklayıcılar, malware analizinde vazgeçilmez araçlardır. Bir hata ayıklayıcı, analistin kodu talimat talimat çalıştırmasına ve programın çalışma zamanı davranışları üzerinde tam kontrol sahibi olmasına olanak tanır.[47] `x64dbg`[48], OllyDbg ve Immunity Debugger[49] gibi araçlar, karmaşık davranışları gözlemlemek ve anti-hata ayıklama tekniklerini atlatmak için kullanılır. Bir analist, *step over* (talimatın üzerinden geçerek çalıştırma) ve *step into* (işlevin içine girerek çalıştırma) gibi komutları kullanarak kod akışını hassas bir şekilde kontrol edebilir.[48] Bu, analiste şüpheli işlevleri atlama veya derinlemesine inceleme esnekliği sağlar.

12.5.3 Anti-Analiz Yöntemleri ve Bunları Aşma

Tersine mühendislik, malware yazarları ve analistler arasında süregelen bir "silahlanma yarışı"dır. Her yeni analiz tekniği, malware yazarlarının buna karşı koymak için yeni kaçınma mekanizmaları geliştirmesine neden olur.

- **Anti-Hata Ayıklama (Anti-Debugging):** Malware'in bir hata ayıklayıcıda çalıştığını tespit etmek için kullanılan tekniklerdir.[51] Bunlar, programın çalışma süresini kontrol etmek, bellek bütünlüğünü denetlemek veya donanım kesme noktalarını aramak gibi yöntemleri içerir.[51] Bir analist, bu tür kontrolleri aşmak için donanım kesme noktaları kullanabilir veya kötü amaçlı kodun algılama rutinlerini yama (*patch*) yaparak devre dışı bırakabilir.[52]
- **Anti-Sanallaştırma (Anti-Virtualization):** Malware, sanal bir makinede çalıştığını anladığında çalışmayı durdurabilir.[53] Bu tespit, belirli donanım talimatlarını sorgulamak (örneğin, VirtualBox için `VBoxVBoxVBox`[54]), sistemde kaç CPU çekirdeği olduğunu kontrol etmek veya sistem zamanlamalarını ölçmek gibi yöntemlerle yapılır. Bu teknikleri aşmak için, analist VM yapılandırmalarını değiştirerek fiziksel bir makineyi taklit edebilir veya API kancalama (*hooking*) yoluyla anti-VM işlevlerinin dönüş değerlerini manipüle edebilir.[54]

12.5.4 Kriptografik Uygulama Analizi

Malware, verileri korumak veya şifrelemek için genellikle kriptografik algoritmalar kullanır. Özellikle fidye yazılımları, AES ve RSA gibi güçlü ve standart algoritmaları kullanarak kurbanın dosyalarını şifreler.[7] Bir analist, hata ayıklayıcıları kullanarak, şifreleme öncesi aşamada bellekteki şifreleme anahtarlarını veya algoritma girdilerini gözlemleyebilir.[55] Bu, dosyaların şifreleme işlemi tamamlanmadan önce kurtarılmasını sağlayabilir. Bazı durumlarda, saldırganlar şifrelemeyi zorlaştırmak veya analizden kaçmak için basit bir XORlama gibi özel ("*custom*") ve zayıf algoritmalar kullanabilir.[55] Bu, analiste şifreleme mekanizmasını tersine mühendislikle çözerek dosyaları kurtarma fırsatı verebilir.

12.6 Gelişmiş Konular ve Olay Analizi

12.6.1 Malware Sınıflandırmasında Makine Öğrenimi

Makine öğrenimi, imza tabanlı tespitin sınırlılıklarını aşmak için malware analizinde giderek daha fazla kullanılmaktadır. Makine öğrenimi modelleri, daha önce hiç görülmemiş dosyalardaki kötü amaçlı desenleri ve davranışları tespit etmek için eğitilebilir.[3] Bu modeller, PE başlıklarından ve API çağrı dizilerinden çıkarılan özellikler gibi statik veya dinamik verileri kullanır.[43] Ancak, makine öğreniminin de kendi zorlukları vardır. Malware yazarları, algılama modellerini atlatmak için sürekli olarak yeni varyantlar yayınlar.[56] Modelin eğitildiği veri dağılımı sabit olmadığından, sürekli güncellenmesi gerekir. Bu durum, analistin makine öğrenimi çıktısını kesin bir doğruluk kaynağı olarak görmemesi, aksine insan analizi ve uzmanlığı ile birleştirmesi gerektiğini gösterir.

12.6.2 Kod ve Altyapı Tekrarı Analizi

Siber suç endüstrisinin profesyonelleşmesiyle birlikte, tehdit aktörleri verimlilik sağlamak için kod ve altyapıyı tekrar kullanmaktadır. Örneğin, siber suç gruplarının (örneğin Black Basta ve Qakbot) paylaşılan altyapıyı, komuta ve kontrol sunucularını veya hatta çekirdek kod parçalarını yeniden kullandığı gözlemlenmiştir.[57] Bu durum, malware geliştirmenin münferit bir çaba olmaktan çıkıp, tedarik zincirleri ve ortaklıklarla çalışan bir "hizmet olarak malware" (MaaS) modeline dönüştüğünü göstermektedir. Kod tekrarı analizi, bir analistin yeni görünen bir malware'in aslında bilinen bir ailenin veya tehdit aktörünün varyantı olduğunu anlamasını sağlar.[58]

12.6.3 Tehdit Aktörü Atfı ve Kampanya Analizi

Atıf (*Attribution*), bir siber saldırının arkasındaki belirli bir bireyi, grubu veya devleti tanımlama sürecidir.[59] Bu, sadece bir saldırıyı tespit etmekten daha fazlasıdır; saldırganın kimliğini, motivasyonunu ve gelecekteki olası eylemlerini anlamayı hedefler. Atıf, bir dizi karmaşık yöntemin bir araya gelmesiyle gerçekleştirilen bir süreçtir.[59]

- **Teknik Eserler:** Malware `hash`'leri, IP adresleri, alan adları ve e-posta başlıkları gibi dijital parmak izleri toplanır ve analiz edilir.[59]
- **Taktikler, Teknikler ve Prosedürler (TTPs):** Bir saldırganın belirli davranışsal kalıpları (örneğin, yanal hareket için PowerShell kullanması, veri sızdırma için bulut depolama hizmetlerini kullanması) belirlenir. Bu davranışlar, teknik göstergelerden daha zor taklit edilir ve MITRE ATT&CK gibi çerçevelerle eşleştirilebilir.[61]
- **Kod ve Altyapı Tekrarı:** Saldırganın farklı kampanyalarda aynı C2 sunucusunu veya aynı benzersiz kod parçacığını kullanıp kullanmadığı kontrol edilir.[59]
- **Tehdit İstihbarat Veri Kaynakları:** Yüksek kaliteli tehdit istihbarat beslemeleri, bilinen tehdit aktörü grupları (örneğin, APT29, FIN7) ve onların hedefleri hakkında bağlamsal bilgiler sağlar.[59]

Atıf süreci belirsiz olabilir; saldırganlar sahte bayrak (*false flag*) kullanabilir veya altyapıyı diğer aktörlerle paylaşabilir.[59] Bu nedenle, atıf genellikle bir "mozaik" olarak tanımlanır. Güvenilir bir sonuca ulaşmak için birden fazla kanıtın ve kaynağın birleştirilmesi gerekir.[59] Aşağıdaki tablo, bir saldırıyı kimin gerçekleştirdiği hakkında bir çıkarım yapmak için kullanılan kanıt ve atıf seviyelerini göstermektedir.

Atıf Seviyesi	**Açıklama**	**Gerekli Kanıt Türleri**	**Güven Derecesi**
Aktivite Kümesi	Ortak TTP'leri veya altyapıyı paylaşan ilişkili gözlemlere atıf	IP'ler, alan adları, URL'ler, benzersiz malware `hash`'leri	Düşük
Geçici Tehdit Grubu	Tek bir aktörün dahil olduğu, ancak isimlendirmeye yetecek kadar kanıtın olmadığı tutarlı aktivite	Tutarlı TTP'ler, araç takımları, operasyonel güvenlik (OPSEC) hataları	Orta
Adlandırılmış Tehdit Aktörü	Topluluk tarafından bilinen ve takip edilen, birden fazla kampanyada tutarlı davranışlar sergileyen aktöre atıf	Tutarlı ve sürekli operasyonlar, operasyonel hatalar, hedefler, motivasyon	Yüksek

12.7 Vaka İncelemeleri

12.7.1 WannaCry Fidye Yazılımı Saldırısı

2017 yılının Mayıs ayında ortaya çıkan WannaCry fidye yazılımı, dünya genelinde 150'den fazla ülkede 230.000'den fazla bilgisayarı etkileyen küresel bir salgına neden oldu.[62] Analiz, bu saldırının fidye yazılımı ve solucan özelliklerini birleştirdiğini ortaya koydu. WannaCry, eski Windows sürümlerindeki SMB (Sunucu Mesaj Bloğu) protokolü zafiyetini (EternalBlue) istismar ederek bir ağ içinde kendi kendine yayıldı.[62] Bu saldırı, en yaygın kullanılan işletim sistemlerindeki yamasız zafiyetlerin, basit bir fidye yazılımının nasıl küresel bir salgına dönüşebileceğini göstermesi açısından önemli bir örnektir.

12.7.2 Ryuk Fidye Yazılımı Kampanyası

Ryuk fidye yazılımı, büyük kuruluşları ve yüksek profilli hedefleri seçerek gerçekleştirdiği hedefli saldırılarla bilinir.[64] Ryuk, saldırı zincirinin başlangıcında tek başına kullanılmaz. Genellikle Emotet veya TrickBot gibi diğer malware'ler tarafından bir ağa sızmak için birincil bir vektör olarak teslim edilir.[64] İçeri girdikten sonra saldırgan, Adfind ve `nltest` gibi "*living-off-the-land binaries*" (LOLBins) olarak bilinen meşru sistem araçlarını kullanarak ağ keşfi yapar ve ayrıcalıklarını yükseltir.[64] Bu vaka, modern siber saldırıların çok aşamalı, hedefli ve karmaşık doğasını gösterir. Saldırganlar, tespit edilmekten kaçınmak için tek bir malware yerine, birden fazla aracı ve meşru sistem komutunu bir araya getirir.

12.7.3 Emotet Malware Ailesi

Emotet, başlangıçta bir bankacılık Truva atı olarak ortaya çıkan, ancak daha sonra diğer malware'leri dağıtmak için bir "altyapı olarak hizmet" (IaaS) sunan modüler bir botnete dönüşen sofistike bir malware ailesidir.[66] *Malspam* (kötü amaçlı e-posta) ile yayılır ve polimorfik yapısı nedeniyle imza tabanlı tespitlerden kaçabilir.[67] Emotet, sanal makineleri tespit etme yeteneğine sahiptir ve kalıcılık için kayıt defteri anahtarlarını veya zamanlanmış görevleri kullanır. Bu evrim, siber suç dünyasında bir "iş modeli" değişimini temsil eder ve analiste sadece bir malware'i analiz etmek yerine, arkasındaki ekonomik motivasyonları ve tedarik zincirlerini de anlaması gerektiğini gösterir.

Bölüm 13

SOSYAL MÜHENDİSLİK VE İNSAN FAKTÖRÜ

Giriş

Sosyal mühendislik ve insan faktörü, siber güvenlikte teknolojik savunmaların ötesinde insan psikolojisi ve davranışlarıyla ilgili kritik bir alandır. Bu bölümde sosyal mühendislik teknikleri, farkındalık eğitimleri ve insan odaklı güvenlik stratejilerini ele alacağız.

13.1 İnsan Psikolojisi ve Sosyal Mühendislik Temelleri

Sosyal mühendislik, insan davranışlarının ve psikolojik eğilimlerinin, bir kişiyi normalde yapmayacağı bir eylemi gerçekleştirmesi veya gizli bilgileri ifşa etmesi için manipüle edilmesi sanatıdır. Bu, teknolojik zafiyetlerden ziyade, insan doğasının öngörülebilir tepkilerine dayanan bir saldırı vektörüdür. Sosyal mühendisler, hedeflerinin güvenini kazanmak, korku veya aciliyet hissi yaratmak veya bir otorite figürünü taklit etmek gibi çeşitli taktikler kullanırlar.

13.1.1 Bilişsel Yanılgılar ve Karar Verme Zafiyetleri

İnsan beyninin karar verme süreci, Nobel Ödüllü psikolog Daniel Kahneman'ın "Sistem 1" ve "Sistem 2" teorisi ile açıklanır. Sistem 1, hızlı, sezgisel ve otomatik tepkiler verirken, Sistem 2 daha yavaş, analitik ve mantıksal düşünme süreçlerini yönetir. Siber tehditler karşısında, özellikle stres veya bilişsel aşırı yük altında kalan çalışanlar, karmaşık dijital risklere uyum sağlayamayan Sistem 1'i aktive etme eğilimindedir. Bu durum, güvenlik eğitimi almış kişilerin bile basit hatalar yapmasına, örneğin sahte bir bağlantıya tıklamasına yol açabilir.

Saldırganlar, insan beyninin bu iki sistem arasındaki zayıflıktan yararlanarak, belirli bilişsel yanılgıları hedeflerler:

- **Kötümserlikten Uzak Durma (Unrealistic Optimism):** Bu yanılgı, insanların "bu sadece başkasının başına gelir" inancını yansıtır. Çalışanlar, siber saldırı riskini kişisel olarak hafife alarak, güvenlik yönergelerini göz ardı etme veya şüpheli ekleri doğrulamama gibi pervasız davranışlar sergileyebilir.
- **Otorite Yanılgısı (Authority Bias):** Bireylerin, meşru olarak algılanan bir figürden gelen talimatlara sorgusuz uymasıdır. Saldırganlar, üst düzey yöneticileri veya tanınmış kurumları taklit ederek bu yanılgıdan yararlanır ve fon transferi veya hassas bilgi ifşası gibi aceleci eylemleri tetikler.
- **Tanıdıklık ve Alışkanlık Yanılgısı (Familiarity Bias):** İnsanların, tanıdık gelen bir şeye karşı gardını indirme eğilimidir. Yoğun bilgi akışı altında çalışan bir kişi, tanıdık bir isimden veya logodan gelen mesajın orijinalliğini doğrulamak yerine, hızlı bir yanıt vermeyi tercih edebilir.

- **Kullanılabilirlik Yanılgısı (Availability Bias):** Bir riskin olasılığını, yakın zamanda yaşanan deneyimlere dayanarak değerlendirme eğilimidir. Bir çalışan daha önce hiç oltalama saldırısıyla karşılaşmadıysa, bu tehdidi düşük bir tehlike olarak algılayabilir.
- **Olumsuz Duygular ve Bilişsel Yük:** Yorgunluk, stres ve bilişsel aşırı yük, analitik yetenekleri zayıflatarak muhakeme yeteneğini bozar. Bu durum, "acil" gibi terimler içeren e-postaları alışkanlıkla açmaya veya hızlı ve içgüdüsel kararlar vermeye neden olur.

