SAÜ BİLGİSAYAR MÜHENDİSLİĞİ SIZMA TESTİ(PENTEST) EĞİTİMİ

BEN KİMİM?

- Ben Yasin ALTUNBAŞAK
- Sakarya Bilgisayar ve Bilişim Bilimleri Fakültesinde, Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisiyim.
- Lisans döneminde tanıştığım Siber Güvenlik alanında 5 yıldır aktif çalışmalar yapmaktayım.
- Siber Güvenlik alanındaki motivasyonum sistemlerde dışarıdan bakınca görülemeyeni görmek ve uygun bir şekilde müdahale etmektir.

EĞİTİM İÇERİĞİ

- Sızma Testi(Pentest) Nedir?
- Hacker Nedir? Hacker Türleri Nelerdir?
- Pentest Kavramları
- Zaafiyetlerin İncelenmesi
- Sosyal Mühendislik ve Phishing
- Skor Hesaplama
- Pentest Raporlama ve Güvenlik Ölçütleri
- TryHackme Lab Çözümleri

SIZMA TESTİ(PENTEST) Nedir?

- Sızma testi, bir kurumun bilgi sistemlerine **gerçek bir saldırgan gibi davranarak** güvenlik açıklarını bulmayı amaçlayan kontrollü bir güvenlik değerlendirmesidir. Bu testte amaç, saldırgan bakış açısıyla sisteme sızılabilir mi, kritik verilere erişilebilir mi ve mevcut güvenlik önlemleri saldırıya ne kadar dayanabilir sorularına yanıt bulmaktır.
- Pentest, sadece açıkları listeleyen bir tarama değildir; zafiyetlerin **gerçekte istismar edilip edilemeyeceğini**, ne kadar ileri gidilebildiğini ve bunun kuruma olan **iş etkisini** ortaya koyar. Bu nedenle kurumların, verilerini ve hizmetlerini korumak için düzenli pentest yaptırması, sistemlerinin güçlü ve zayıf yanlarının tespit edilerek risk değerlendirmesi sağlamak için yapılır. [1]

SIZMA TESTİ

- Sızma testleri, bir kurumun bilgi sistemlerini değerlendirmek için kullanılan yaklaşım ve bilgi seviyesine göre farklı türlere ayrılır. Kurumlar, hem dış tehditleri hem de içeriden gelebilecek riskleri doğru modellemek için farklı test türlerini kullanır. **Black Box**, **Gray Box** ve **White Box** yöntemleri, saldırganın bilgi seviyesini temel alan en yaygın sınıflandırmadır ve her biri farklı bir senaryoyu temsil eder.
- Bunun yanı sıra, pentestler kurumların dijital varlıklarını çok boyutlu değerlendirebilmek için dış ağ, iç ağ, web uygulamaları, mobil uygulamalar, kablosuz ağlar ve sosyal mühendislik gibi çeşitli hedeflere göre de yapılır. Böylece kurumlar gerçek tehditlerin farklı yüzlerini görebilir ve kapsamlı bir güvenlik resmi elde edebilir.

WHITE BOX(BEYAZ KUTU)

- Uzmanların sisteme dair **tam bilgi** ile test yaptığı en kapsamlı pentest modelidir. Kod, topoloji, konfigürasyon, API anahtarları ve ağ politikalarına erişim sağlanır. Bu model yalnızca zafiyeti bulmakla kalmaz, **problemin teknik kök nedenini** de ortaya çıkarır.
- **Kullanım Senaryoları**
  - Kurumsal uygulamalarda kaynak kodu güvenlik analizi
  - API güvenliği ve veri akışı testleri
  - Kritik altyapılar (ICS/OT), finans sistemleri, bankacılık yazılımları
  - Network mimarisi ve segmentasyon güvenliği denetimleri

AVANTAJLAR	DEZAVANTAJLAR
En kapsamlı güvenlik değerlendirmesidir; kod, konfigürasyon ve mimari hataları açığa çıkarır	Gerçek dış saldırgan senaryosunu tam olarak temsil etmez
Zafiyetlerin **kök neden analizi** yapılabilir; düzeltme önerileri daha net sunulur	Kurumdan yüksek seviyede bilgi ve dokümantasyon paylaşımı gerektirir
Güvenlik mekanizmalarının nerede zayıf, nerede yanlış yapılandırıldığını teknik olarak detaylı gösterir	Fazla bilgi, testçinin doğal saldırı akışını bozabilir (knowledge bias)
Kritik altyapılar ve finans uygulamalarında gereken derin kod güvenliği incelemesini sağlar	Elde edilen bulgular gerçek saldırıdan çok “teorik açıklık” seviyesinde olabilir

BLACK BOX(SİYAH KUTU)

- Test uzmanının hedef hakkında **hiçbir iç bilgiye sahip olmadığı**, tamamen dış saldırgan perspektifiyle yürütülen pentest yöntemidir. Sistemin dışarıya açık yüzeyindeki (attack surface) en küçük zayıflığın bile ne kadar ileri taşınabileceğini gösterir. Bu model, özellikle **keşif (recon)** aşamasına ve servis davranışlarının analitik olarak yorumlanmasına dayanır.
- **Kullanım Senaryoları**
  - İnternete açık web servislerinin güvenlik seviyesi ölçümü
  - Kurumsal dış yüzey taraması (external attack surface)
  - WAF/Firewall arkasında unutulmuş portları bulma
  - Yeni açılmış bir servisin saldırı yüzeyini değerlendirme

AVANTAJLAR	DEZAVANTAJLAR
Gerçek dış saldırganın yetenek ve kısıtlarını en doğru şekilde modeller	İç yapı, topoloji ve kod tabanlı zafiyetler görünmez kalabilir
Dışarıya açık servislerdeki yanlış konfigürasyonları ve “unuttulmuş” sistemleri hızlı tespit eder	Keşif aşaması zaman alır; sonuçlar testçinin OSINT ve analiz becerisine çok bağlıdır
Firewall, WAF, IDS/IPS gibi savunmaların gerçek saldırı karşısındaki davranışını ölçer	Kurumun iç risk profilini tam olarak yansıtmaz
Sosyal mühendislik + dış ağ kombinasyonlarında yüksek etki sağlar	Test süreci öngörülemez; bazı hedeflerde boş geçebilir

GRAY BOX(GRİ KUTU)

- Uzmanın hedefe dair **sınırlı bilgi** ile hareket ettiği hibrit pentest yaklaşımıdır. Tipik olarak düşük yetkili kullanıcı hesabı veya belirli bir API bilgisinin verilmesiyle yapılır. Hem iç hem dış tehdit senaryosunu aynı modelde değerlendirdiği için kurumsal yapılarda en gerçekçi güvenlik görünümünü sunar.
- **Kullanım Senaryoları**
  - Uygulamada normal kullanıcı perspektifiyle güvenlik testi
  - Kimlik yönetimi (IAM) ve erişim kontrolü denetimi
  - Oturum yapısının ve yetkilendirme mekanizmasının sınanması
  - İçeriden gelen tehditlerin modellenmesi