13.1.2 Güvenlik Bağlamında Sosyal Psikoloji İlkeleri

Davranışsal psikolog Robert Cialdini'nin ikna prensipleri, sosyal mühendislik saldırganlarının manipülasyon taktiklerinin temelini oluşturur. Bu prensipler, insan davranışını yönlendiren evrensel eğilimleri kullanır ve saldırganların bu ilkelerden nasıl yararlandığı aşağıda detaylandırılmıştır:

- **Karşılık Verme (Reciprocity):** İnsanlar, bir iyiliğe veya hediyeye karşılık verme eğilimindedir. Saldırganlar, kurbanlarına "ücretsiz bir araç" veya "yardımcı bir bilgi" sunarak bir borçluluk duygusu yaratır ve bu sayede daha büyük bir talep için zemin hazırlarlar.
- **Taahhüt ve Tutarlılık (Commitment and Consistency):** Bir kez küçük bir eylemde bulunan bireyler, daha sonra bu davranışlarıyla tutarlı kalmaya daha yatkın olurlar. Saldırganlar, masum bir taleple başlayıp, kademeli olarak daha hassas bilgi taleplerine geçerek bu prensibi kullanır.
- **Sosyal Kanıt (Social Proof):** Bir eylemin veya fikrin başkaları tarafından da benimsendiğine dair kanıt görmek, uyma eğilimini artırır. Saldırganlar, sahte referanslar veya "meslektaşlarınızın %90'ı zaten şifrelerini güncelledi" gibi mesajlar kullanarak bu prensibi istismar eder.
- **Otorite (Authority):** İnsanlar, yetkili olarak algılanan figürlerden gelen isteklere boyun eğmeye daha yatkındır. Saldırganlar, CEO'ları, IT yöneticilerini veya tanınmış kurumları taklit ederek bu prensipten yararlanır.
- **Beğenme (Liking):** Bireyler, tanıdık, benzer veya kendilerinden biri olarak gördükleri kişilere daha kolay güvenir. Saldırganlar, ortak ilgi alanlarını veya sosyal bağlantıları kullanarak güven oluşturur.
- **Kıtlık (Scarcity):** Bir şeyin sınırlı sayıda veya sürede olduğuna inanıldığında, ona olan talep artar. "Yalnızca 5 koltuk kaldı" veya "bu özel erişim 24 saat içinde sona eriyor" gibi aciliyet ifadeleri, hızlı ve düşüncesiz kararlar almaya yol açar.

13.1.3 Güven Oluşturma ve İkna Teknikleri

Güven, bir sosyal mühendislik saldırısının temelini oluşturur ve saldırganlar, sahte bir kimlik (pretext) oluşturarak bu güveni inşa eder. Bu süreç, hedefin adını, çalıştığı departmanı, teknoloji kullanım alışkanlıklarını ve hatta kişisel/profesyonel ilişkilerini bilmek gibi detayları içerir. Bu bilgilerle, saldırganlar inandırıcı bir senaryo hazırlar ve kurbanın şüpheciliğini bypass eder.

Daha derinlemesine ikna teknikleri arasında, hedefe sempati duymak, karşılıklılık ilkesini kullanmak ve egoyu okşamak bulunur. Örneğin, bir saldırgan, bir aile acil durumuyla ilgili yardım istiyormuş gibi yaparak hedefin sempati duygusunu harekete geçirebilir veya küçük bir iyilik yaparak bir borçluluk bağı kurabilir.

13.1.4 Psikolojik Profilleme ve Hedef Seçim Metotları

Bir sosyal mühendislik saldırısının ilk ve en kritik adımı, hedef hakkında kapsamlı bilgi toplamaktır. Bu süreç, pasif ve aktif bilgi toplama olarak ikiye ayrılır. Pasif bilgi toplama, hedeften bağımsız olarak halka açık verilerin analizini içerir. Sosyal medya platformları (LinkedIn, Facebook, Twitter) bu aşamada kritik bir rol oynar, çünkü saldırganlar hedefin ilgi alanları, işi ve ailesi hakkında değerli bilgiler edinebilir ve bu bilgiler kişiselleştirilmiş bir saldırı için kullanılabilir. Aktif bilgi toplama ise hedefle doğrudan etkileşimi gerektirir, ancak bu, gündelik bir sohbet stratejisiyle yapılır ve hedefin sorgulanıyormuş gibi hissetmesi engellenir.

Sosyal mühendislik saldırılarının başarısı, yalnızca teknolojik zafiyetlere değil, aynı zamanda bilişsel yanılgıları istismar eden ve duygusal tepkileri tetikleyen ikna prensiplerine de dayanır. Saldırgan, bilişsel yanılgıları istismar eden ikna prensiplerini kullanarak, hedefin beynindeki hızlı ve sezgisel karar veren Sistem 1'ini aktive eder. Bu, hedefin mantıksal doğrulama süreçlerini (Sistem 2) atlamasını sağlar. Bu zincirleme reaksiyon, saldırganın az çabayla büyük bir başarı elde etmesini mümkün kılar. Örneğin, bir "CEO dolandırıcılığı" senaryosu, otorite yanılgısını kullanarak bir finans çalışanının normalde yapmayacağı bir fon transferini gerçekleştirmesini sağlayabilir.

İnsan beynindeki bu "özelliklerin" istismarı, savunma stratejilerinin sadece teknik kontrollere (güvenlik duvarı, anti-virüs yazılımları) odaklanamayacağını, aynı zamanda psikolojik güvenliği (PsySec) de içermesi gerektiğini gösterir. Sosyal mühendislik, sadece bilgi çalma girişimi değil, aynı zamanda bir psikolojik durum manipülasyonudur. Saldırgan, önce halka açık kaynaklardan (OSINT) bilgi toplayarak bir hedef belirler. Ardından, aciliyet duygusu yaratan bir senaryo (örneğin, bir hesabın askıya alınma tehdidi) tasarlar. Bu senaryo, bilişsel olarak aşırı yüklenmiş bir çalışanı hızlı ve düşüncesiz bir eyleme sevk eder. Bu, sosyal mühendisliğin neden bu kadar tehlikeli ve kalıcı bir tehdit olduğunu açıklar.

Psikolojik İlke / Bilişsel Yanılgı	**Kısa Tanım**	**Sosyal Mühendislik Taktikleri**	**Pratik Saldırı Senaryosu**
Otorite Yanılgısı	Bireylerin yetkili figürlerin talimatlarına uyma eğilimi.	CEO, IT yöneticisi veya tanınmış kurumları taklit etme.	Bir finans çalışanına, CEO'dan geldiği iddia edilen bir e-posta ile acil fon transferi talimatı gönderilmesi.
Kıtlık	Bir şeyin sınırlı olduğu algısının talebi artırması.	"Sınırlı sayıda," "yalnızca bugün" gibi aciliyet ifadeleri kullanma.	"Hesabınızın güvenliği tehlikede, 1 saat içinde şifrenizi güncellemezseniz hesabınız askıya alınacak."
Alışkanlık Yanılgısı	Tanıdık gelen bir şeye karşı gardını indirme eğilimi.	Sahte e-posta veya web sitelerinde tanıdık marka veya logo kullanımı.	Microsoft veya banka logosu içeren, gerçekle birebir aynı görünen bir oturum açma sayfası.
Karşılık Verme	Bir iyiliğe karşılık verme yükümlülüğü hissetme.	Ücretsiz bir hediye, indirim veya yardımcı bir araç sunma.	"Ücretsiz bir güvenlik aracı" indirme bağlantısı içeren bir e-posta gönderilmesi; bu aracın aslında kötü amaçlı yazılım olması.

Tablo 13.1: Bilişsel Yanılgılar ve İkna İlkeleri Karşılaştırması

13.2 Dijital Sosyal Mühendislik Saldırı Teknikleri

Dijital sosyal mühendislik, teknolojiyi kullanarak insanları manipüle etme ve kandırma sürecidir. Bu saldırılar, genellikle e-posta, sosyal medya, anlık mesajlaşma ve diğer dijital iletişim kanalları aracılığıyla gerçekleştirilir. Saldırganlar, hedeflerini kötü amaçlı yazılım indirmeye, sahte web sitelerine kişisel bilgilerini girmeye veya yetkisiz para transferleri yapmaya ikna etmek için çeşitli taktikler kullanırlar.

13.2.1 Oltalama (Phishing) ve Hedef Odaklı Oltalama (Spear-phishing) Kampanya Tasarımı

– **Genel Oltalama (Phishing):** Geniş bir kitleye gönderilen, güvenilir bir kaynaktan (banka, e-posta sağlayıcısı) geliyormuş gibi görünen dolandırıcılık girişimleridir. Saldırının amacı, kurbanın kişisel bilgilerini veya kimlik bilgilerini ele geçirmek için kötü amaçlı bir bağlantıya tıklamasını veya sahte bir web sitesine

yönlendirilmesini sağlamaktır. Geleneksel olarak imla hataları gibi bariz işaretlerle tanınsa da, günümüzde bu saldırılar çok daha sofistike ve inandırıcıdır.

– **Hedef Odaklı Oltalama (Spear-phishing):** Belirli bir kişiyi veya grubu hedef alan, çok daha kişisel ve incelikli bir oltalama türüdür. Saldırgan, hedefin sosyal medyadaki dijital ayak izini (LinkedIn, Facebook, vb.) kapsamlı bir şekilde araştırarak, hedefin bir meslektaşından veya yöneticisinden geliyormuş gibi görünen ikna edici e-postalar hazırlar. Bu kişiselleştirme, kurbanın mesajın meşruiyetine inanma olasılığını büyük ölçüde artırır.

13.2.2 İş E-postası Ele Geçirme (BEC) ve CEO Dolandırıcılığı

– **BEC (Business Email Compromise):** Finansal işlemleri hedef alan, sosyal mühendislik tabanlı bir e-posta saldırısıdır. Bu saldırılar, kötü amaçlı bir bağlantı veya ek içermemesiyle diğer oltalama türlerinden ayrılır ve bu sayede geleneksel e-posta filtrelerini atlatabilir. Saldırgan, bir yöneticiyi veya üçüncü taraf bir satıcıyı taklit ederek, hedefe bir fon transferi yapması talimatını verir.

– **CEO Dolandırıcılığı:** BEC'nin, özellikle üst düzey bir yöneticiyi (CEO, CFO) taklit etmeye odaklanan bir türüdür. Saldırgan, yöneticinin e-posta adresini taklit edebilir veya yazışma stilini taklit ederek mesajın inandırıcılığını artırır. Talebin aciliyeti vurgulanır ve hedefin normal doğrulama süreçlerini atlaması hedeflenir.

13.2.3 Vishing (Sesli Oltalama) ve Smishing (SMS Oltalama)

– **Vishing:** Telefon görüşmeleri aracılığıyla gerçekleştirilen bir oltalama türüdür. Saldırganlar, bankalar, hükümet kurumları veya teknik destek temsilcileri gibi meşru kuruluşları taklit ederek kurbandan hassas bilgileri veya kredi kartı detaylarını doğrudan elde etmeyi amaçlar.

– **Smishing:** SMS veya anlık mesajlaşma uygulamaları (WhatsApp, Telegram) üzerinden gerçekleştirilen oltalama türüdür. Saldırgan, kötü amaçlı bir bağlantı içeren mesajlar gönderir ve insanlar SMS'lere hızlı tepki verme eğiliminde oldukları için smishing oldukça etkilidir. Ortak senaryolar arasında banka sahtekarlığı uyarıları, kargo teslimat bildirimleri veya sahte ödül bildirimleri bulunur.

13.2.4 Sosyal Medya Manipülasyonu ve Sahte Profiller

Sosyal medya manipülasyonu, sahte profiller oluşturarak veya mevcut hesapları ele geçirerek kurbanın güvenini kazanmayı amaçlayan bir dijital kimlik hırsızlığı biçimidir. Saldırganlar, kişisel bilgileri (isim, resim, konum) çalarak gerçeğe yakın sahte profiller oluşturur ve bu profilleri, kurbanın çevresindeki kişilerle etkileşim kurarak bilgi toplamak ve spear-phishing veya başka bir saldırı için gerekli istihbaratı elde etmek için kullanırlar.

13.2.5 Derin Sahte (Deepfake) ve Yapay Zeka Üretimi İçerik Kullanımı

– **Deepfake:** Yapay zeka kullanılarak oluşturulan ve gerçekçi görünen sahte video, ses veya görsel içeriklerdir. Bu teknoloji, bir kişinin sesini taklit ederek (ses klonlama) veya yüzünü başka bir videodaki bir kişinin üzerine bindirerek kullanılır.

– **Kullanım Alanları:** Saldırganlar, bir CEO'nun sesini klonlayarak bir finans çalışanına fon transferi talimatı vermek için kullanabilir. Bu tür saldırılar, sesin veya görüntünün anlık doğrulamasının zorluğundan yararlanır.

– **Teknik Adımlar (Ses Klonlama):**

1. **Veri Toplama:** Kurbanın sesinden örnekler (sosyal medya videoları, sesli mesajlar, röportajlar) toplanır. Birkaç saniyelik bir ses örneği bile yeterli olabilir.

2. **Model Eğitimi:** Toplanan ses örnekleri, yapay zeka ses sentezi modellerine (Tacotron 2, Vall-E gibi) beslenir. Model, kurbanın sesindeki benzersiz özellikleri (tonlama, ritim, nefes alma kalıpları) öğrenir.
3. **Sahte Ses Üretimi:** Saldırgan, istediği metni yazar ve model, bu metni kurbanın sesini kullanarak okur. Bu, önceden oluşturulmuş bir ses dosyası olarak kullanılabileceği gibi, gerçek zamanlı bir telefon görüşmesinde de uygulanabilir.

Geleneksel olarak, kitlesel oltalama (phishing) geniş bir kitleyi hedef alırken, hedef odaklı oltalama (spear-phishing) yüksek başarı oranı için daha fazla çaba gerektirir. Yapay zeka bu çelişkiyi ortadan kaldırmaktadır. Saldırganlar artık, açık kaynak istihbarat araçlarıyla topladıkları verileri yapay zeka dil modellerine besleyerek, binlerce kişiye aynı anda gönderilebilen, dilbilgisi hatasız ve son derece kişiselleştirilmiş saldırılar oluşturabilirler. Bu yeni yaklaşım, geleneksel savunma yöntemlerini (imla hatalarını kontrol etme gibi) etkisiz hale getirir ve savunma paradigmalarının doğrulamaya ve sıfır güvene doğru kaymasını hızlandırır.

Ayrıca, siber suç ekosistemi de giderek daha karmaşık hale gelmektedir. Oltalama (phishing) kitlerinin (hazır oltalama web siteleri) incelenmesi, bu kitleri yazan geliştiricinin, kitleri kullanan saldırganlardan gizlice çaldıkları verilerin bir kopyasını kendilerine yönlendirdiğini göstermektedir. Bu, siber suç tedarik zincirindeki karmaşıklığı ve güven eksikliğini ortaya koymaktadır.

13.3 Fiziksel Sosyal Mühendislik ve OSINT

Fiziksel sosyal mühendislik, bir saldırganın hedefine fiziksel olarak yaklaşarak bilgi toplaması veya yetkisiz erişim sağlamasıdır. Bu, genellikle bir binaya veya kısıtlı bir alana sızmayı içerir. Açık Kaynak İstihbaratı (OSINT), halka açık kaynaklardan bilgi toplama sürecidir. Sosyal mühendisler, hedefleri hakkında bilgi toplamak ve saldırılarını daha inandırıcı hale getirmek için OSINT tekniklerini sıklıkla kullanırlar.

13.3.1 Arkadan Girme (Tailgating), Omuzdan Gözetleme (Shoulder Surfing) ve Fiziksel Sızma

– **Arkadan Girme (Tailgating):** Bir yetkisiz kişinin, yetkili bir kişiyi takip ederek kısıtlı bir alana girmesidir. Saldırgan, ”kapıyı açık tutar mısınız?” gibi kibarlığa dayalı bir sosyal normu istismar eder. Ellerinde paketler bulunan bir kurye gibi davranmak veya sahte bir kimlik kartı gösterme taklidi yapmak, bu tekniğin yaygın örneklerindendir.

– **Omuzdan Gözetleme (Shoulder Surfing):** Bir kişinin şifre, PIN veya diğer hassas bilgileri girerken omzunun üzerinden gizlice izlemesi veya kaydetmesidir. Bu, kafeler veya havaalanları gibi kamusal alanlarda çok yaygın, düşük teknolojili ancak etkili bir saldırıdır.

– **Fiziksel Sızma:** Daha genel bir kavram olup, saldırganın sahte bir kimlikle (pretext) bir binaya veya kısıtlı alana girmesini içerir. Bir merdiven veya alet kemeri taşımak gibi basit eylemler bile, saldırganın ”ait olduğu” izlenimini yaratmasına yardımcı olur.

13.3.2 Çöp Karıştırma (Dumpster Diving) ve Fiziksel Bilgi Toplama

Çöp karıştırma, atılan fiziksel veya dijital atıklardan bilgi çıkarma eylemidir. Bu, ilkel bir yöntem gibi görünse de, saldırganlara paha biçilmez veriler sağlayabilir. Bu veriler arasında, müşteri listeleri, organizasyon şemaları, şifrelerin yazılı olduğu not defterleri, hatta eski kimlik kartları bulunabilir. Elde edilen bilgiler, hedef odaklı oltalama (spear-phishing) e-postaları hazırlamak için kullanılabilir. Bir organizasyon şemasının bulunması, saldırganın doğru kişileri hedeflemesini ve inandırıcı bir senaryo oluşturmasını kolaylaştırır. Bu, düşük teknolojili bir yöntemin nasıl yüksek etkili bir siber saldırının öncüsü olabileceğini gösterir.

13.3.3 Ön Metin Oluşturma (Pretexting) ve Telefon Tabanlı Sosyal Mühendislik

Pretexting, saldırganın kurbandan bilgi almak veya eylem yaptırmak için uydurulmuş bir senaryo (pretext) oluşturmasıdır. Saldırgan, IT teknisyeni, banka temsilcisi veya meslektaş gibi güvenilir bir figürü taklit eder. Bu senaryo, hedefin şüphelenmesini engellemek için mantıklı ve acil görünmelidir. Telefon tabanlı sosyal mühendislik, pretexting'in telefon üzerinden uygulanmasıdır. Saldırgan, arayanın numarasını (caller ID) sahte gösterebilir.

13.3.4 Açık Kaynak İstihbaratı (OSINT) Toplama ve Sosyal Medya Keşfi

Açık Kaynak İstihbaratı (OSINT), halka açık, ücretsiz kaynaklardan (arama motorları, sosyal medya, DNS kayıtları) istihbarat toplama sürecidir. Bu, sosyal mühendislik saldırısının hazırlık aşamasında kritik bir adımdır.

- **theHarvester:** Alan adları, alt alan adları (subdomains), e-posta adresleri ve IP'ler gibi halka açık bilgileri toplamak için kullanılan yaygın bir araçtır.
  - **Kullanım Örneği:** `theharvester -d example.com -b all` komutu, `example.com` alan adıyla ilişkili tüm halka açık bilgileri toplar.
- **Maltego:** Bilgileri görsel bir grafik üzerinde ilişkilendiren bir veri madenciliği aracıdır. Sosyal medya profillerini, e-posta adreslerini ve ilgili altyapıları haritalandırarak saldırı için hedef seçimi ve kişiselleştirme için gerekli bilgileri sağlar.
- **Social-Engineer Toolkit (SET):** Sosyal mühendislik saldırıları için tasarlanmış açık kaynaklı, Python tabanlı bir araçtır. Oltalama, spear-phishing ve diğer saldırı vektörleri için hazır modüller sunar.
  - **Kullanım Örneği:** `sudo apt install set` komutu ile kurulum yapılır ve ardından `setoolkit` komutu ile çalıştırılır.

Sosyal mühendislik, siber veya fiziksel olmak üzere tek bir kategoriye ayrılamaz. Saldırganlar her iki alanı birleştiren bir "hibrit yaklaşım" kullanır. Saldırgan, bir çalışanın sosyal medya profilinden (dijital OSINT) adını ve rolünü öğrenir. Ardından, bir çöp karıştırma eylemiyle (fiziksel) o kişinin eski bir kimlik kartını veya dahili bir organizasyon şemasını bulur. Bu bilgilerle, IT teknisyeni kılığında ofise fiziksel olarak sızar veya hedefin telefonuna "teknik bir sorunu düzeltmek" için pretexting yapar. Bu zincirleme eylemler, saldırının tek bir zafiyeti değil, bir dizi zafiyeti istismar ettiğini gösterir. Bir güvenlik sisteminin en zayıf halkası, genellikle en düşük teknolojili saldırıya maruz kalan bileşenidir. Bir çalışanın çöpe attığı şifre notu, milyarlarca dolarlık bir şirketin güvenlik duvarını atlatmak için yeterli olabilir. Bu nedenle, savunma stratejilerinin bütünsel bir bakış açısıyla tasarlanması gerekir.

Araç Adı	Açıklama	Temel Kullanım Senaryosu	Komut Satırı / Kullanım Örneği
theHarvester	Halka açık kaynaklardan e-posta, alt alan adı, IP ve host bilgilerini toplar.	Bir şirket hakkında ilk aşama keşif (footprinting) yapmak.	`theharvester -d example.com -b a`
Maltego	Açık kaynak verilerini görsel bir grafikte ilişkilendirir, gizli bağlantıları ortaya çıkarır.	Sosyal medya profilleri, e-posta adresleri ve ilişkili altyapıları haritalandırma.	`maltego` komutu ile arayüz başlatılır, ardından "Company Stalker" makinesi ile hedef alan adı girilir.

Araç Adı	Açıklama	Temel Kullanım Senaryosu	Komut Satırı / Kullanım Örneği
Social-Engineer Toolkit (SET)	Oltalama, spear-phishing ve diğer sosyal mühendislik saldırıları için hazır modüller sunan Python tabanlı bir araç.	Gerçekçi oltalama e-postaları ve sahte web siteleri oluşturma.	`sudo apt install set` ile kurulum yapılır, ardından `setoolkit` komutu ile çalıştırılır.

Tablo 13.2: OSINT Araçları ve Kullanım Alanları

13.4 Kurumsal Sosyal Mühendislik Zafiyetleri

Kuruluşlar, hem teknolojik hem de insan kaynaklı çeşitli sosyal mühendislik zafiyetlerine sahiptir. Bu zafiyetler, çalışanların güvenlik farkındalığı eksikliğinden, zayıf güvenlik politikalarına ve yetersiz erişim kontrollerine kadar uzanabilir. Sosyal mühendisler, bu zafiyetleri istismar ederek, bir kuruluşun ağına sızabilir, hassas verileri çalabilir veya iş süreçlerini kesintiye uğratabilir.

13.4.1 Çalışan Güvenlik Farkındalığı Açık Değerlendirmesi

Güvenlik açık değerlendirmesi, bir organizasyonun mevcut güvenlik duruşunu, belirlenmiş standartlara göre değerlendiren sistematik bir süreçtir. Bu değerlendirme, çalışanların güvenlik farkındalığındaki boşlukları belirlemek için anketler, röportajlar ve belge incelemeleri gibi yöntemleri kullanır.

- **Değerlendirme Aşamaları:**
  1. **Kapsam Tanımlama:** Hangi sistemlerin, departmanların veya veri türlerinin değerlendirileceği belirlenir.
  2. **Veri Toplama:** Personel ile yapılan görüşmeler ve anketler, çalışanların davranışları hakkında nitel veriler sağlar.
  3. **Risk Değerlendirmesi:** Toplanan veriler, nicel veya nitel yöntemlerle analiz edilerek potansiyel riskler belirlenir.
  4. **Açık Tespiti:** Belirlenen riskler, NIST veya CIS gibi bilinen güvenlik çerçeveleriyle karşılaştırılarak mevcut zafiyetler ortaya konur.
  5. **Eylem Planı ve Raporlama:** Tespit edilen açıklar için önceliklendirilmiş bir eylem planı oluşturulur. Bu plan, belirli adımları, zaman çizelgelerini ve sorumlu kişileri içermelidir.

13.4.2 İç Tehdit Göstergeleri ve Davranışsal Analiz

İç tehditler, organizasyon içinden gelen risklerdir ve iki ana kategoriye ayrılır: Kötü niyetli (malicious) ve kasıtsız (unintentional) tehditler. Kasıtsız tehditler, dikkatsizlik veya kötü muhakeme sonucunda ortaya çıkar ve aslında çoğu siber olay insan hatasından kaynaklanır (%75-95). Bir iç tehdidin dijital ve fiziksel davranışları, potansiyel bir riskin anahtar göstergeleridir.

- **Davranışsal Göstergeler:**
  * **Alışılmadık Erişim Kalıpları:** Bir çalışanın normalde yapmadığı saatlerde, konumlardan veya cihazlardan oturum açması.
  * **Veri Sızdırma Girişimleri:** Geleneksel eşiklerin altında kalacak şekilde çoklu küçük dosya transferleri gibi alışılmadık boyutta veri hareketleri.

    - **Yetki Kötüye Kullanımı:** Bir çalışanın işi için gerekli olmayan sistemlere veya verilere erişim talepleri veya bu yönde davranışlar sergilemesi.
- **Tespit Teknolojileri:**
    - **Kullanıcı ve Varlık Davranış Analizi (UEBA):** Kullanıcıların normal davranışlarını izleyen ve bir temel çizgi (baseline) oluşturan ileri düzey bir güvenlik yaklaşımıdır. Bu temel çizgiden sapan her türlü anormallik, potansiyel bir tehdit olarak işaretlenir.
    - **Veri Kaybı Önleme (DLP):** Hassas verilerin organizasyon dışına izinsiz transferini engellemeyi amaçlayan bir stratejidir.

İç tehdit, sadece bir teknoloji sorunu değil, aynı zamanda psikolojik ve örgütsel bir sorundur. Bir çalışanın yetkilerini aşması veya olağandışı saatlerde oturum açması, finansal stres, hayal kırıklığı veya intikam arzusu gibi duygusal motivasyonlardan kaynaklanabilir. Bu nedenle, bir iç tehdit programı sadece teknolojik izleme araçlarını değil, aynı zamanda çok disiplinli bir tehdit yönetimi ekibini de içermelidir. Bu ekip, bir tehdidin sadece ne yaptığını değil, aynı zamanda neden yaptığını da analiz etmelidir.

İç Tehdit Davranışı	Davranışsal Göstergeler	Tespit Teknolojileri	Tespit Mekanizması
Yetki Kötüye Kullanımı	Bir çalışanın işi için gerekli olmayan kritik sistemlere veya verilere erişim denemeleri.	UEBA, SIEM	Kullanıcıların rolleri ve normal davranışları için bir temel çizgi oluşturarak anormallikleri tespit etme.
Veri Sızdırma Girişimi	Çok sayıda küçük dosyanın, geleneksel eşiklerin altında kalacak şekilde dış e-postalara veya harici depolama cihazlarına transferi.	DLP, UEBA	Hassas verilerin sınıflandırılması ve bu verilerin organizasyon dışına izinsiz çıkışını engelleme.
Alışılmadık Erişim	Bir çalışanın normalde oturum açmadığı saatlerde (örn. gece 3'te) veya konumlardan (örn. yurtdışından) oturum açma girişimleri.	UEBA, SIEM	Kullanıcının normal erişim kalıplarını öğrenerek, bu kalıplardan sapan her türlü davranışı işaretleme ve risk skorlaması yapma.

Tablo 13.3: İç Tehdit Göstergeleri ve Tespit Yöntemleri

13.4.3 Yönetici Hedefleme (Whaling) ve Yüksek Değerli Hedeflere Yönelik Saldırılar

Whaling, yüksek rütbeli bir yöneticiyi (CEO, CFO) hedef alan son derece kişiselleştirilmiş bir oltalama saldırısıdır. Saldırganlar, hedefin finansal hesaplara, maaş bordrosu bilgilerine ve gizli verilere erişimini kullanmayı amaçlar. Saldırganlar, bir CEO'ya odaklanarak tüm şirketin fonlarına veya gizli verilerine erişim elde edebilirler. Bu, saldırganların "en az çaba en çok etki" prensibini nasıl uyguladığının bir örneğidir.

13.4.4 Üçüncü Taraf ve Tedarik Zinciri Sosyal Mühendislik

Saldırganlar, bir organizasyonun tedarikçileri, iş ortakları veya danışmanları üzerinden bilgi sızdırmaya çalışabilir. Tedarikçi E-posta Ele Geçirme (VEC), karmaşık tedarik zincirlerine sahip sektörlerde önemli bir tehdit oluşturmaktadır. Bu saldırılar, bir üçüncü taraf satıcının ödeme talimatlarını güncelliyormuş gibi görünerek finansal dolandırıcılıkları hedefler.

13.4.5 Uzaktan Çalışma Ortamı Sosyal Mühendislik Riskleri

Hibrit veya uzaktan çalışma ortamları, sosyal mühendislik saldırıları için yeni riskler yaratır. Çalışanların ev ağlarında veya ortak alanlarda çalışması, teknik güvenlik kontrollerini aşmayı kolaylaştırır. Bu durum, kimlik doğrulama, şifre yönetimi ve cihaz güvenliği gibi konularda ek zafiyetler yaratır.

13.5 Sosyal Mühendislik Savunma Stratejileri

Sosyal mühendislik saldırılarına karşı savunma, yalnızca teknolojik kontrollere dayanmaz; aynı zamanda güçlü güvenlik politikaları, sürekli çalışan eğitimi ve proaktif bir güvenlik kültürü gerektirir. Bir kuruluş, sosyal mühendislik tehditlerine karşı çok katmanlı bir savunma stratejisi benimseyerek, insan faktöründen kaynaklanan riskleri önemli ölçüde azaltabilir.