AVANTAJLAR	DEZAVANTAJLAR
Dış ve iç tehditleri aynı testte birleştirerek **kurumsal gerçeklikle en uyumlu** modeli sunar	Gerçek saldırganın sahip olmayacağı bilgiler testçi lehine avantaj yaratabilir
Yetki yükseltme, oturum yönetimi, erişim kontrolü gibi kritik alanlarda çok daha doğru sonuç verir	Verilen bilgi seviyesinin yanlış belirlenmesi test kapsamını etkileyebilir
Performans–kapsam dengesini en iyi sağlayan yöntemdir	İç ağ mimarisi hakkında White Box kadar derin bir resim sunmaz
Hem mantıksal hataları hem pratik istismar yollarını aynı anda görünür kılar	Testçiye(pentester) verilen erişim sınırları net tanımlanmazsa sonuçlar değişken olabilir

SIZMA TESTİ(PENTEST) SÜRECİ

- Sızma testi, rastgele yapılan saldırı denemelerinden ziyade belirli adımları takip eden metodolojik bir süreçtir. NIST, PTES ve SANS metodolojilerinde görüldüğü üzere tüm pentestler ortak bir yaşam döngüsü izler. Bu döngü, saldırgan davranışlarını kontrollü bir çerçevede taklit ederek kurumun güvenlik seviyesini gerçekçi şekilde ölçmeyi amaçlar.
- Pentest süreci; kapsam, kurallar ve teknik sınırların belirlendiği **Planlama & Kapsam** ile başlar. Ardından hedef sistem hakkında veri toplamak için **Bilgi Toplama (Recon)** aşaması yürütülür. Toplanan veriler, açıklık eşleştirmesi ve risk analizi yapılan **Zafiyet Analizi** ile değerlendirilir. Uygun görülen açıklıklar **İstismar (Exploitation)** aşamasında test edilir; Bu aşamada sömürülen açıklar üzerinden yetki yükseltme ve erişim kalıcılığı ve etkisi incelenir. Bu aşamadan sonra tüm bulgular teknik ve iş etkisi açısından değerlendirilerek **Raporlama** ile süreç tamamlanır.

PLANLAMA VE KAPSAM

- Sızma testlerinin güvenli, hukuki ve teknik olarak doğru yürütülmesi için planlama ve kapsam belirleme aşaması kritik öneme sahiptir. Bu aşamada pentest ekibi ile kurum arasında **resmi bir anlaşma (Authorization Letter / Rules of Engagement)** hazırlanır ve testin sınırları kesin çizgilerle tanımlanır. Kapsam formu; hangi varlıklara test yapılacağını, hangi yöntemlerin kullanılabileceğini ve hangi risklerin kabul edildiğini belirleyen temel dokümandır.
- Kapsam formunda genellikle hedef alınacak **IP aralıkları, domain ve alt domain listeleri, bulut ortamları, iç/dış ağ segmentleri**, test dışı bırakılan kritik üretim sistemleri, sosyal mühendislik izninin olup olmadığı, DoS/DDoS benzeri sistem kesintisine neden olabilecek testlerin yasaklanıp yasaklanmadığı açıkça belirtilir. Ayrıca test türüne (Black/Gray/White Box) göre şirkete verilecek bilgi düzeyi; örneğin kullanıcı hesapları, VPN erişimi, kod veya mimari doküman paylaşımı gibi unsurlar da bu aşamada netleştirilir.

PLANLAMA VE KAPSAM

- Bazı kurumlar kapsam dahilinde **fiziksel sızma testlerini** de talep eder. Bu testlerde pentester, kurum binasına yetkili bir çalışan gibi giriş yapmaya çalışabilir, güvenlik kontrollerini atlatmayı deneyebilir veya sosyal mühendislik yoluyla fiziksel erişim elde etmeye çalışabilir. Fiziksel testler, dijital saldırılarla kombine edildiğinde (ör. kablolu ağ erişimi, iç switch bağlantısı, masaüstü cihaz kullanımı) gerçekçi ve çok katmanlı risk görünürlüğü sağlar. Bu nedenle birçok uluslararası firmada fiziksel pentest, kapsam planlamasının standart bir unsurudur.
- Planlama ve kapsam aşaması, yalnızca hedeflerin belirlenmesinden ibaret değildir; aynı zamanda test süresini, iletişim protokollerini, acil durumda aranacak kişileri, log paylaşım süreçlerini ve raporlama formatını da içerir. Bu aşamadaki her eksiklik, sürecin yanlış yönetilmesine veya hukuki risklere yol açabileceğinden, profesyonel ortamlarda kapsam formu pentestin en önemli adımı olarak kabul edilir.

BİLGİ TOPLAMA(RECON)

- Bilgi toplama, sızma testinin saldırı zincirindeki ilk teknik aşamasıdır ve hedef sistem, kullanıcılar, altyapı, teknolojiler ve potansiyel saldırı yüzeyi hakkında **mümkün olan en fazla verinin toplanmasını** amaçlar. Bu aşama, tüm pentestin kalitesini belirlediği için profesyonel süreçlerde testin en önemli ve en uzun adımlarından biri kabul edilir. Elde edilen bilgiler, sonraki aşamalarda istismar edilebilecek zafiyetlerin nerede ve nasıl aranacağı konusunda doğrudan yol gösterici olur.
- Bilgi Toplama Süreci Pasif ve Aktif Bilgi toplama adı altında ikiye ayrılır.

PASİF BİLGİ TOPLAMA

- Pasif bilgi toplama, hedef sistemle **doğrudan etkileşime girmeden**, üçüncü taraf kaynaklardan ve internet üzerindeki açık verilerden bilgi çekme yöntemidir. Bu yaklaşım, hedefin loglarında iz bırakmadığı için düşük riskli ve "gizli" kabul edilir. Amaç, hedefin fark etmediği bir şekilde saldırı yüzeyini keşfetmektir.
- **Pasif teknikler:**
  - WHOIS, DNS kayıtları, registrar verileri
  - Shodan/Censys gibi tarama motorlarından servis bilgileri
  - Veri sızıntısı arama (HaveIBeenPwned, paste siteleri)
  - Google dorking ve açık dizin aramaları
  - Çalışan bilgileri, e-posta formatları (LinkedIn, sosyal medya)
  - Teknoloji tespiti (BuiltWith, Wappalyzer)
  - Sertifika kayıtları (crt.sh → subdomain keşfi)
  - Basın açıklamaları, proje dökümanları, ihale belgeleri
  - Github, GitLab gibi ortamlarda açık kalmış yapılandırma dosyaları
  - Pasif recon, aktif testlere yol gösteren "keşif haritası" niteliğindedir. Profesyonel pentestlerde pasif aşama ne kadar derin ve özenli yapılırsa, sonraki aşamalarda başarı oranı o kadar artar.

AKTİF BİLGİ TOPLAMA

- Aktif bilgi toplama, hedef sistemle **doğrudan etkileşime girilerek** çalışan servisler, portlar, versiyonlar, yapılandırmalar ve ağ davranışlarının teknik olarak analiz edilmesidir. Bu aşamada yapılan birçok adım hedef tarafından loglanabilir; dolayısıyla profesyonel pentester, aktif teknikleri kontrollü ve kapsam formuna uygun şekilde yürütmelidir.
- **Aktif teknikler:**
  - Port taraması (Nmap, Masscan)
  - Servis/versiyon analizi (banner grabbing)
  - Alt domain brute force ve DNS zone transfer denemeleri
  - HTTP probing (web dizin tarama → dirsearch, gobuster)
  - SMB/NFS/RDP/SSH servis keşfi
  - SNMP numaralandırma (public/private strings)
  - SIP, FTP, SMTP gibi protokollerde numaralandırma
  - WAF/firewall davranış analizi
  - Ağ haritalama ve topoloji çıkarma
  - Aktif recon, elde edilen verileri **istismar edilebilir saldırı yollarına** dönüştürür. Saldırı yüzeyinin hangi noktalarının zayıf olduğu, hangi servislerin yanlış yapılandırıldığı ve hangi teknolojilerin zafiyet barındırdığı büyük ölçüde bu aşamada belirlenir.

ZAFİYET ANALİZİ

- Zafiyet analizi, bilgi toplama aşamasında elde edilen verilerin hedefin barındırdığı potansiyel açıklıklarla karşılaştırıldığı ve doğrulandığı aşamadır. Bu süreç hem **otomatik tarama araçlarıyla yapılan hızlı analizleri** hem de **manuel testçi uzmanlığıyla yapılan derinlemesine doğrulamaları** içerir. Amaç, sistem üzerindeki gerçek riskleri ortaya çıkarmak ve istismar edilme olasılığı bulunan açıklıkları belirlemektir.

OTOMATİK ZAFİYET ANALİZİ

- Bu bölümde hedef sistem tarayıcı araçlarla analiz edilerek bilinen açıklıklar, yanlış yapılandırmalar ve riskli servisler hızlıca tespit edilir. Otomatik tarama geniş kapsam sağlar ancak “yanlış pozitif” üretme oranı yüksektir, bu yüzden sonuçlar mutlaka manuel olarak doğrulanmalıdır. Otomatik tarama, “nerede sorun olabilir?” sorusuna hızlı yanıt verir ve istismar aşaması için başlangıç noktası oluşturur.
- **Kullanılan popüler araçlar:**
  - **Nessus**, **OpenVAS**, **Nexpose / InsightVM**
  - **Nmap + NSE Scriptleri** (CVE detection, http-vuln-* scriptleri)
  - **Nikto** (web sunucu tarayıcısı)
  - **Acunetix**, **Burp Suite Scanner** (web uygulama taraması)
  - **Wapiti**, **Arachni**, **OWASP ZAP** (web vulnerability scanners)

MANUEL ZAFİYET ANALİZİ

- Manuel analiz, otomatik taramaların ürettiği sonuçları doğrulamak ve yalnızca insan bakış açısıyla ortaya çıkarılabilen mantıksal ve davranışsal zafiyetleri tespit etmek için yapılır. Profesyonel pentestlerin en kritik kısmı manuel doğrulamadır. Manuel analiz olmadan yapılan bir zafiyet analizi, profesyonel pentest standartlarına göre eksik kabul edilir.
- **Manuel analizde kullanılan araç ve yöntemler:**
  - **Burp Suite Professional / Community** (proxy, repeater, intruder, decoder, comparer)
  - **Postman / Insomnia** (API testleri)
  - **FoxyProxy + tarayıcı geliştirici araçları**
  - **WFuzz / FFUF** (endpoint brute-force, parametre tarama)
  - **Kerbrute**, **Impacket** (AD servisleri manuel testleri)
  - **SMBClient**, **RPCClient**, **CrackMapExec** (ağ servisleri numaralandırma)
  - **Manuel teknik kontroller:**
    - IDOR denemeleri
    - Access Control zafiyetleri
    - Parametre manipülasyonu
    - Mantık hataları (logic flaws)
    - Yetkilendirme zayıflıkları
    - Session management testleri
    - Misconfiguration analizleri (CORS, header hataları, SSL konfigürasyonu vb.)

SÖMÜRME(EXPLOITATION) SÜRECİ

- Zafiyet analizinden sonra başlayan bu aşama, sızma testinin hedef sistem üzerinde gerçek saldırgan davranışını modelleyen en kritik bölümünü oluşturur. Önce tespit edilen açıklıklar istismar edilerek sisteme **ilk erişim** sağlanır, ardından mevcut haklar genişletilerek **daha yüksek yetkiler** elde edilir ve son olarak erişimin **ağın geri kalanındaki etkisi**, kalıcılık yöntemleri ve veri güvenliği üzerindeki olası sonuçları değerlendirilir.
- Bu süreç; ilk giriş noktasından başlayarak saldırganın sistem içinde nereye kadar ilerleyebileceğini, hangi savunma mekanizmalarını aşabildiğini ve kurumun iç ağ yapısının ne kadar dayanıklı olduğunu ortaya koyması açısından profesyonel pentestlerde temel değerlendirme alanıdır.

EXPLOITATION

- **Exploitation**, sızma testinin hedef sisteme giriş noktası sağlayan fazıdır. Recon ve zafiyet analizinde tespit edilen açıklıkların, teknik saldırı yöntemleri kullanılarak **gerçek anlamda istismar edilmesi** bu aşamada gerçekleşir. Exploitation, bir açıklığın gerçekten "tehlikeli" olup olmadığını, yani istismar edilebilirlik seviyesini ortaya koyar.
- Bu aşamanın temel amacı yalnızca sisteme erişim sağlamak değil; **zafiyetin kurum açısından doğurduğu gerçek iş etkisini** gösterebilmektir. Başarılı bir exploitation, saldırganın nasıl içeri girebileceğini, hangi bileşenlerin savunmasız olduğunu ve hangi savunma mekanizmalarının (WAF, EDR, AV, IDS) atlatılabileceğini belirgin biçimde ortaya çıkarır.
- Profesyonel pentestlerde exploitation, MITRE ATT&CK'in **Initial Access (TA0001)** ve **Execution (TA0002)** taktikleriyle doğrudan bağlantılıdır; yani bu faz, saldırı zincirinin gerçek saldırgan davranışlarını en çok yansıtan adımıdır.