13.5.1 Güvenlik Farkındalığı Eğitim Programı Tasarımı

Etkili bir güvenlik eğitim programı, riskleri azaltmak ve proaktif bir güvenlik yaklaşımı benimsemek için çok önemlidir.

- **Tasarım Aşamaları:**
  1. **Üst Yönetim Desteği (Executive Buy-in):** Program için bütçe ve kaynak sağlamak ve güvenlik farkındalığını şirket önceliği haline getirmek için üst yönetimin desteği alınmalıdır.
  2. **İhtiyaç Analizi:** Çalışanların güvenlik bilgilerindeki boşlukları (gaps) belirlemek için anketler ve başlangıç oltalama simülasyonları yapılır.
  3. **Hedeflerin Belirlenmesi:** Programın amaçları, somut ve ölçülebilir hedefler (örn. tıklama oranlarını düşürme, raporlama oranını artırma) olarak tanımlanır.
  4. **Uygulama ve Katılım:** Programın iç tanıtımı yapılır ve gamification (oyunlaştırma) gibi stratejilerle katılım teşvik edilir.
  5. **Ölçüm ve Özelleştirme:** Programın etkinliği, tamamlanma oranları ve raporlama metrikleri gibi verilerle düzenli olarak ölçülür. Yüksek riskli çalışanlar için özel eğitimler tasarlanır.

13.5.2 Oltalama Simülasyonu ve Ölçüm Programları

Oltalama simülasyonları, çalışanları gerçek dünya oltalama girişimlerine maruz bırakarak, güvenli bir ortamda öğrenmelerini sağlayan pratik bir eğitim bileşenidir. Bu programlar, geleneksel eğitimlerden çok daha etkilidir.

- **Uygulama Adımları:**
  1. **Planlama:** Simülasyonun hedefleri (eğitim, zafiyet seviyesini belirleme) ve kapsamı (hedef kitle, senaryo karmaşıklığı) belirlenir.
  2. **Senaryo Oluşturma:** Gerçekçi e-posta şablonları (sahte fatura, şifre sıfırlama talebi) hazırlanır. İçerik, hedef departmanın sorumluluklarına göre özelleştirilir.
  3. **E-postaların Gönderilmesi:** Şüphe uyandırmamak ve gerçek bir saldırıyı taklit etmek için e-postaların dağıtımı rastgele yapılır.
  4. **Yanıtların İzlenmesi:** Tıklama, indirme, kimlik bilgilerini girme veya e-postayı raporlama gibi kullanıcı etkileşimleri izlenir.
  5. **Geribildirim ve Eğitim:** Saldırıya maruz kalan çalışanlara, atladıkları ipuçlarını (red flags) gösteren anında geri bildirim ve ek eğitim materyalleri sunulur.

13.5.3 Güvenlik Kültürü Geliştirme ve Davranış Değişikliği

Etkili bir güvenlik programının nihai hedefi, çalışanların bilgiyi davranışa dönüştürdüğü kalıcı bir güvenlik kültürü oluşturmaktır. Bu kültürel değişim, sadece tıklama oranlarını düşürmeyi değil, aynı zamanda proaktif bir raporlama mekanizması oluşturmayı da içerir.

- **Kültür Değişimi için Anahtar Faktörler:**
  - **Pozitif Takviye (Positive Reinforcement):** Hata yapanları cezalandırmak yerine, şüpheli e-postaları rapor edenleri ödüllendirmek gerekir. Oyunlaştırma (gamification) ve liderlik tablosu gibi mekanizmalar, olumlu duygular ve motivasyon yaratır.
  - **Psikolojik Güvenlik:** "Hata yapma hakkı" tanınan bir ortam yaratmak, çalışanların cezalandırma korkusu olmadan hatalarını bildirmelerini sağlar. Bu, bir saldırının tespiti ve müdahalesini geciktiren korku tabanlı bir kültürün aksine, olumlu bir etki yaratır.
  - **Sürekli Pratik:** Öğrenmelerin pekişmesi için simülasyonların sık sık (örneğin, her 10 günde bir) tekrarlanması gerekir.
  - **İlgililik ve Özelleştirme:** Eğitim materyallerinin her çalışan rolüne ve yaşadığı gerçek senaryolara göre uyarlanması katılımı artırır.

Metrik	Ölçüm Yöntemi	Güvenlik Kültürü İçin Anlamı	Hedef
Tıklama Oranı	Tıklama yapan kullanıcı sayısı / Toplam kullanıcı sayısı	Çalışanların saldırılara karşı genel duyarlılığı ve farkındalık seviyesi.	Zaman içinde tutarlı bir şekilde düşüş göstermesi.
Raporlama Oranı	Şüpheli e-postayı raporlayan kullanıcı sayısı / Tıklama yapan kullanıcı sayısı	Çalışanların güvenlik sürecine aktif katılımı ve psikolojik güvenlik seviyesi.	Sürekli artış göstermesi.
Olayı Bildirme Süresi	E-postanın gönderildiği an ile ilk raporlamanın yapıldığı an arasındaki süre.	Tehdide karşı organizasyonun reaksiyon hızı.	Mümkün olduğunca kısalması.

Tablo 13.4: Oltalama Simülasyonu Başarı Metrikleri ve Anlamları

13.5.4 Teknik Kontroller: E-posta Filtreleme, Web Koruması

Güvenlik farkındalığı eğitimleri tek başına yeterli değildir, teknik önlemlerle desteklenmelidir. E-posta filtreleme sistemleri, kötü amaçlı yazılım içeren ekleri veya oltalama bağlantılarını otomatik olarak engelleyerek saldırganların kullanıcıya ulaşmasını zorlaştırır. Çok Faktörlü Kimlik Doğrulama (MFA) kullanımı, çalınan kimlik bilgilerinin etkisini büyük ölçüde azaltır.

13.5.5 Olay Bildirme ve Müdahale Mekanizmaları

Etkili bir savunmanın son adımı, bir saldırı meydana geldiğinde nasıl hareket edileceğine dair net bir plana sahip olmaktır. Bu plan, olay bildirme protokollerini, müdahale ekibinin rollerini ve sorumluluklarını içermelidir. Olay müdahale planları, gerçek bir olay meydana gelmeden önce düzenli olarak test edilmelidir.

13.6 İleri Seviye Sosyal Mühendislik ve Gelecek Tehditler

Sosyal mühendislik, sürekli olarak gelişen ve daha sofistike hale gelen bir tehdittir. Gelecekteki sosyal mühendislik saldırılarının, yapay zeka (AI), makine öğrenimi (ML) ve deepfake gibi teknolojileri kullanarak daha kişiselleştirilmiş ve inandırıcı olması beklenmektedir. Bu teknolojiler, saldırganların hedeflerini daha etkili bir şekilde manipüle etmelerine ve geleneksel savunma mekanizmalarını atlatmalarına olanak tanıyabilir.

13.6.1 Yapay Zeka Destekli Sosyal Mühendislik Saldırıları

Yapay zeka (AI), geleneksel sosyal mühendislik taktiklerini çok daha etkili ve ölçeklenebilir hale getirmektedir. Generatif AI, dilbilgisi hatası olmayan, son derece kişiselleştirilmiş ve gerçekçi e-postalar, SMS mesajları ve telefon görüşmeleri oluşturabilir. Bu, geleneksel filtreleri ve insan gözünü kolayca atlatabilir.

- **Deepfake ve Ses Klonlama:** Saldırganlar, bir kişinin sesini klonlayarak veya videosunu taklit ederek bir yöneticiyi veya meslektaşı ikna edici bir şekilde taklit edebilir.
- **AI Destekli Sohbet Botları:** Gerçek zamanlı olarak insan etkileşimini taklit eden otomatik sohbet botları, aynı anda binlerce kişiyle etkileşim kurarak hassas bilgiler elde edebilir.

Yapay zeka, sosyal mühendisliğin doğasını ve ölçeğini kökten değiştirmektedir. Daha önce beceri ve zaman gerektiren bir saldırı (örneğin, bir CEO'nun yazma stilini taklit etmek), artık birkaç saniye içinde bir AI tarafından yapılabilmektedir. Bu, savunmanın artık "bariz hataları" (imla hataları) arayamayacağını, bunun yerine kimlik doğrulamaya ve çok faktörlü doğrulamaya (MFA) daha fazla odaklanması gerektiğini gösterir. Bu, savunmanın doğrulama (verify) ve güvenmeme (zero trust) prensiplerine geçişini hızlandırır.

13.6.2 Sentetik Kimlik Oluşturma ve Manipülasyonu

Sentetik kimlik, gerçek ve uydurulmuş bilgilerin birleştirilmesiyle oluşturulan tamamen sahte bir kimliktir. Bu kimlikler, genellikle ölen veya çocukların sosyal güvenlik numaraları gibi gerçek unsurları, sahte isim ve adreslerle birleştirir. Geleneksel kimlik hırsızlığından farklı olarak, sentetik kimlikler kredi notu oluşturmak ve finansal dolandırıcılık yapmak için aylar veya yıllar boyunca kullanılabilir.

13.6.3 Etki Operasyonları ve Dezenformasyon Kampanyaları

Etki operasyonları, siber uzayda kamuoyunu manipüle etmeyi ve karar verme süreçlerini etkilemeyi amaçlayan kasıtlı faaliyetlerdir. Bu operasyonlar, yanlış veya yanıltıcı bilgileri (dezenformasyon) yayarak hedefin inançlarını ve davranışlarını değiştirmeyi hedefler. Dezenformasyon kampanyaları, sahte profiller, otomatik botlar ve yapay zeka tarafından oluşturulan inandırıcı içeriklerle yürütülür ve kamuoyu, seçimler veya şirket itibarları üzerinde yıkıcı etkilere sahip olabilir.

13.6.4 Hibrit Savaş ve Ulus-Devlet Sosyal Mühendisliği

Hibrit savaş, siber saldırıları, dezenformasyonu, ekonomik baskıyı ve fiziksel operasyonları bir araya getiren bir savaş biçimidir. Sosyal mühendislik, bu savaşın kritik bir bileşenidir, çünkü hedefin insan katmanını, yani "yumuşak hedefi" istismar eder. Ulus-devletler, entelektüel mülkiyeti, ticari sırları veya gizli hükümet materyallerini sızdırmak için siber casusluk operasyonlarında sosyal mühendisliği bir araç olarak kullanırlar.

13.6.5 Karşı İstihbarat ve Savunmacı Sosyal Mühendislik

Gelecekteki tehditlere karşı en güçlü savunma, pasif bir duruştan (saldırının gelmesini bekle) aktif bir duruşa (düşmanın ne yapacağını tahmin et ve önle al) geçmeyi gerektirir. Bu yaklaşım, sadece saldırıları engellemeye çalışmak yerine, saldırganın taktiklerini ve düşünce süreçlerini anlamayı amaçlayan ”karşı-istihbarat” ve ”savunmacı sosyal mühendislik” kavramlarını içerir. Bu, güvenlik uzmanlarının sadece teknoloji bilmesi değil, aynı zamanda bir saldırgan gibi düşünmesi gerektiğini gösterir.

Bölüm 14

GÜVENLİK OPERASYONLARI VE SOC/NOC YÖNETİMİ

Giriş

Güvenlik operasyonları ve SOC/NOC yönetimi, organizasyonların siber tehditlere karşı sürekli izleme ve müdahale yeteneklerini sağlayan kritik operasyonel disiplinlerdir. Bu bölümde SOC/NOC mimarilerini, güvenlik operasyonları yönetimini ve SIEM platformlarını detaylı olarak inceleyeceğiz.

14.1 Güvenlik Operasyonları Merkezi (SOC) Mimarisi

Bir SOC'nin temel mimari bileşenlerini, operasyonel modellerini ve iç işleyişini derinlemesine inceleyen bu bölüm, siber savunma yeteneklerinin omurgasını oluşturan yapıları detaylandırmaktadır.

14.1.1 SOC Organizasyonel Modelleri: Kurum İçi, Dış Kaynak ve Hibrit

Her kuruluş, kendine özgü kaynak, risk ve kontrol gereksinimlerine göre en uygun SOC modelini seçmelidir. Bu karar, yalnızca teknik bir tercih değil, aynı zamanda stratejik bir iş kararıdır.

Kurum İçi (In-house) SOC

Kurum içi SOC, bir kuruluşun kendi personeli ve altyapısı ile yönettiği modeldir. Bu yaklaşımın temel gücü, güvenlik süreçlerinin doğrudan kontrol edilebilir ve kuruluşun özel ihtiyaçlarına göre özelleştirilebilir olmasıdır. Dahili ekipler, altyapıya, verilere ve iş hedeflerine dair derinlemesine bir bilgi birikimi geliştirirler, bu da daha hassas tehdit algılama ve daha hızlı yanıt süreleri sağlar. Hassas verilerin şirket içinde kalması, özellikle yasal düzenlemelerle uyum açısından kritik bir avantajdır.

Ancak bu model, yüksek başlangıç ve sürekli işletme maliyetleri gerektirir. Uzman personel işe alma, ileri düzey güvenlik araçlarına yatırım yapma ve sürekli eğitim masrafları önemli bir finansal yük oluşturur. Sınırlı personel sayısı, özellikle 7/24 izleme gereken durumlarda analistlerde tükenmişlik (burnout) riskini artırabilir. Ek olarak, kurum içi ekiplerin bilgi birikimi kendi ortamlarıyla sınırlı kalabilir, bu da daha geniş tehdit ortamına dair kör noktalar oluşturabilir.

Dış Kaynak (Outsourced) SOC

Dış kaynak SOC hizmeti, üçüncü taraf bir sağlayıcı (MSSP) tarafından sunulur. Bu model, özellikle maliyetleri düşürme ve uzmanlığa hızlı erişim sağlama açısından caziptir. Dış kaynak SOC'ler, genellikle 7/24 kesintisiz izleme yeteneğine sahiptir ve farklı sektörlerden topladıkları geniş tehdit istihbaratı veritabanından faydalanırlar. Bu, kuruluşun kendi başına erişemeyeceği geniş bir yetenek ve bilgi havuzuna erişimini sağlar.

Bu modelin başlıca dezavantajları, kontrol ve esneklik kaybıdır. Hizmet kalitesi sağlayıcıdan sağlayıcıya büyük farklılık gösterebilir ve hizmetin kapsamı sözleşme koşullarıyla sınırlıdır. Dış kaynak ekipler, bir kuruluşun iş kültürünü veya özel altyapısını tam olarak anlayamayabilir, bu da hizmette boşluklara yol açabilir. Uzun vadede, belirli bir sağlayıcının araçlarına ve süreçlerine bağımlılık, başka bir modele geçişi maliyetli ve karmaşık hale getirebilir.