YAYGIN EXPLOITATION TEKNİKLERİ

1. Public-Facing Application Exploitation(Web uygulamalarında açıklıkların istismarı)

- İnternete açık web uygulamalarındaki zafiyetlerin kullanılarak sisteme ilk erişimin sağlandığı aşamadır. SQL Injection, Command Injection, LFI/RFI, SSTI ve Authentication Bypass gibi tekniklerle veritabanı, uygulama mantığı veya sunucu üzerinde kontrol elde edilir. Burp Suite Pro (Intruder/Repeater), SQLMap, FFUF, WFuzz, OWASP ZAP, Arachni ve Nuclei bu fazda kullanılan temel araçlardır.

  SQL Injection → Veritabanı Ele Geçirme → RCE Zinciri

  – Saldırgan önce SQL Injection ile tablo ve kullanıcı bilgilerini toplar.
  Ardından DB üzerinden komut çalıştırmayı sağlayan fonksiyonları (xp_cmdshell, UDF injection, stacked query) kullanarak uygulama sunucusunda **komut yürütme** elde eder.
  Bu zincir genelde **reverse shell** ile sonuçlanır.

  File Upload → Zararlı Dosya → Web Shell

  – Uygulamada yüklenen dosyaların düzgün filtrelenmemesi durumunda saldırgan kendine ait bir PHP/ASPX web shell yükler.
  Bu shell üzerinden sunucu üzerinde komut çalıştırabilir, dosya okuyabilir veya yeni payload'lar indirebilir.
  Bu senaryo gerçek saldırıların **%60+**ında görülür.

  LFI → Log Poisoning → Shell Alma

  – LFI ile /var/log/apache2/access.log gibi dosyalar okunur.
  Saldırgan buraya RCE içeren bir user-agent göndererek log dosyasını "enjekte eder".
  LFI üzerinden log dosyasını çalıştırarak **komut yürütme** elde eder.

  Weak JWT / IDOR → Admin Panel Ele Geçirme

  – Zayıf imzalı JWT token kırılır veya predictable user IDs ile admin endpoint'lere erişilir.
  Sisteme giriş yapılmadan yönetim paneli kontrol altına alınır.

YAYGIN EXPLOITATION TEKNİKLERİ

2. Network Service Exploitation(Sunucu servislerinin açıklıklarının istismarı)

- SMB, RDP, LDAP, FTP, SSH, SMTP, SNMP gibi servislerdeki yanlış yapılandırmalar veya zafiyetler üzerinden istismardır. EternalBlue, PrintNightmare, Zerologon gibi bilinen exploitler bu kategoriye girer. Impacket (psexec, smbexec, wmiexec, ntlmrelayx), CrackMapExec, Nmap NSE scriptleri, Responder, MITM6 gibi araçlar burada kritik rol oynar.

  SMB Signing Off → NTLM Relay → Admin Erişimi

  – Ağdaki bir cihaz NTLM isteği gönderir.
  Responder bu isteği alır ve smbrelayx ile Domain Controller'a yönlendirir.
  SMB signing kapalıysa saldırgan **administrator** olarak sistemi kontrol eder.

  SNMP "public/private" String → Yapılandırma Sızıntısı

  – SNMP community string varsayılan bırakılmışsa:
  Saldırgan router/switch yapılandırmasını okur → IP yapısı, VLAN'lar, admin şifre politikaları görülür.
  Bu bilgiler privilege escalation(yetki yükseltme) aşamasında kritik hale gelir.

  RDP Misconfiguration → Unauthorized Login

  – Network-level authentication (NLA) kapalıdır veya zayıf parola kullanılmıştır.
  Saldırgan brute force veya credential stuffing ile doğrudan RDP üzerinden sisteme giriş yapar.

  FTP / SSH / SMTP / POP3 / IMAP Exploitation

  – Bu kategori, kurumların temel iletişim ve yönetim servislerinde yapılan yanlış yapılandırmaların istismar edilmesiyle gerçekleşir. Genellikle "küçük görünen" hatalar zincirlenerek **ilk erişim**, **credential toplama** veya **iç servislere pivot** elde edilir.

YAYGIN EXPLOITATION TEKNİKLERİ

3. Remote Code Execution (RCE)

- RCE, saldırganın hedef sistem üzerinde **uzaktan komut çalıştırmasına** imkân veren en kritik zafiyet türüdür. Bir RCE açığı, doğrudan sistemin tam kontrolünü ele geçirmeye kadar gidebilen yüksek etkili bir saldırı zincirinin başlangıcıdır. Bu nedenle pentest süreçlerinde RCE, "en tehlikeli açıklık" sınıfında değerlendirilir ve acil müdahale gerektirir.

  Log İşleme ve Input Injection Kaynaklı RCE

  – Log4j (Log4Shell), Apache Struts, Spring4Shell gibi zafiyetler kullanıcı girişinin loglama/işleme süreçlerinde tehlikeli fonksiyonlara aktarılmasıyla oluşur.
  Saldırgan özel crafted payload gönderir → uygulama bunu JNDI, OGNL veya template motoru üzerinden işler → uzaktan kod çalıştırma gerçekleşir.

  Deserialization Vulnerabilities ile RCE

  – Java, PHP, .NET gibi platformlarda serialize edilmiş nesnelerin güvenli doğrulanmadan tekrar açılması (deserialize edilmesi) RCE'ye yol açabilir.
  Saldırgan bir **gadget chain** içeren zararlı nesne gönderir → uygulama deserialize ederken zincir tetiklenir → komut yürütülür.

  File Upload Abuse → Web Shell / Command Execution

  – Yüklenen dosyaların tür kontrolü veya MIME doğrulaması doğru yapılmadığında saldırgan .php, .aspx, .jsp gibi dosyalar yükleyebilir.Sunucu bu dosyayı çalıştırdığında saldırgan web shell elde eder.

YAYGIN EXPLOITATION TEKNİKLERİ

4. Credential & Authentication Attacks

- Credential & Authentication Attacks, sistemlerdeki kimlik doğrulama mekanizmalarının zayıflıklarını hedef alan saldırı sınıfıdır. Amaç; kullanıcı parola politikalarını, oturum yönetimini, protokol davranışlarını veya sızıntıya uğramış kimlik bilgilerini kullanarak **doğrudan yetkili erişim** elde etmektir. Bu kategori, exploitation zincirinde "no noise initial access" olarak bilinir—yani saldırgan çok az iz bırakarak sisteme giriş yapabilir.

  Password Spraying ile Sessiz İlk Erişim

  – Saldırgan tek bir parola (ör. Password123) dener.
  Hesap kilitlemesi yaşanmadan bir çalışan hesabı ele geçirilir.
  Bu yöntem **en sessiz initial access senaryosudur**.

  Kerberos Pre-Auth Disabled → AS-REP Roasting

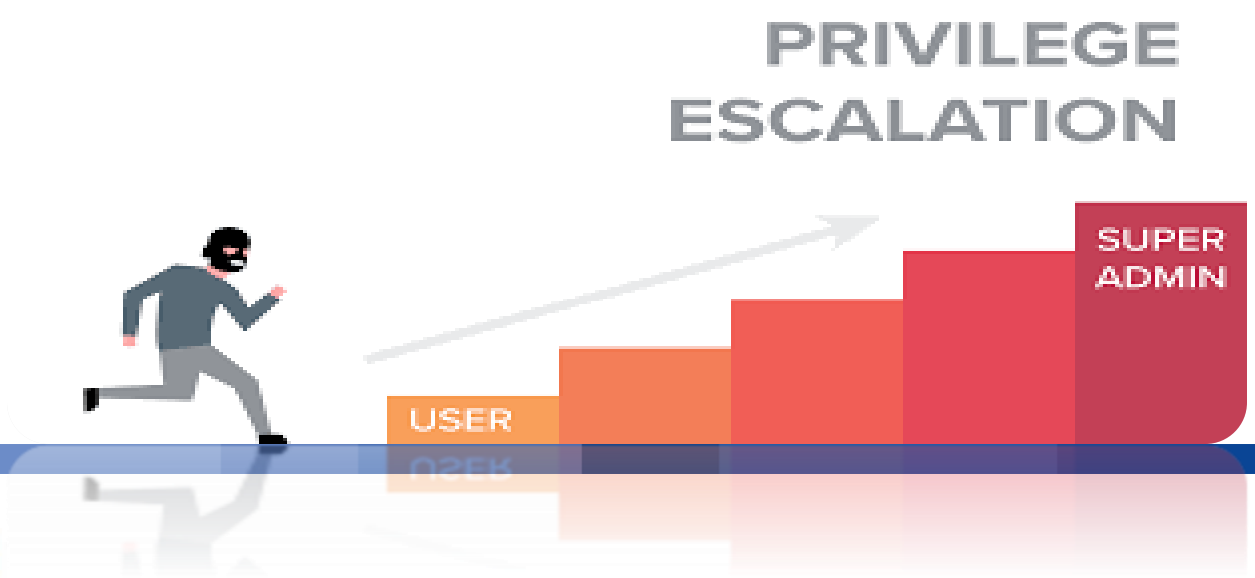
  – Domain'de bazı kullanıcı hesaplarının Kerberos pre-auth özelliği kapalıdır.
  Saldırgan KDC'den hash'lenmiş ticket alır → Hashcat ile kırar → doğrudan parola elde eder.

  Session Hijacking → MFA Bypass

  – Bir endpoint güvenli olmayan cookie tutuyorsa, saldırgan cookie ele geçirir ve MFA devre dışı kalarak oturum açabilir.
  (Modern web exploitation'da en kritik zincirlerden biri.)

PRIVILEGE ESCALATION (YETKİ YÜKSELTME)

- Exploitation aşamasında ilk erişim sağlandıktan sonra saldırgan için süreç burada bitmez; aksine gerçek saldırı zinciri tam bu noktada derinleşmeye başlar. Sisteme düşük yetkilerle girilmiş olması çoğu zaman yalnızca bir adım taşından ibarettir. Privilege Escalation, elde edilen bu başlangıç erişimin **daha yüksek haklara taşınması**, yani saldırganın sistem ve ağ üzerinde **kalıcı, daha güçlü ve daha geniş bir kontrol** kurduğu sürecin doğal devamıdır.
- Bu aşamada amaç; exploitation ile ele geçirilen düşük yetkili oturumu kullanarak işletim sistemindeki izin hatalarını, servis yapılandırmalarını, credential depolarını ve Active Directory ilişkilerini analiz edip sömürmek ve böylece **SYSTEM**, **root** veya **Domain Admin** gibi üst düzey haklara ulaşmaktır. Başarılı bir exploitation'ın gerçek etkisi ancak bu adımda ortaya çıkar; yetki yükseltme olmadan bir saldırganın sisteme giriş yapmış olması çoğu zaman sınırlı etki yaratır.



PRIVILEGE ESCALATION

1. Local Privilege Escalation (Windows & Linux)

- Local Privilege Escalation, exploitation sonrası elde edilen **low-priv shell'i** kullanarak aynı makine üzerinde **SYSTEM/root** seviyesine çıkma sürecidir. Amaç, işletim sistemi üzerindeki izin hatalarını, servis yapılandırmalarını veya kernel açıklarını sömürerek makinayı tamamen kontrol edebilecek yetkiye ulaşmaktır.

Örnek teknikler:

- Windows: Unquoted service path, zayıf service ACL'leri, AlwaysInstallElevated, UAC bypass, DLL hijacking
- Linux: SUID/SUDO yanlış yapılandırmaları (sudo -l ile tespit), cron job hijacking, writable script'ler, eski kernel exploit'leri (DirtyCow, DirtyPipe)
- Yanlış dosya izinleri: Log, backup veya script dosyalarının yazılabilir olması
- PATH hijacking: Yetkili script'in $PATH içindeki farklı bir binary'yi çağırması
- **Sık kullanılan araçlar:**
- **WinPEAS**, **Seatbelt**, **PowerUp** (Windows otomatik enumeration)
- **LinPEAS**, **pspy**, manuel sudo -l, find / -perm -4000 (Linux)

PRIVILEGE ESCALATION

2. Credential Harvesting & Token Abuse

- Credential Harvesting & Token Abuse , makinede veya oturumda mevcut olan **parola, hash, ticket ve access token** gibi kimlik bilgilerini toplayarak yetki yükseltmeyi hedefler. Çoğu gerçek saldırıda "asıl sıçrama" bu aşamada gerçekleşir; saldırgan, makinede zaten kayıtlı olan kimlik bilgilerini çekip daha üst seviye hesaplara geçer.

Örnek teknikler:

- LSASS dump alarak (mini dump / nanodump) içinden NTLM hash ve cleartext parolaları çıkarmak
- SAM + SYSTEM hive'larını export edip offline olarak hash kırmak
- Kerberos ticket'larını çalmak (**Pass-the-Ticket**, ticket reuse)
- Access token impersonation ile başka bir sürecin yetkilerini devralmak
- Tarayıcı, RDP, VPN client gibi uygulamaların credential store'larından parola çekmek
- **Sık kullanılan araçlar:**
- **Mimikatz**, **Rubeus**, **LaZagne**, **SharpDPAPI**
- sekurlsa::logonpasswords, lsadump::sam, dpapi::credential benzeri komutlar

PRIVILEGE ESCALATION

3. Service & Application Misconfigurations

- Servisler ve uygulamalar çoğu zaman **yüksek yetkiyle (SYSTEM/root)** çalışır; bu yüzden üzerlerindeki küçük bir yanlış yapılandırma bile doğrudan tam yetkili shell'e dönüşebilir. Burada hedef, servislerin nasıl başlatıldığını, hangi dosyaları hangi izinlerle kullandığını analiz edip bu zincire zararlı bir bileşen enjekte etmektir.