Hibrit SOC

Hibrit model, kurum içi ve dış kaynak yaklaşımlarının bir kombinasyonudur. Bu modelde, kuruluş kritik ve hassas operasyonları içeride tutarken, rutin izleme gibi daha az kritik görevleri dışarıdan sağlar. Bu yaklaşım, bütçeyi dengelemeye, ölçeklenebilirliği artırmaya ve yetenekleri en iyi şekilde kullanmaya olanak tanır.

Bu modelin en büyük zorluğu, yönetim karmaşıklığıdır. Farklı ekipler ve sözleşmeler arasında koordinasyon sağlamak, özel yönetim becerileri gerektirir. Kurum içi ve dış kaynak ekipler arasında etkili bir iletişim kurmak zor olabilir ve birden fazla sağlayıcıyı yönetmek, başlangıçta öngörülemeyen gizli maliyetler yaratabilir. Verilerin birden fazla yerde saklanması, yönetimi ve güvenliği daha karmaşık hale getirir. Her modelin kendine özgü güçlü ve zayıf yönleri, SOC model seçimini yalnızca bir teknoloji kararı olmaktan çıkarıp, risk, kontrol ve finansal çeviklik arasında bir denge kurmayı gerektiren, sürekli bir optimizasyon süreci haline getirmektedir.

Kriter	**Kurum İçi (In-house)**	**Dış Kaynak (Outsourced)**	**Hibrit**
Tanım	Kendi personeli ve altyapısıyla yönetilir.	Üçüncü taraf bir sağlayıcı (MSSP) tarafından sunulur.	Kritik fonksiyonlar kurum içinde tutulur, rutin görevler dış kaynak olarak kullanılır.
Maliyet	Yüksek başlangıç ve işletme giderleri.	Genellikle daha düşük maliyetli, abonelik tabanlı.	Dengeleyici, ancak gizli yönetim maliyetleri olabilir.
Kontrol Düzeyi	Maksimum kontrol ve kişiselleştirme.	Kısıtlı kontrol, sözleşme şartlarına bağlı.	Kritik fonksiyonlarda yüksek, diğerlerinde kısıtlı kontrol.
Uzmanlık Erişimi	Sınırlı, kendi personel yetkinliğine bağlı.	Geniş yetenek ve tehdit istihbaratı havuzuna erişim.	Her iki dünyanın avantajı: Hem iç yetenek hem dış uzmanlık.
Ölçeklenebilirlik	Sınırlı ve maliyetlidir.	Kolay ve hızlı ölçeklenebilir.	Kaynakları hem içeriden hem dışarıdan ekleyerek kolay ölçeklenir.
İletişim Zorluğu	Kolay, aynı çatı altında.	Zorlu, farklı kuruluşlar ve süreçler arası.	Karmaşık, iki ekip arasında sürekli koordinasyon gerektirir.

14.1.2 SOC Teknoloji Yığını ve Araç Entegrasyonu

Bir SOC'nin operasyonel yeteneklerini destekleyen teknoloji yığını, basit bir araç koleksiyonundan ziyade, her bir bileşenin birbiriyle entegre olması gereken karmaşık bir ekosistemdir. Bu entegrasyon, bir SOC'nin "kör noktaları ve boşlukları gidermek" için hayati önem taşır.

Temel Araçlar:

- **SIEM (Security Information and Event Management):** Bir SOC'nin temel taşıdır. Tüm log ve olay verilerini kuruluşun çeşitli kaynaklarından (sunucular, ağ cihazları, uygulamalar, güvenlik araçları) toplar, normalleştirir ve korelasyon analizi yaparak potansiyel güvenlik olaylarını tespit eder.
- **SOAR (Security Orchestration, Automation and Response):** SIEM'den gelen uyarıları ele alarak güvenlik araçlarını entegre eder ve otomatik iş akışları (playbook'lar) oluşturur. Bu, tekrarlayan görevleri otomatikleştirir ve olay yanıtını hızlandırır.
- **EDR/XDR (Endpoint Detection and Response/Extended Detection and Response):** Uç noktalarda (bilgisayarlar, sunucular) tehditleri algılar, analiz eder ve yanıt verir. EDR/XDR sistemlerinden gelen telemetri verileri, SIEM'e gönderilerek olayların daha geniş bir bağlamda analiz edilmesini sağlar.
- **TIP (Threat Intelligence Platform):** Saldırganların taktikleri, teknikleri, araçları ve göstergeleri hakkında dış tehdit istihbaratı beslemelerini yönetir ve SIEM ve SOAR gibi araçlara entegre ederek uyarıların zenginleştirilmesini sağlar.

Bu araçlar arasındaki kusursuz veri akışı, bir SOC'nin etkinliğini belirleyen ana faktördür. Bir SIEM, tek başına bir veri depolama aracı olarak kalırken, diğer güvenlik çözümleriyle entegre olduğunda, her birinin değerini katlayan güçlü bir istihbarat motoruna dönüşür. Tek bir çözümün yetersiz olduğu göz önüne alındığında, bir aracın değeri, sadece kendi yeteneklerinden değil, aynı zamanda diğer araçlarla ne kadar iyi işbirliği yapabildiğinden kaynaklanmaktadır. Bu durum, teknoloji yığınını yönetmeyi basit bir satın alma işleminden daha karmaşık bir ekosistem mühendisliği problemine dönüştürmektedir.

14.1.3 SOC Rolleri ve Sorumlulukları (L1, L2, L3 Analistleri)

Bir SOC, operasyonel verimliliği ve insan sermayesi yönetimini optimize etmek için katmanlı bir yapıda çalışır. Bu katmanlı model, iş akışlarını düzenler ve uzmanlık alanlarının en verimli şekilde kullanılmasını sağlar.

L1 (Tier 1) Analisti

L1 analistleri, bir SOC'nin ilk müdahale ekibidir. Başlıca sorumlulukları, gelen güvenlik uyarılarını önceliklendirmek (triage), hatalı pozitifleri (false positives) filtrelemek ve standart prosedürlerle çözülebilecek basit olayları ele almaktır. Bu filtreleme süreci, L2 ve L3 analistlerini, sık sık görülen uyarı hacminin yarattığı "alarm yorgunluğu"ndan korur. L1 analistleri, bir olay kendi kapsamlarını aştığında, daha derinlemesine bir araştırma için olayı ve toplanan tüm bilgileri L2 analistlerine iletirler.

L2 (Tier 2) Analisti

L2 analistleri, L1'den eskalasyonla gelen olayları derinlemesine incelemekten sorumludur. Bu görevler, kapsamlı log analizi, adli bilişim (forensics) incelemeleri ve tehditlerin nasıl sınırlandırılıp onarılacağına dair detaylı stratejiler geliştirmeyi içerir. Ayrıca, yeni tehditleri tespit etmek için özel korelasyon kuralları ve tespit mantığı oluşturma yeteneğine de sahiptirler.

L3 (Tier 3) Analisti

L3 analistleri, en yüksek uzmanlık seviyesini temsil ederler. En karmaşık güvenlik olaylarıyla (malware tersine mühendisliği, ağ adli bilişimi) ilgilenirler. Olay müdahalesinin ötesinde, tehdit avcılığı (threat hunting) metodolojileri oluşturur, şirket çapında güvenlik stratejileri tasarlar ve L1/L2 analistlerine mentorluk yaparak organizasyonel bilginin kurum içinde kalmasını sağlarlar. Bu katmanlama, SOC'nin sadece var olan tehditlere reaktif olarak yanıt vermekle kalmayıp, aynı zamanda geleceğe yönelik olarak savunma yeteneklerini proaktif bir şekilde güçlendirmesini sağlayan stratejik bir yapıdır.

Rol	Ana Sorumluluklar	Gerekli Beceriler	Ana Araçlar

L1 Analisti	Uyarı triyajı, hatalı pozitif filtrelemesi, olay eskalasyonu, izleme araçlarının ayarlanması.	Olay yönetim bilgisi, temel log analizi, güvenlik araçları bilgisi.	SIEM, EDR, Ticketing System.
L2 Analisti	Derinlemesine olay analizi, adli bilişim, içerik oluşturma (korelasyon kuralı geliştirme).	İleri seviye log analizi, adli bilişim, tehdit istihbaratı entegrasyonu.	SIEM, SOAR, EDR/XDR, TIP.
L3 Analisti	Uzman olay müdahalesi, malware analizi, tehdit avcılığı, güvenlik stratejisi geliştirme, mentorluk.	Tersine mühendislik, ağ ve sistem adli bilişimi, güvenlik mimarisi tasarımı.	SIEM, SOAR, EDR/XDR, TIP, Özel araçlar.

14.1.4 7/24 Operasyon Yönetimi ve Vardiya Planlaması

Kesintisiz güvenlik izlemesi, modern bir SOC için hayati önem taşır. Ancak, 7/24 operasyonun yönetimi, analistlerde hızlı tükenmeye yol açabilecek en büyük zorluklardan biridir. Yüksek personel devir hızı, ekip performansını ve kurumsal bilgiyi olumsuz etkiler.

Bu zorluğun üstesinden gelmek için akıllı vardiya planlaması ve otomasyon stratejileri benimsenmelidir. Yaygın vardiya modelleri arasında 8 veya 12 saatlik vardiyalar bulunur. Popüler 12 saatlik modellerden biri, analistlere dört gün çalışma ve ardından dört gün tatil imkanı sunan ”4-on, 4-off” modelidir. DuPont ve Pitman gibi daha karmaşık rotasyon programları, personel yorgunluğunu azaltmak ve sürekli uyanıklığı sağlamak için tasarlanmıştır.

Vardiya planlaması sadece bir çizelge sorunu değildir; aynı zamanda insan sermayesi yönetimi sorunudur. İş yükünün vardiyalar arasında adil bir şekilde dağıtılması ve tekrarlayan görevlerin otomasyona devredilmesi, tükenmişliği önlemenin anahtarıdır. Otomasyon, özellikle basit log analizleri ve triyaj gibi görevleri üstlenerek analistlerin iş yükünü hafifletir ve daha karmaşık, stratejik görevlere odaklanmalarını sağlar. Vardiya değişimlerinde, kritik bilginin ve bağlamın bir ekipten diğerine kesintisiz bir şekilde aktarılması için kapsamlı devir prosedürlerinin oluşturulması gereklidir. Bu süreçler, olay yanıtında kritik bilgi kaybını önler ve tutarlı bir performans sağlar.

14.1.5 SOC Olgunluk Modelleri ve Yetenek Değerlendirmesi

Bir SOC'nin ne kadar etkili çalıştığını değerlendirmek için, operasyonel olgunluğunu ölçen modeller kullanılır. SOC-CMM (SOC Capability Maturity Model) gibi modeller, bir SOC'nin mevcut durumunu değerlendirmek ve onu reaktif bir ”itfaiyeci” rolünden proaktif, stratejik bir güvenlik fonksiyonuna dönüştürmek için bir yol haritası sunar.

Bu yetenek değerlendirme süreçleri, mevcut güvenlik kontrollerindeki boşlukların belirlenmesiyle başlar. L3 analistleri gibi üst düzey uzmanlar, mevcut güvenlik duruşunu analiz eder, tehdit avcılığı metodolojileri geliştirir ve operasyonel süreçlerin nasıl iyileştirileceğine dair stratejik tavsiyelerde bulunur. Bu değerlendirmeler, organizasyonların sadece anlık tehditlere yanıt vermekle kalmayıp, aynı zamanda sürekli öğrenme ve iyileştirme döngüsünü kurumsallaştırmasını sağlar. Bir olgunluk modeli, soyut iyileştirme hedeflerini somut, ölçülebilir ve uygulanabilir bir sürece dönüştüren yapısal bir çerçevedir. Bu, SOC'nin sürekli gelişen tehdit ortamına uyum sağlaması ve savunma yeteneklerini sürekli olarak güçlendirmesi için hayati bir mekanizmadır.

14.2 Ağ Operasyon Merkezi (NOC) ve SOC Entegrasyonu

Bu bölüm, NOC ve SOC arasındaki temel ayrımı, işbirliği mekanizmalarını ve iki fonksiyonun birleşimi olan Birleşik Operasyon Merkezi (UOC) modelini incelemektedir. Bu entegrasyon, operasyonel verimliliği artırırken güvenlik duruşunu güçlendirmek için kritik öneme sahiptir.

14.2.1 NOC İşlevselliği ve Ağ İzleme Yetenekleri

Bir Ağ Operasyon Merkezi (NOC), ağ sistemlerini 7/24 izleyen ve yöneten merkezi bir birimdir. NOC, bir kuruluşun tüm ağ altyapısının kalbi olarak görev yapar ve modern iş operasyonlarının hayati teknolojik omurgasını oluşturur.

Temel NOC İşlevleri:

- **Ağ Performans İzleme:** Bant genişliği kullanımı, gecikme (latency), paket kaybı ve jitter gibi kritik ağ metriklerinin sürekli izlenmesi
- **Altyapı Sağlığı Yönetimi:** Router, switch, firewall ve sunucuların çalışma durumunun real-time takibi
- **Kapasite Planlama:** Ağ trafiği trendlerini analiz ederek gelecekteki kapasite ihtiyaçlarını öngörme
- **Konfigürasyon Yönetimi:** Ağ cihazlarının ayar değişikliklerini takip etme ve versiyon kontrolü
- **Sorun Giderme ve Root Cause Analysis:** Ağ kesintilerinin kök nedenlerini belirleme ve çözme
- **Yama ve Güncelleme Yönetimi:** Sistem güncellemelerinin planlanması ve uygulanması
- **Backup ve Felaket Kurtarma:** Veri yedekleme operasyonları ve iş sürekliliği planlarının yürütülmesi

İleri Seviye NOC Yetenekleri:

NOC, basit ağ izlemesinin ötesinde karmaşık ağ optimizasyon ve otomasyon yetenekleri sunar. **Intent-Based Networking (IBN)** gibi modern teknolojiler, NOC'nin ağ politikalarını otomatik olarak uygulamasına ve ağın kendini yapılandırmasına olanak tanır. **Network Digital Twins** kavramı ile NOC, fiziksel ağın sanal bir modelini oluşturarak değişikliklerin etkilerini test edebilir. Ayrıca **Predictive Analytics** kullanarak potansiyel ağ sorunlarını önceden tahmin eder ve proaktif bakım planları geliştirir.

NOC'nin asıl amacı, iş sürekliliğini ve müşteri memnuniyetini sağlamaktır. Sağlıklı bir ağ altyapısı, güvenlik operasyonlarının temelini oluşturur. Ancak, ağdaki bir anormallik (örneğin, alışılmadık trafik hacmi veya bir sunucu hatası) operasyonel bir sorundan (NOC'nin alanı) veya siber bir tehditten (SOC'nin alanı) kaynaklanabilir. Bu belirsizlik, iki merkezin işbirliğinin neden vazgeçilmez olduğunu ortaya koymaktadır.

14.2.2 Failover ve Yedeklilik Mekanizmaları

NOC operasyonlarının kritik başarı faktörlerinden biri, sistem arızalarında hızlı failover ve yedeklilik (redundancy) sağlama yeteneğidir. Modern iş dünyasında kesintinin maliyeti çok yüksek olduğundan, NOC'nin çoklu katmanlarda yedeklilik stratejileri geliştirmesi gerekir.