Örnek teknikler:

- Windows'ta **unquoted service path**: "C:\Program Files\My Service\service.exe" gibi yolda boşluklar varsa araya kendi exe'nizi koymak
- Service binary'si veya çalıştığı dizin yazılabilir ise → binary'yi değiştirip servis restart ile SYSTEM almak
- Linux'ta root ile çalışan cron job'ların yazılabilir script'lere işaret etmesi
- Startup uygulamalarının, PATH'te saldırganın yazabildiği bir klasörden binary çağırması (PATH hijacking)
- Config dosyalarının içinde hassas credential veya komut satırı çağrıları bulunması
- **Sık kullanılan araçlar:**
- **WinPEAS**, **PowerUp**, **accesschk** (Windows servis izin analizi)
- **LinPEAS**, systemctl, crontab -l, ls -l /etc/cron* (Linux servis/cron analizi)

PRIVILEGE ESCALATION

4. Active Directory (AD) Privilege Escalation

- Kurumsal ortamlarda gerçek güç, tek makinede değil, **domain üzerinde** yetki almaktır. AD Privilege Escalation, domain içi kullanıcı–grup–ACL ilişkilerini, service account'ları ve yanlış yapılandırılmış izinleri kullanarak **Domain Admin** gibi üst düzey hesaplara sıçramayı hedefler.

Örnek teknikler:

- **Kerberoasting**: SPN tanımlı service account'ların ticket'larını çekip offline hash kırma
- **AS-REP Roasting**: Pre-auth disabled kullanıcılar için KDC'den alınan hash'leri kırarak parola elde etme
- **Yanlış ACL/ACE:** Bir kullanıcının başka bir kullanıcıya ResetPassword, GenericAll, GenericWrite yetkisine sahip olması
- **GPO Abuse**: Yanlış yapılandırılmış Group Policy üzerinden domain makinelerine zararlı script enjekte etmek
- **Shadow Credentials / Certificate Abuse**: Sertifika tabanlı kimlik doğrulama mekanizmasının kötüye kullanılması
- **Sık kullanılan araçlar:**
- **BloodHound / SharpHound** (AD ilişki haritalama ve privesc yolu bulma)
- **Rubeus**, **Impacket** (Kerberos saldırıları)
- **CrackMapExec**, **ADRecon**, Get-AD* PowerShell cmdlet'leri

POST-EXPLOITATION

- Exploitation ve Privilege Escalation aşamalarından sonra saldırgan artık sistemin içinde aktif olarak hareket edebilir hale gelir. Post-Exploitation, elde edilen erişimin *gerçek etkisini* ortaya koyan, saldırgan davranışının en kritik ve en kapsamlı bölümüdür. Bu aşamanın amacı, içeri girilen sistemi yalnızca kontrol etmek değil; aynı zamanda ağın geri kalanını haritalamak, yetkileri genişletmek, kalıcı bir yer edinmek, kimlik bilgilerini toplamak ve hedefin iş süreçleri üzerinde ne kadar etki oluşturulabileceğini ölçmektir.
- Bu faz, gerçek saldırganların kurum içindeki yolculuğunu modellediği için pentest çalışmalarında en fazla değer üreten bölümlerden biridir. Post-Exploitation sayesinde sadece "sisteme girilebildiği" değil, aynı zamanda içeride ne kadar ilerlenebildiği, hangi verilerin tehlikede olduğu ve saldırganın ne kadar süre tespit edilmeden kalabileceği objektif olarak gösterilir.

POST-EXPLOITATION

1. Lateral Movement(Sistemde Yayılma Hareketi)

- Lateral Movement, saldırganın ele geçirdiği bir makineden ağ içindeki **diğer sistemlere doğru genişlemesi** sürecidir. Post-Exploitation'ın en kritik aşamalarından biridir çünkü saldırgan artık sadece bir makineyi değil, bütün ağı hedefleyebilir hale gelir. Bu aşamada amaç; mevcut kimlik bilgileri, servis izinleri, oturum anahtarları ve paylaşımlar kullanılarak ağ içinde **sessiz ve kontrollü bir şekilde ilerlemek**, daha yetkili sistemlere ulaşmak ve nihayetinde kritik sunuculara veya domain controller'a erişim sağlamaktır. Lateral Movement saldırganı "bir noktadan" çıkarıp "ağın tamamına yayılabilir" hale getirdiği için, gerçek dünyada kurum içi ihlallerin en tehlikeli bölümünü oluşturur.

Lateral Movement Teknikleri

- **Pass-the-Hash / Pass-the-Ticket**
  Ele geçirilen NTLM hash veya Kerberos ticket, parola olmadan diğer makinelere giriş için kullanılır.
- **Psexec / WMI / WinRM üzerinden uzaktan komut yürütme**
  Windows ağlarında Impacket araçlarıyla yapılan en sessiz ve hızlı yayılma yöntemleri.
- **SMB üzerinden session reuse**
  Ele geçirilen oturumu SMB servislerinde yeniden kullanarak farklı sistemlere sıçrama.
- **RDP / SSH pivoting**
  Elde edilen kullanıcıyla uzak masaüstü veya terminal oturumu açarak yeni makinelere giriş.
- **Shared folder & admin share abuse**
  C$, ADMIN$, IPC$ gibi yönetim paylaşımlarının saldırgan tarafından kötüye kullanılması.
- **Service account privilege reuse**
  Bir servisin çalıştığı hesaba erişildiğinde genellikle ağ genelinde çok daha fazla yetki bulunur.

POST-EXPLOITATION

2. Persistence (Kalıcı Erişim)

- Persistence, saldırganın sisteme bir kez sızdıktan sonra erişimini **kalıcı hale getirme** sürecidir. Amaç; sistem yeniden başlasa, kullanıcı oturumu kapansa veya saldırgan bağlantıyı kaybetse bile hedefe tekrar erişebilmektir. Bu aşama, gerçek saldırgan davranışını taklit eden Post-Exploitation tekniklerinin en önemli parçalarındandır.

Persistence Teknikleri

- **Scheduled Tasks / Cron Jobs**
  Windows'ta zamanlanmış görev, Linux'ta cron job oluşturarak saldırgan komut veya payload'un düzenli aralıklarla tekrar çalışması sağlanır.
- **Registry Run Keys (Windows)**
  HKCU\Software\Microsoft\Windows\CurrentVersion\Run gibi anahtarlara zararlı bir komut eklenerek sistem her açıldığında saldırgan kodu tetiklenir.
- **Startup Application Abuse**
  Başlangıç klasörüne veya systemd/service yapılarına eklenen arka kapı uygulamaları ile yeniden başlatmalar engel olmadan kalıcı erişim sağlanır.
- **SSH Key Persistence (Linux)**
  ~/.ssh/authorized_keys içine saldırganın public key'i eklenir; böylece parola gerekmeden tekrar giriş yapılabilir.
- **User/Group Manipulation**
  Yeni yönetici kullanıcı oluşturmak veya mevcut bir hesabı daha yetkili gruplara eklemek.

POST-EXPLOITATION

3. Credential Harvesting (Kimlik Bilgisi Toplama)

- Credential Harvesting, Post-Exploitation aşamasında saldırganın sistemlerde mevcut olan **parola, hash, token, ticket ve oturum anahtarlarını** toplayarak daha yüksek yetkiler veya yeni erişim noktaları elde etme sürecidir. Bu yöntem çoğu gerçek saldırıda "asıl sıçrama noktası" olarak görülür çünkü çoğu zaman saldırganın aradığı kritik kimlik bilgileri zaten makinenin içinde depolanmıştır.

Credential Harvesting Teknikleri

- **LSASS Memory Dump (Windows)**
  LSASS sürecinin belleği çıkarılarak hem NTLM hash'leri hem de bazı durumlarda cleartext parolalar elde edilir.
  Araçlar: Mimikatz, nanodump, procdump
- **Credential Store Harvesting**
  Tarayıcı parolaları, RDP kayıtlı oturumları, VPN client bilgileri ve uygulama içi credential depoları toplanır.
  Araçlar: LaZagne, SharpDPAPI, Browser forensic tools
- **Token Impersonation & Delegation**
  Access token çalarak başka bir kullanıcının yetkilerini devralmak mümkün olur.
  Araçlar: Mimikatz (token::list / token::impersonate), Incognito
- **SSH / Private Key Extraction (Linux & Windows)**
  SSH private key dosyaları veya agent oturumları ele geçirilerek oturum açma sağlanır.

POST-EXPLOITATION

4. Data Exfiltration (Veri Sızdırma)

- Data Exfiltration, saldırganın ağ içinde ele geçirdiği kritik verileri **fark edilmeden dışarı çıkarması** sürecidir. Post-Exploitation aşamalarının en riskli adımıdır çünkü doğrudan kurumun gizlilik, bütünlük ve yasal yükümlülükleri üzerinde etki yaratır. Bu fazda saldırgan; hassas dokümanları, veritabanı dump'larını, kimlik bilgilerini, finansal verileri veya özel anahtarları farklı kanallar üzerinden dışarı aktarır.

Data Exfiltration Teknikleri

- **HTTP / HTTPS üzerinden gizli yük (covert channels)**
  Normal web trafiği içine küçük paketler gömülerek veri dışarıya aktarılır; tespiti oldukça zordur.
- **DNS Tunneling**
  Veri, DNS sorgularının içine encode edilerek dış bir DNS sunucusuna gönderilir. Güvenlik cihazları çoğu zaman bunu "normal trafik" olarak algılar.
- **SMB / SFTP / FTP Transferleri**
  Direkt dosya aktarımı yapılarak bilgi sızdırılır. Ağda outbound port'lar açıksa en hızlı yöntemlerden biridir.
- **Email Exfiltration**
  Kurum içi e-posta hesabını kullanarak dışarıya dosya veya metin gönderilir. Çok sık kullanılan pratik bir yöntemdir.
- **Archive + Encrypt + Split yöntemleri**
  Büyük veritabanı dump'ları önce ZIP/7z ile şifrelenip bölümlere ayrılır; ardından parça parça farklı kanallarla aktarılır.

POST-EXPLOITATION

5. Defense Evasion (Tespit Önleme / İz Saklama)

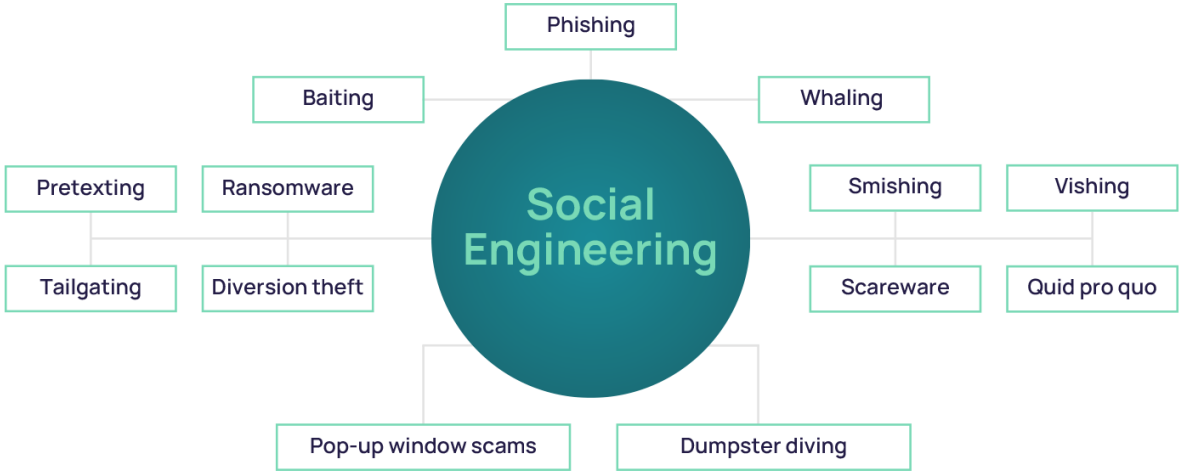
- Defense Evasion, saldırganın sistemdeki faaliyetlerini **gizlemek**, güvenlik çözümlerini atlatmak ve içeride daha uzun süre fark edilmeden kalmak için uyguladığı yöntemleri kapsar. Bu aşama, özellikle APT gruplarının operasyonlarında kritik bir yer tutar ve Post-Exploitation'ın gizlilik boyutunu temsil eder. Amaç; saldırganın davranışlarının loglara, güvenlik sensörlerine ve analiz araçlarına "normal kullanıcı aktivitesi" gibi görünmesini sağlamaktır.