Failover Türleri ve Uygulamaları:

- **Aktif-Pasif Failover:** Birincil sistem çalışırken yedek sistem bekleme modunda bulunur. Arıza durumunda yedek sistem devreye girer.
- **Aktif-Aktif Failover:** Her iki sistem de aynı anda çalışır ve yük paylaşımı yapar. Bir sistemin arızalanması durumunda diğeri tüm yükü üstlenir.
- **Load Balancing ile Failover:** Trafiği birden fazla sunucu arasında dağıtan ve arızalı sunucuları otomatik olarak devre dışı bırakan mekanizma.

Kritik Altyapı Bileşenleri için Yedeklilik: - **Güç Kaynakları:** UPS sistemleri ve jeneratörler ile elektrik kesintilerine karşı koruma - **Ağ Bağlantıları:** Birden fazla ISP ve farklı rotalar üzerinden internet erişimi - **Veri Merkezleri:** Coğrafi olarak dağıtılmış veri merkezleri ile felaket kurtarma - **Kritik Cihazlar:** Router, switch ve firewall'ların yedek örneklerinin hazır bulunması

Failover Test Prosedürleri: NOC, düzenli olarak failover testleri gerçekleştirmeli ve Recovery Time Objective (RTO) ile Recovery Point Objective (RPO) hedeflerini doğrulamalıdır. Bu testler, gerçek bir felaket anında sistemlerin beklendiği gibi çalışacağını garanti eder.

14.2.3 Network Monitoring ve Performance Management

Etkili NOC operasyonları, ağ performansının proaktif izlenmesi ve yönetilmesi üzerine kuruludur. Modern ağ izleme sistemleri, basit ping testlerinin ötesine geçerek derinlemesine analitik yetenekler sunar.

Kritik İzleme Metrikleri:

- **Throughput (İş Hacmi):** Belirli bir zaman diliminde ağdan geçen veri miktarı
- **Latency (Gecikme):** Veri paketinin kaynak ile hedef arasında seyahat süresi
- **Packet Loss (Paket Kaybı):** İletim sırasında kaybolan veri paketlerinin yüzdesi
- **Jitter:** Gecikme sürelerindeki değişkenlik, özellikle real-time uygulamalar için kritik
- **Bandwidth Utilization:** Kullanılabilir bant genişliğinin ne kadarının aktif olarak kullanıldığı
- **Error Rate:** Ağ cihazlarında meydana gelen hata oranları

İleri Seviye Monitoring Teknolojileri: - **Deep Packet Inspection (DPI):** Ağ trafiğinin içeriğini analiz ederek uygulama bazında performans izleme - **Network Flow Analysis:** NetFlow, sFlow ve IPFIX protokolleri ile trafik akış analizi - **Synthetic Monitoring:** Yapay test trafiği oluşturarak ağ performansını sürekli test etme - **Real User Monitoring (RUM):** Gerçek kullanıcı deneyimini izleyerek performans metriklerini toplama

14.2.4 Capacity Planning ve Ölçeklenebilirlik

NOC'nin stratejik sorumluluklarından biri, gelecekteki kapasite ihtiyaçlarını öngörmek ve ağın büyümeyi destekleyecek şekilde ölçeklenmesini sağlamaktır.

Kapasite Planlama Metodolojisi:

1. **Baseline Belirleme:** Mevcut ağ kullanımının detaylı analizi ve normal operasyon parametrelerinin belirlenmesi
2. **Trend Analizi:** Geçmiş verileri kullanarak büyüme trendlerinin matematiksel modellemesi
3. **İş Büyüme Projeksiyonları:** İş birimlerinin büyüme planları ile ağ kapasitesi gereksinimlerinin ilişkilendirilmesi
4. **Peak Load Analizi:** En yoğun kullanım dönemlerinin analizi ve aşırı yük senaryolarının planlanması
5. **Teknoloji Roadmap:** Yeni teknolojilerin (5G, IoT, Cloud) ağ kapasitesi üzerindeki etkilerinin değerlendirilmesi

Ölçeklenebilirlik Stratejileri: - **Horizontal Scaling:** Daha fazla cihaz ekleyerek kapasiteyi artırma - **Vertical Scaling:** Mevcut cihazların performansını artırma - **Software-Defined Networking (SDN):** Yazılım tabanlı ağ yönetimi ile dinamik ölçeklendirme - **Network Function Virtualization (NFV):** Ağ fonksiyonlarının sanallaştırılması ile maliyet etkin ölçekleme

14.2.5 SOC-NOC İşbirliği ve Bilgi Paylaşımı

Geleneksel olarak, NOC ve SOC ekipleri, farklı hedeflere sahip ayrı silolar olarak çalışır. Ancak, her iki ekibin de nihai amacı hizmetlerin sürekli kullanılabilirliğini sağlamaktır. Bu nedenle, bu siloları kırmak, operasyonel verimsizliği ve güvenlik risklerini azaltmak için kritik öneme sahiptir.

Entegre İşbirliği Modelleri:

- **Cross-Functional Teams:** NOC ve SOC personelinin ortak projeler ve olay müdahalelerinde birlikte çalışması
- **Shared Dashboards:** Her iki ekibin erişebileceği ortak izleme panelleri ve alarm sistemleri
- **Joint Incident Response:** Güvenlik ve operasyonel olayların birleşik müdahale prosedürleri ile ele alınması
- **Knowledge Sharing:** Düzenli bilgi paylaşımı toplantıları ve çapraz eğitim programları

İşbirliği, ortak veri kaynakları, entegre araçlar ve standartlaştırılmış süreçler aracılığıyla gerçekleştirilir. Örneğin, bir NOC tarafından tespit edilen anormal trafik desenleri, bir siber tehdide işaret edebileceğinden derhal SOC ekibine aktarılmalıdır. Bu işbirliği, operasyonel ve güvenlik sorunlarının hızla ilişkilendirilmesini sağlar. Veri çoğaltmasını en aza indirerek ve bilgi paylaşımını artırarak, bu entegrasyonlar olay yanıt sürelerini hızlandırır ve maliyet etkinliğini artırır. NOC'nin gördüğü bir ağ performansı sorunu, bir SOC'nin araştırdığı veri sızdırma girişimiyle ilişkili olabilir.

14.2.6 Change Management ve Konfigürasyon Kontrolü

NOC operasyonlarının kritik bir bileşeni, ağ altyapısındaki değişikliklerin kontrollü ve güvenli bir şekilde yönetilmesidir. Yanlış yapılandırılan bir değişiklik, tüm ağı çökertebilir ve işletmeye milyonlarca dolar zarara neden olabilir.

Change Management Süreç Adımları:

1. **Change Request (RFC):** Değişiklik talebinin resmi olarak dokumentasyonu ve gerekçelendirilmesi
2. **Impact Assessment:** Değişikliğin sistem üzerindeki potansiyel etkilerinin analizi
3. **Change Advisory Board (CAB):** Teknik ve iş temsilcilerinin katıldığı değişiklik onay komitesi
4. **Testing:** Lab ortamında değişikliğin test edilmesi ve validation
5. **Implementation:** Üretim ortamına kontrollü değişiklik uygulaması
6. **Post-Implementation Review:** Değişikliğin başarısının değerlendirilmesi ve rollback planı

Konfigürasyon Yönetimi Best Practice'leri: - **Version Control:** Tüm network cihaz konfigürasyonlarının versiyonlanması ve merkezi depoda saklanması - **Automated Backup:** Konfigürasyon değişikliklerinden önce otomatik yedekleme - **Configuration Drift Detection:** Beklenmedik konfigürasyon değişikliklerinin otomatik tespiti - **Standard Templates:** Standart konfigürasyon şablonları ile tutarlılık sağlama - **Emergency Change Procedures:** Acil durumlar için hızlandırılmış değişiklik prosedürleri

Change Management Araçları: Modern NOC'ler, change management süreçlerini destekleyen ITIL tabanlı araçlar kullanır. ServiceNow, Remedy ve Jira Service Management gibi platformlar, değişiklik yaşam döngüsünü otomatikleştirir ve audit trail sağlar.

14.2.7 NOC Automation ve Orchestration

Modern NOC operasyonları, manuel süreçlerin otomasyonuyla verimliliği artırır ve insan hatalarını azaltır. Otomasyon, rutin görevlerden karmaşık iş akışlarına kadar geniş bir spektrumda uygulanır.

Temel Otomasyon Alanları:

- **Network Discovery:** Yeni cihazların otomatik keşfi ve envantere eklenmesi
- **Configuration Management:** Konfigürasyonların otomatik dağıtımı ve güncellenmesi
- **Patch Management:** Güvenlik yamalarının otomatik test edilmesi ve uygulanması
- **Backup Automation:** Düzenli veri yedekleme işlemlerinin otomatikleştirilmesi
- **Health Checks:** Sistem sağlık kontrollerinin otomatik gerçekleştirilmesi
- **Incident Response:** Belirli alarm türleri için otomatik müdahale prosedürleri

Network Orchestration Teknolojileri: - **Ansible Network Automation:** Ağ cihazlarının yapılandırması için playbook tabanlı otomasyon - **Terraform Network Modules:** Infrastructure as Code yaklaşımıyla ağ altyapısı yönetimi - **Python Network Scripts:** Özel otomasyon görevleri için Python tabanlı scriptler - **NETCONF/YANG:** Standartlaştırılmış ağ yönetimi protokolleri - **REST API Integration:** Ağ cihazlarının API'leri üzerinden otomatik yönetimi

Self-Healing Networks: Gelişmiş NOC otomasyon sistemleri, belirli sorunları otomatik olarak tespit edip düzeltebilen "self-healing" yetenekleri geliştirir. Bu sistemler, alarm aldıktan sonra otomatik olarak tanılama yapabilir, geçici çözümler uygulayabilir ve gerekirse failover işlemlerini başlatabilir.

14.2.8 Birleşik Operasyon Merkezi (UOC) Modelleri

UOC, NOC ve SOC işlevlerini birleştirerek, tüm iş operasyonları için tek bir merkezi komuta ve kontrol noktası oluşturan bir yaklaşımdır. Bu model, mühendislik, operasyon, güvenlik, performans ve finansal verileri tek bir "cam bölme" (single pane of glass) görünümünde birleştirir.

UOC'nin Temel Bileşenleri:

- **Unified Dashboard:** NOC, SOC ve diğer operasyonel verilerin tek ekranda görüntülenmesi
- **Cross-Functional Analytics:** Güvenlik, performans ve operasyonel metriklerin korele edilmesi
- **Integrated Incident Management:** Tüm olay türleri için birleşik müdahale süreçleri
- **Shared Knowledge Base:** Ortak prosedürler ve çözüm kütüphanesi
- **Joint Training Programs:** Çapraz fonksiyonel eğitim ve sertifikasyon programları

Bu birleşik yaklaşım, veri ve iletişim silolarını kırar ve daha iyi işbirliği, daha hızlı karar alma ve artan operasyonel verimlilik sağlar. UOC, özellikle enerji, su yönetimi ve akıllı şehirler gibi endüstrilerde, operasyonel teknolojiler (OT) ile bilgi teknolojileri (IT) arasındaki verileri entegre ederek uçtan uca görünürlük sağlar. Bu, güvenlik ve operasyonel kararların, işletmenin daha geniş hedefleriyle uyumlu hale getirilmesine olanak tanıyan, fonksiyon merkezli bir modelden (NOC, SOC) iş çıktısı merkezli bir modele stratejik bir geçişi temsil eder.

14.2.9 Hizmet Seviyesi Anlaşmaları (SLA) ve Performans Metrikleri

Hizmet Seviyesi Anlaşmaları (SLA), bir hizmet sağlayıcı ile müşteri arasındaki hizmet kalitesini tanımlayan resmi sözleşmelerdir. Bu anlaşmalar, performansı somut ve ölçülebilir hale getirmek için çeşitli metrikleri içerir.

Kritik Metrikler:

- **Ortalama Tespit Süresi (MTTD - Mean Time to Detect):** Bir tehdidin oluştuğu an ile SOC tarafından tespit edildiği an arasında geçen ortalama süreyi ölçer. Düşük bir MTTD, saldırganların sistemde kalış süresinin azaldığı anlamına gelir, bu da potansiyel hasarı sınırlar.
- **Ortalama Yanıt Süresi (MTTR - Mean Time to Respond):** Bir uyarının tespit edilmesinden sonra, analistin olaya müdahale etmek için geçirdiği ortalama süreyi ölçer. Bu, genellikle olayın önceliğine bağlı olarak değişir.

- **Ortalama Çözüm Süresi (MTTR - Mean Time to Resolution):** Bir olayın başlangıcından itibaren tamamen çözülmesine ve normal operasyonlara dönülmesine kadar geçen toplam süreyi ölçer.
- **Hatalı Pozitif Oranı (FPR):** Bir güvenlik uyarısının yanlışlıkla bir tehdit olarak etiketlenme yüzdesini gösterir. Yüksek bir FPR, analist yorgunluğuna yol açar ve verimliliği düşürür.

Bu metrikler, SOC ve NOC'nin teknik performansını somutlaştırarak, siber güvenlik yatırımlarının iş üzerindeki etkisini ve getirisini kanıtlamaya olanak tanır. Düşük bir MTTR, doğrudan operasyonel riski ve olası finansal kayıpları azaltır. Bu nedenle, metrikler sadece bir raporlama aracı değil, aynı zamanda sürekli iyileştirme için temel bir yol göstericidir.

14.2.10 Eskalasyon Prosedürleri ve Olay Devir Süreçleri

Etkili bir olay yanıtı, her olayın önceliğine ve niteliğine göre doğru ekibe, doğru zamanda ve doğru bilgilerle aktarılmasını gerektirir. İyi tanımlanmış eskalasyon prosedürleri, olay yanıt sürecinde bir olayın "kaybolmasını" önleyen, yanıt süresini minimize eden ve doğru uzmanlığın doğru zamanda devreye girmesini sağlayan kritik bir operasyonel kontrol mekanizmasıdır.

Süreç, L1 analistleri tarafından yapılan olay triyajıyla başlar. Olay, aciliyetine ve karmaşıklığına göre önceliklendirilir. Eğer olay, L1'in yetki alanını aşarsa, önceden belirlenmiş protokollere göre L2'ye veya duruma göre diğer birimlere (örneğin, üst yönetim, hukuk veya insan kaynakları) eskalasyon yapılır. Bu süreçte, tüm bağlamsal bilgilerin (loglar, analiz notları, ilgili sistemler) eksiksiz bir şekilde aktarılması büyük önem taşır.

NOC ile SOC arasındaki devir süreçleri de bu prosedürlerin bir alt kümesidir. Örneğin, bir NOC, ağda bir anormallik tespit ettiğinde ve bunun potansiyel bir siber saldırı olduğunu belirlediğinde, olayı resmi bir devir süreciyle SOC'ye aktarır. Bu tür formalize edilmiş prosedürlerin olmaması, kaotik bir durumda bile koordineli bir yanıtı engelleyerek olay yanıtında kritik gecikmelere yol açabilir.

14.3 SIEM Platform Yönetimi ve Log Analizi

Bir SIEM platformunun yönetimi ve log analizi, modern bir SOC'nin temel yeteneklerini oluşturur. Bu bölüm, bir SIEM'in mimarisini, veri işleme süreçlerini ve tehdit tespiti için nasıl kullanıldığını ele almaktadır.

14.3.1 SIEM Mimari Tasarımı ve Ölçeklenebilirlik

SIEM mimarisi, bir SOC'nin gelecekteki büyüme ve veri hacmiyle başa çıkma yeteneğini doğrudan belirler. Temel mimari bileşenler arasında veri toplama, normalizasyon, korelasyon motoru, uyarı ve raporlama, log yönetimi ve saklama yer alır.

Modern bir SIEM'in en büyük zorluğu, yüksek hacimli, hızlı ve çeşitli güvenlik verilerini yönetmektir. Geleneksel (on-premise) SIEM mimarileri, bu tür büyük veri hacimlerini yönetmede ve maliyet etkin bir şekilde saklamada zorluklar yaşayabilir. Bu zorlukların üstesinden gelmek için, SIEM'ler dağıtık mimarilere (horizontal scaling) ve bulut tabanlı çözümlere yönelmektedir. Bulut tabanlı SIEM'ler, otomatik ölçeklendirme ve esneklik sunarak, bir kuruluşun finansal riskini sermaye harcamasından (CAPEX) operasyonel harcamaya (OPEX) dönüştürür. Dağıtık bir mimari, veri işlemeyi ve depolamayı birden fazla düğüme dağıtarak yüksek performansın sürdürülmesini sağlar.

14.3.2 Log Toplama, Normalizasyon ve Zenginleştirme

SIEM'in analitik yeteneklerinin temelini, işlenen verinin kalitesi oluşturur. Bu kalite, log toplama, normalizasyon ve zenginleştirme süreçleriyle sağlanır.

- **Log Toplama:** Bu ilk adımda, kuruluşun tüm altyapısından (sunucular, uygulamalar, ağ cihazları, güvenlik cihazları gibi) log verileri merkezi SIEM platformuna aktarılır.
- **Normalizasyon:** Farklı kaynaklardan gelen ham log verileri, farklı format ve şemalara sahiptir. Normalizasyon, bu dağınık veriyi SIEM'in kolayca analiz edebileceği tutarlı ve standart bir formata dönüştürür. Bu süreç, veri arama ve analizini hızlandırır, aynı zamanda depolama maliyetlerini optimize eder.
- **Zenginleştirme:** Normalleştirilmiş log verisine ek bağlamsal bilgi (kullanıcı kimliği, coğrafi konum, varlık bilgisi, tehdit istihbaratı) eklenmesidir. Örneğin, başarısız bir oturum açma girişiminin loguna, kaynağın bilinen kötü niyetli bir IP adresi olup olmadığı veya coğrafi konumunun anormal olup olmadığı bilgisi eklenir. Bu bağlamsal bilgi, analistlerin bir olayın gerçek bir tehdit olup olmadığını hızla belirlemesine yardımcı olur, böylece hatalı pozitifleri ve alarm yorgunluğunu azaltır.

14.3.3 Korelasyon Kuralı Geliştirme ve Ayarlama

Korelasyon kuralları, bir SIEM'in zekasını oluşturan mantık parçalarıdır. Tek başına zararsız görünen birden fazla olayı mantıksal olarak birleştirerek, karmaşık tehditleri (örneğin, brute-force saldırısı, ayrıcalık yükseltme) tespit etmek için kullanılır. Kurallar, olaylar arasında zaman pencereleri, belirli bir olay dizisi ve mantıksal koşullar (AND/OR) temel alınarak oluşturulur.

Kural geliştirme, bir organizasyonun kendine özgü "normal" trafiğini belirlemeyi ve eşik değerlerini buna göre sürekli olarak ayarlamayı gerektiren bir süreçtir. Eğer bir kural çok geniş tutulursa, normal aktiviteler yanlışlıkla uyarıları tetikleyebilir ve hatalı pozitiflere yol açabilir. Bu durum, "alarm yorgunluğunun" en yaygın kaynaklarından biridir. Bu nedenle, kural ayarlama (tuning), sürekli bir operasyonel görevdir. SIEM'in etkinliğini korumak için, ağda yeni cihazlar eklendikçe veya yazılım güncellemeleri yapıldıkça kuralların gözden geçirilmesi ve ayarlanması gerekir.

14.3.4 Use Case Geliştirme ve Tespit Mühendisliği

Tespit mühendisliği, reaktif bir "uyarı yanıtlayıcı" rolünden, stratejik bir "savunma oluşturucu" rolüne geçişin kurumsal mekanizmasıdır. Bu, tehdit istihbaratını (MITRE ATT&CK gibi) kullanarak, bir organizasyon için en ilgili tehditlere karşı proaktif olarak tespitler oluşturma sürecidir.

Bu sürecin yaşam döngüsü aşağıdaki adımları içerir:

1. **Tehdit Modelleme:** Kuruluşun tehdit profilini anlamak ve MITRE ATT&CK çerçevesini kullanarak hangi tehdit aktörlerinin ve TTP'lerin en alakalı olduğunu belirlemekle başlar.
2. **Veri Gereksinimleri:** Belirlenen tehditleri tespit etmek için hangi log ve telemetri verilerinin gerekli olduğunu saptamak.
3. **Mantık Geliştirme:** SIEM'de veya diğer araçlarda, tespit mantığını kural veya makine öğrenimi modeli olarak yazmak.
4. **Test ve Doğrulama:** Oluşturulan kuralın bir test ortamında, kontrollü saldırı senaryolarıyla (örneğin, kırmızı takım-mavi takım işbirliğiyle) çalıştığının doğrulanması.
5. **Dağıtım ve İyileştirme:** Kuralı canlı ortama almak ve performansını izleyerek sürekli olarak ayarlamaktır.

Bu döngü, bir SOC'nin rastgele uyarıları ele almaktan, iş hedefleri ve tehdit modeline dayalı olarak kasıtlı bir şekilde savunma yetenekleri oluşturmaya geçtiğini gösterir.

14.3.5 SIEM Performans Optimizasyonu ve Depolama Yönetimi

SIEM'in teknik performansı (sorgu hızı, veri alımı), doğrudan operasyonel performansı ve olay yanıt süresini etkiler. Bu, altyapı yönetiminin, SOC'nin başarısı için bir ön koşul olduğu anlamına gelir. Performans optimizasyonu, log normalizasyonu ve veri toplama süreçlerinin iyileştirilmesiyle başlar. Verilerin tutarlı bir formata

dönüştürülmesi ve gereksiz alanların kaldırılması, sorgu performansını artırır ve depolama gereksinimlerini azaltır.

Depolama yönetimi, SIEM'in yüksek veri hacmiyle başa çıkabilmesi için hayati önem taşır. Veri yaşam döngüsü yönetimi stratejileri, sıcak, soğuk veya arşiv depolama katmanlarını kullanarak verilerin maliyet etkin bir şekilde saklanmasını sağlar. Eğer bir analist, basit bir sorgunun çalışması için dakikalarca beklemek zorunda kalırsa, bu operasyonel verimliliği düşürür ve olay çözümünü geciktirir.

14.4 Güvenlik Orkestrasyonu, Otomasyonu ve Yanıtı (SOAR)

SOAR, bir SOC'nin verimliliğini, hızını ve tutarlılığını dönüştüren, güvenlik operasyonlarını merkezileştiren ve otomatikleştiren bir platformdur.

14.4.1 SOAR Platform Seçimi ve Uygulama

SOAR, güvenlik orkestrasyonu, otomasyon ve yanıt yeteneklerini birleştirir. En büyük değeri, analistlerin tekrarlayan, manuel görevlerden kurtulmasını sağlaması, olay yanıt sürelerini (MTTR) kısaltması ve playbook'lar aracılığıyla yanıt süreçlerini standartlaştırmasıdır. SOAR'ın getirdiği verimlilik, analistlerin daha karmaşık ve stratejik görevlere (örneğin, proaktif tehdit avcılığı) odaklanmasını sağlar.

SOAR platformu seçimi, bir kuruluşun mevcut güvenlik araçlarıyla (SIEM, EDR vb.) kusursuz bir şekilde entegre olabilme yeteneğine bağlıdır. Platformun API yönetimini ve diğer araçlarla entegrasyonu ne kadar iyi desteklediği, SOAR'ın potansiyelini doğrudan belirler. SOAR, SIEM ile başlayan "görünürlük" yolculuğunu, "eyleme geçirilebilirlik" aşamasına taşıyan bir köprüdür.

14.4.2 Playbook Geliştirme ve Otomatize İş Akışları

Playbook'lar, belirli bir tehdit türüne (örneğin, kimlik avı) karşı izlenecek adımları tanımlayan, otomatikleştirilmiş iş akışlarıdır. Geliştirme süreci, en sık ve tekrarlayan görevlerin belirlenmesiyle başlar. Playbook'lar, saldırının tipine göre veri zenginleştirme, zararlı göstergeleri engelleme ve etkilenen uç noktaları izole etme gibi adımları içerir.

Pratik Senaryo: Otomatik Kimlik Avı Yanıt Playbook'u

1. **Tetkik:** Bir çalışan, şüpheli bir e-postayı güvenlik ekibine bildirir veya bir güvenlik aracı kimlik avı girişimi tespit eder. Bu olay, playbook'u otomatik olarak tetikler.
2. **Veri Ayıklama:** Playbook, e-postadan göstergeleri (URL, IP, dosya karması) otomatik olarak ayıklar.
3. **Zenginleştirme:** Ayıklanan göstergeler, VirusTotal gibi üçüncü taraf tehdit istihbaratı araçlarıyla çapraz kontrol edilir ve e-postanın sahte olup olmadığı doğrulanır.
4. **Otomatik Engelleme:** Eğer URL veya IP adresi kötü niyetli olarak doğrulanırsa, playbook ilgili güvenlik duvarı veya DNS ayarlarında bu göstergeleri otomatik olarak engeller.
5. **Bildirim ve Vaka Oluşturma:** Playbook, olayı analiz etmesi için bir analist için otomatik olarak bir vaka yönetim sisteminde bir ticket oluşturur ve topladığı tüm bağlamsal bilgileri (veri zenginleştirmesi, otomatik eylemler) bu ticketa ekler.

Bu süreç, manuel olarak saatler sürebilecek bir görevi saniyeler içinde tamamlayarak analistlerin iş yükünü büyük ölçüde azaltır ve yanıt süresini kısaltır.

14.4.3 Güvenlik Aracı Entegrasyonu ve API Yönetimi

SOAR'ın gerçek değeri, diğer güvenlik araçlarıyla entegrasyon yeteneğine bağlıdır. SIEM, EDR, güvenlik duvarları ve IAM gibi araçların SOAR ile entegrasyonu, playbook'ların bu araçlar üzerinde eylemler gerçekleştirmesini sağlar. Bu entegrasyonların temelini API'lar oluşturur.

API'lar, güvenlik araçları arasındaki iletişim için kritik bir altyapıdır. API güvenliği (kimlik doğrulama, yetkilendirme) ve yaşam döngüsü yönetimi, bu entegrasyonun güvenilirliği için hayati önem taşır. Güvenlik operasyonlarında API yönetimi, yalnızca işlevselliği sağlamakla kalmaz, aynı zamanda API'ları siber saldırılara karşı koruyarak veri güvenliğini de sağlar.

14.4.4 Vaka Yönetimi ve Ticket Sistemi Entegrasyonu

SOAR, olayları otomatik olarak ele alırken, insan ekiplerle senkronize çalışabilmek için vaka yönetimi ve ticket sistemleri (örneğin, Jira, ServiceNow) ile entegre olur. Bu entegrasyon, otomatikleştirilmiş yanıt süreçlerinin ilerlemesini takip etmek ve analistlerin gözetimini sağlamak için kritik öneme sahiptir.

Playbook'lar, bir olay tetiklendiğinde otomatik olarak bir ticket oluşturabilir ve bu ticketa bağlamsal bilgileri (loglar, tehdit göstergeleri, otomatik eylemler) ekleyebilir. Bu, analistlerin manuel olarak veri toplama ve raporlama ihtiyacını ortadan kaldırır. Bu entegrasyon, otomatikleştirilmiş yanıt ile insan gözetimi ve raporlaması arasındaki döngüyü tamamlar, böylece olayların tek bir yerden takibini ve anahtar metriklerin (MTTR) kolayca ölçülmesini sağlar.

14.4.5 Otomasyonun ROI Hesaplanması ve Süreç Optimizasyonu

SOAR yatırımlarının geri dönüşü (ROI), basit bir maliyet tasarrufu hesaplamasının ötesindedir. Gerçek değeri, insan sermayesinin optimizasyonu ve reaktif bir iş gücünün proaktif bir ekibe dönüştürülmesi yeteneğidir.

SOAR'ın ROI'si, manuel ve otomatik süreçler arasında kaydedilen zamana dayalı olarak hesaplanabilir. Ancak, otomasyonun asıl değeri, manuel iş yükünün azalmasının yanı sıra, analistlerin daha karmaşık ve stratejik görevlere (örneğin, tehdit avcılığı) odaklanmasını sağlamasıdır. Otomasyonla boşalan zaman, operasyonel işlerden tehdit avcılığı ve güvenlik mimarisi geliştirmeleri gibi daha değerli işlere yönlendirilebilir. Bu, SOAR'ın yalnızca para tasarrufu sağlamadığını, aynı zamanda mevcut analistlerin değerini ve becerilerini artırdığını gösterir.

14.5 Tehdit Tespit Mühendisliği ve Tehdit Avcılığı Operasyonları

Bu bölüm, modern bir SOC'nin en proaktif iki fonksiyonunu, tespit mühendisliğini ve tehdit avcılığını derinlemesine ele almaktadır.

14.5.1 Tespit Kullanım Senaryosu Geliştirme Yaşam Döngüsü

Tehdit tespit mühendisliği, tehdit aktörlerinin taktiklerini, tekniklerini ve prosedürlerini (TTP'ler) anlamak ve bunlara karşı etkili savunmalar oluşturmak için disiplinli bir yaklaşımdır. Bu süreç, reaktif bir "uyarı yanıtı" kültüründen, aktif olarak "tehditleri tasarlayıp engelleme" kültürüne geçişi temsil eder.

Yaşam döngüsü şu adımları içerir:

1. **Tehdit Modelleme:** Kuruluş için en alakalı tehditlerin MITRE ATT&CK çerçevesi kullanılarak belirlenmesi.
2. **Veri Kaynağı Gereksinimleri:** Tespiti gerçekleştirmek için hangi log ve telemetri verilerinin gerekli olduğunun saptanması.

3. **Mantık Geliştirme:** SIEM'de veya diğer araçlarda, tespit mantığını kural veya makine öğrenimi modeli olarak yazmak.
4. **Test ve Doğrulama:** Kuralın bir test ortamında, kontrollü saldırı senaryolarıyla (örneğin, kırmızı takım-mavi takım işbirliğiyle) çalıştığının doğrulanması.
5. **Ayarlama ve Sürekli İyileştirme:** Hatalı pozitifleri azaltmak ve yeni tehditlere uyum sağlamak için kuralın sürekli ayarlanması.

Bu döngü, bir SOC'nin sürekli gelişen tehdit ortamına uyum sağlamasını ve reaktif yeteneklerini proaktif stratejilerle desteklemesini sağlar.

14.5.2 Davranışsal Analiz ve Anomali Tespit Kuralları

Davranışsal analiz, kullanıcı ve varlıkların "normal" davranışlarını öğrenen ve bu normal profillerden sapmaları (anomalileri) tespit ederek tehditleri ortaya çıkaran bir teknolojidir. Geleneksel kural tabanlı sistemler, yalnızca bilinen imzaları veya kalıpları arar. Davranışsal analiz ise, bilinen imzası olmayan veya henüz bilinmeyen tehditleri (sıfırıncı gün saldırıları, iç tehditler) tespit etme yeteneği sunar.

Çalışma mekanizması, istatistiksel analiz ve makine öğrenimi algoritmalarına dayanır. Örneğin, bir çalışanın normalde erişmediği dosyalara erişmesi veya alışılmadık saatlerde oturum açması gibi davranışsal anormallikler, sistem tarafından hemen işaretlenir. Ancak, bu teknolojinin en büyük zorluğu "normal"in tanımını yapmak ve hatalı pozitif riskini yönetmektir. Bu, sistemin sürekli olarak ince ayar yapılmasını ve evrilen tehditlere adapte olmasını gerektirir.

14.5.3 Tehdit Avcılığı Programı Uygulaması

Tehdit avcılığı, otomatik araçlar tarafından gözden kaçırılan veya bilinmeyen tehditleri proaktif olarak arayan, insan odaklı bir faaliyettir. Amacı, reaktif olay yanıtını tamamlayarak, kuruluşu daha siber dirençli hale getirmektir. Tehdit avcılığı, bir SOC'nin mevcut tespit yeteneklerini test eden ve doğrulayan bir kalite güvence (QA) süreci olarak da hizmet eder.

Bir tehdit avcılığı programı şu temel adımları içerir:

1. **Hipotez Geliştirme:** Bir araştırma sorusu veya hipoteziyle başlama.
2. **Veri Toplama:** Hipotezi test etmek için gerekli logları ve telemetri verilerini toplama.
3. **Soruşturma ve Analiz:** Toplanan veri üzerinde desen ve anormallik arama.
4. **Doğrulama ve Yanıt:** Tehdidin varlığını doğrulama ve olay yanıt ekibine eskalasyon.

14.5.4 Avlanma Hipotezi Geliştirme ve Doğrulama

Tehdit avcılığının en önemli adımı, araştırmayı yönlendiren, spesifik ve test edilebilir bir hipotez oluşturmaktır. Hipotez, tehdit istihbaratına, önceki olaylara veya SIEM/UEBA tarafından belirlenen anormal bir aktiviteye dayanabilir.

Bir hipotezi detaylandırmak için ABLE (Actor, Behavior, Location, Evidence - Aktör, Davranış, Konum, Kanıt) çerçevesi kullanılabilir. Örneğin, "Bir tehdit aktörü, DNS tünelleme kullanarak hassas finansal verileri sızdırıyor olabilir" gibi spesifik bir hipotez, tehdit avcısını doğru veri kaynaklarına ve belirli bir hedefe yönlendirerek boşa harcanan çabayı azaltır. Bu, sınırlı SOC kaynaklarının en etkili şekilde kullanılmasını sağlar.

14.5.5 Tespit Kapsamı Değerlendirmesi ve Boşluk Analizi

Tespit kapsama değerlendirmesi, bir SOC'nin mevcut güvenlik kontrollerinin, MITRE ATT&CK gibi bir çerçeveye göre ne kadar etkili olduğunu değerlendirme sürecidir. Bu süreç, log kaynağı yapılandırma hataları, bozuk log toplayıcılar veya yetersiz kural setleri nedeniyle oluşan tespit boşluklarını belirlemeye yardımcı olur.

Bu analizin amacı %100 kapsama ulaşmak değil, kuruluşun tehdit modeline göre en kritik tehditleri önceliklendirmektir. Bu süreç, soyut kapsama verilerini somut metriklere dönüştürür ve gelecekteki güvenlik yatırımlarını riske dayalı verilere göre gerekçelendirmek için kullanılır.

14.6 SOC Performans Yönetimi ve Sürekli İyileştirme

Bir SOC'nin performansı, operasyonel verimliliğini nasıl ölçtüğüne ve sürekli olarak nasıl artırdığına bağlıdır.

14.6.1 SOC Metrik Geliştirme: Verimlilik, Etkililik, Kalite

SOC metrikleri, operasyonel hedeflere ne kadar ulaşıldığını gösteren nicel ölçümlerdir.

Metrik	Tanım	Hesaplama Formülü	Hedef Değer Aralığı
MTTD	Bir tehdidin oluştuğu andan tespit edildiği ana kadar geçen ortalama süre.	(Uyarı Oluşturma Zamanı) - (Etkinlik Başlangıç Zamanı)	30 dakika - 4 saat
MTTR	Tehdide müdahale edilmesi için geçen ortalama süre.	(Kontrol Altına Alma Zamanı) - (Tespit Zamanı)	2 saat - 4 saat
MTTI	Bir uyarının doğrulanması ve incelenmesi için geçen ortalama süre.	(Araştırma Başlangıç Zamanı) - (Uyarı Zamanı)	Değişken
FPR	Yanlışlıkla tehdit olarak etiketlenen uyarıların yüzdesi.	(Hatalı Pozitif Sayısı) / (Toplam Uyarı Sayısı)	Kritik: $<$%25, Yüksek: $<$%50

Bu metrikler, SOC'nin kendi iç sağlığını ölçmesi ve iş hedeflerine nasıl katkıda bulunduğunu kanıtlaması için temel araçlardır. Yüksek bir MTTD, saldırganların sistemde kalış süresinin arttığını ve dolayısıyla iş riskinin yüksek olduğunu gösterir. Bu metrikler, sadece raporlama aracı olarak değil, aynı zamanda kök neden analizi ve iyileştirme için de kullanılır.

14.6.2 Alarm Yorgunluğu Azaltma ve Hatalı Pozitif Yönetimi

Alarm yorgunluğu, çok sayıda anlamsız uyarı nedeniyle analistlerin gerçek tehditleri gözden kaçırması ve tükenmesidir. Bu durum, analist moralini düşürür ve doğrudan bir güvenlik ihlali riskine yol açar.

Bu sorunu çözmek için aşağıdaki teknikler uygulanabilir:

– **Net Tanımlama:** Yalnızca acil eylem gerektiren uyarıları gerçek olarak kabul etmek ve diğerlerini raporlara kaydetmek.

- **Kural Temizleme:** Kullanılmayan veya gereksiz varsayılan kuralları devre dışı bırakmak.
- **Ortama Özel Ayarlama:** Kural eşik değerlerini organizasyonun ”normal” trafiğine göre ayarlamak. Bu, bir ağın temelini (baseline) belirlemeyi gerektirir.
- **Bağlam Kullanımı:** Varlık bilgisi ve tehdit istihbaratı gibi bağlamsal verilerle uyarıları zenginleştirmek. Bu, bir SQL saldırısı uyarısının, hedef sunucunun SQL çalıştırmadığı bilgisiyle otomatik olarak hatalı pozitif olarak belirlenmesini sağlayabilir.

14.6.3 Analist Eğitimi ve Beceri Geliştirme Programları

Siber tehdit ortamı sürekli evrildiği için analistlerin becerileri de sürekli güncellenmelidir. Analist eğitimi ve beceri geliştirme, bir SOC'nin uzun vadeli etkinliği için bir ek fayda değil, zorunlu bir operasyonel gerekliliktir.

Programlar, Microsoft Learn gibi platformlardan yapılandırılmış öğrenme yolları, L3 analistlerinin L1 ve L2 analistlerine mentorluk yapması ve proaktif tehdit avcılığı gibi uygulamalı alıştırmaları içermelidir. Teknoloji ne kadar gelişirse gelişsin, insan unsuru, başarılı bir SOC'nin en önemli faktörü olmaya devam etmektedir.

14.6.4 SOC Araç Konsolidasyonu ve Teknoloji Yol Haritası

Kuruluşların ortalama 60-75 güvenlik aracı kullandığı, bu durumun ”araç karmaşası”na (tool sprawl) ve veri silolarına yol açtığı belirtilir. Bu karmaşa, yönetim zorluğunu ve lisanslama maliyetlerini artırır ve kör noktalar oluşturur.

Araç konsolidasyonu, bu soruna stratejik bir yanıttır. Araçları birleştirerek, yönetim karmaşıklığı azaltılır ve ”tek bir cam bölmede” görünürlük artırılır. Bu süreç, reaktif, duruma göre güvenlik ürünü satın alma yaklaşımını bırakıp, daha uyumlu ve entegre bir güvenlik mimarisi oluşturmayı hedefler.

14.6.5 Yönetilen Güvenlik Hizmeti Sağlayıcı (MSSP) Değerlendirmesi

Dış kaynak kullanımına karar verildiğinde, bir MSSP'yi değerlendirme, bir hizmeti satın almaktan daha fazlasıdır; kontrol, esneklik ve maliyet arasında dikkatli bir denge kurmayı gerektiren stratejik bir ortaklık kararıdır.

Değerlendirme, sağlayıcının sunduğu SLA'ları, tehdit avcılığı ve adli bilişim gibi hizmetlerin derinliğini, kuruluşun iş modelini ve sektörünü ne kadar anladığını içermelidir. Bir sağlayıcının hizmetinin sözleşme koşullarıyla sınırlı olması, beklenmedik tehditlere karşı esnekliği sınırlayabilir. Bu nedenle, değerlendirme sürecinin, sağlayıcının teknolojik yeteneklerinin ötesinde, olası operasyonel riskleri de kapsaması önemlidir.

14.7 Güvenlik Operasyonlarında Yükselen Teknolojiler

Bu bölüm, güvenlik operasyonlarının geleceğini şekillendiren yeni ve gelişmekte olan teknolojileri incelemektedir.

14.7.1 SOC Operasyonlarında Yapay Zeka (AI)/Makine Öğrenimi (ML) Entegrasyonu

AI ve ML, SOC için dönüştürücü bir güçtür. Bu teknolojiler, geleneksel kural tabanlı sistemlerin atlayabileceği ince anomalileri ve desenleri tespit ederek tehdit algılamayı önemli ölçüde geliştirir. Aynı zamanda, rutin görevleri (uyarı triyajı gibi) otomatikleştirerek analistlerin iş yükünü hafifletir. AI/ML'nin en önemli özelliklerinden biri, sürekli öğrenme ve adapte olma yeteneğidir. Bu, algoritmaların tehdit ortamı geliştikçe kendilerini sürekli olarak güncellemesini sağlar. Ayrıca, hatalı pozitifleri filtrelemek için geçmiş olaylardan öğrenerek alarm yorgunluğunu azaltır.

14.7.2 Kullanıcı ve Varlık Davranış Analizi (UEBA) Uygulaması

UEBA, kullanıcıların, sunucuların ve diğer varlıkların normal davranışlarını profillendirerek, bunlardan sapmaları tespit eder. Bu teknoloji, kural tabanlı sistemlerin zayıf olduğu bir alanda, yani bir iç tehdidin veya ele geçirilmiş bir hesabın normal görünen davranışlarını tespit etmede kritik bir rol oynar. UEBA, iç tehditlerin, hesap ele geçirmelerinin ve geleneksel araçlar tarafından gözden kaçırılan gelişmiş kalıcı tehditlerin (APT'ler) tespiti için kullanılır.

14.7.3 Bulut-Yerel Güvenlik Operasyonları

Bulut-yerel mimariler (mikro hizmetler, konteynerler), güvenliği ağa dayalı bir yaklaşımdan, kimlik ve iş yükü odaklı bir yaklaşıma kaydırır. Bu değişim, SOC'nin bulut ortamlarındaki dinamik ve kısa ömürlü varlıkları izlemesini gerektirir. Bu, Kimlik ve Erişim Yönetimi (IAM), İş Yükü Güvenliği ve sürekli izleme gibi yeni odak alanları yaratır. Geleneksel güvenlik, bir duvarı korumaya odaklanırken, bulut güvenliği, duvarın içinde sürekli olarak her işlemi doğrulamaya odaklanır.

14.7.4 DevSecOps'un Güvenlik Operasyonları ile Entegrasyonu

DevSecOps, güvenliği yazılım geliştirme yaşam döngüsünün her aşamasına (geliştirme, test, dağıtım) entegre eden kültürel ve pratik bir yaklaşımdır. Bu entegrasyon, geliştirme, operasyon ve güvenlik ekipleri arasındaki siloları yıkar. Güvenlik açıklarını, bir ürün SOC'ye gelmeden önce geliştirme aşamasında bulup düzeltir. Bu proaktif yaklaşım, tehditleri algılamak ve bunlara yanıt vermek yerine, tehditleri oluştukları yerde önleyerek SOC'nin iş yükünü en temelden azaltan en güçlü stratejilerden biridir.

14.7.5 Sıfır Güven (Zero Trust) Mimarisi ve SOC Operasyonlarına Etkisi

Sıfır Güven, "Asla güvenme, her zaman doğrula" ilkesine dayanan bir güvenlik modelidir. Ağın içindeki hiçbir kullanıcıya veya cihaza varsayılan olarak güvenilmez. Bu mimari, SOC'nin görevini daha zorlu ve karmaşık hale getirirken, aynı zamanda iç tehditlere karşı daha dirençli olmasını sağlar.

Sıfır Güven, SOC'nin odak noktasını "ağ sınırını savunmak"tan, "içerideki her işlemi sürekli olarak izlemek ve doğrulamak"a kaydırır. Bu yaklaşım, her işlem loglandığı ve izlendiği için veri hacminde katlanarak artışa neden olur. Bu durum, UEBA ve Gelişmiş Tehdit Tespiti (AI/ML) gibi teknolojilerin, anormal iç hareketleri tespit etmek için temel gereksinim haline geldiğini gösterir. SOC, artık sadece dış tehditlerle değil, aynı zamanda içeriden gelen tehditlerle ve yanal hareketlerle de daha fazla mücadele etmek zorunda kalır. Sıfır Güven felsefesi, kural tabanlı sistemlerin her olası iç tehdit senaryosunu kapsayamayacağı gerçeğinden hareketle, davranışsal analize dayalı teknolojilerin önemini artırır.