Defense Evasion Teknikleri

- **Log Manipülasyonu / Log Silme**
  Windows Event Log'ların veya Linux syslog kayıtlarının temizlenmesi, belirli olay türlerinin gizlenmesi.
- **AV/EDR Bypass**
  Antivirüs ve EDR ürünlerinin imza tabanlı veya davranış tabanlı algılamalarını atlatmak için obfuscation, DLL sideloading, LOLBin (Living-Off-The-Land Binaries) kullanımı.
- **Obfuscated Scripts & Encrypted Payloads**
  PowerShell, JavaScript veya Python scriptlerinin gizlenmiş/şifrelenmiş versiyonlarının çalıştırılması.
- **Living off the Land (LOLbins)**
  powershell.exe, wmic.exe, certutil.exe, rundll32.exe gibi sistemde zaten bulunan araçlarla saldırı adımlarını gerçekleştirme; böylece dışarıdan zararlı binary gerektirmeden ilerleme.
- **Process Injection & Masquerading**
  Zararlı kodun meşru bir sürecin içine enjekte edilmesi veya bir uygulamanın meşru bir sistem süreci taklidi yapması.

SOSYAL MÜHENDİSLİK (SOCIAL ENGINEERING)

- Sosyal Mühendislik, teknolojik zafiyetlerden değil, **insan hatalarından ve davranış zaafiyetlerinden** yararlanan saldırı yöntemlerinin genel adıdır. Amaç; hedef kişinin güvenini kazanarak veya psikolojik manipülasyon uygulayarak **bilgi toplamak**, **erişim elde etmek**, **kurum içi süreçleri atlatmak** veya **kullanıcının kendisine zarar verecek bir eylemi yapmasını sağlatmaktır**.
- Bu tür saldırılar çoğu zaman firewall, antivirüs, IDS/IPS gibi teknik savunmaları kolayca atlatır çünkü "teknolojiye değil doğrudan insana saldırır". Gerçek dünyadaki veri sızıntılarının büyük bir kısmı teknik açıklar değil, sosyal mühendislik hataları nedeniyle gerçekleşir.



SOSYAL MÜHENDİSLİK AŞAMALARI

- Sosyal Mühendislik saldırıları, rastgele yapılan girişimler değildir; belirli bir planlama ve psikolojik manipülasyon süreci içerir. Bu aşamalar, saldırganın hedefi tanımasından nihai bilgi toplamaya kadar olan tüm adımları kapsar.

1) Hedef Profil Çıkarma (Profiling / Recon)
Hedef kişi, departman veya şirket hakkında bilgi toplanır.
Sosyal medya hesapları
E-posta formatları
IT yapısı
Çalışma alışkanlıkları
Amaç: Hedefin davranışlarını ve güven duyabileceği senaryoları anlamak.

2) Yaklaşım & Senaryo Tasarımı (Pretexting)
Saldırgan, hedefi kandırmak için güvenilir bir kimlik veya hikâye oluşturur.
Örnek: IT destek çalışanı, banka görevlisi, kargo şirketi, insan kaynakları.
Amaç: Hedefin mantığına uygun, şüphe çekmeyen bir rol yaratmak.

3) İletişim & Manipülasyon (Engagement)
Hedefle ilk temas kurulur (e-posta, telefon, fiziksel görüşme).
Sorularla yönlendirme
Acele durumu yaratma
Otorite taklidi
Yardım isteği veya panik durumu oluşturma
Amaç: Hedefin davranışını değiştirmek ve güven barajını düşürmek.

4) Sonuç Alma (Exploitation)
Hedef istenen eylemi gerçekleştirir:
Parola vermek
Zararlı dosya açmak
Linke tıklamak
İçeri giriş izni sağlamak
Gizli bilgi paylaşmak

SOSYAL MÜHENDİSLİK TÜRLERİ

- Sosyal mühendislik saldırıları birden fazla kanaldan yapılabilir. Her yöntemin hedefi farklı olsa da hepsi insan zaaflarını kullanır.

1) Phishing (E-posta Oltalama)

- Kullanıcıya sahte e-posta gönderilerek parola çalmak, zararlı dosya açtırmak veya oturum bilgisi yakalamak.
- Sahte banka mesajları
- Kurumsal şifre yenileme e-postaları
- Fatura/IRS/kargo bildirimleri

2) Spear-Phishing

- Belirli bir kişiye veya ekibe özel hazırlanmış phishing.
  Örnek: IT birimindeki bir personele departmana özel teknik dosya gönderme.

3) Whaling

- C-level yöneticilere yapılan hedefli sosyal mühendislik.
  Genelde finansal talimat sahteciliği, gizli belge isteği veya sahte CEO e-postaları içerir.

4) Vishing (Voice Phishing / Telefon Dolandırıcılığı)

- Telefon üzerinden kimlik doğrulama bilgisi alma, OTP kodu çaldırma, “IT destek” gibi davranma.

5) Smishing (SMS Phishing)

- SMS üzerinden zararlı link gönderilerek oturum çalma veya bankacılık bilgilerini ele geçirme.

6) Reverse Social Engineering

- Saldırgan kendisini mağdur gibi gösterip hedefin *kendi isteğiyle* iletişim kurmasını sağlar.

SIZMA TESTİ RAPORU (PENTEST REPORT)

- Raporlama, bir sızma testinin en kritik çıktısıdır. Teknik ekipler saldırıyı nasıl yaptığını bilse de müşteri bu sonucu ancak rapor ile anlayabilir. Profesyonel pentest şirketlerinin en çok önem verdiği aşama burasıdır.
- Bir pentest raporu şu amaçları taşır:
- Kuruma **hangi zafiyetlerin bulunduğunu** göstermek
- Bu zafiyetlerin **iş etkisini çapraz bir şekilde ölçmek**
- **Kanıtlar (PoC)** ile riskin gerçek olduğunu ispatlamak
- **Düzeltme önerileri** sunmak
- Zafiyetlerin **ciddiyet seviyesini derecelendirmek**
- Teknik olmayan yöneticilere bile anlaşılır sonuçlar sunmak
- Bu yüzden rapor, hem teknik ekip hem de yönetim tarafından anlaşılabilir şekilde hazırlanır.

PENTEST RAPORUNUN YAPISI

- Profesyonel pentest raporları genelde aşağıdaki bölümlerden oluşur:

1) Executive Summary (Yönetici Özeti)

- Teknik olmayan yöneticilere yönelik kısa özet.

2) Scope & Methodology (Kapsam ve Metot)

- Test edilen varlıklar, test dışı alanlar, kullanılan yöntemler ve standartlar.

3) Findings (Zafiyetler Bölümü)

- Her zafiyet ayrı başlık, açıklama, PoC, risk etkisi ve çözüm önerisi ile verilir.

4) General Assessment (Genel Değerlendirme)

- Kurumun güvenlik olgunluğu değerlendirilir.

5) Appendices (Ekler)

- Tarama çıktıları, ağ şeması, komut logları, CVSS hesaplamaları.

BİR BULGUNUN FORMATI

- Her bulgu profesyonel biçimde aşağıdaki formatla yazılır:
  - **Bulgu Adı** (ör. SQL Injection, Weak Password Policy)
  - **Risk Seviyesi** (Kritik / Yüksek / Orta / Düşük)
  - **CVSS Skoru**
  - **Açıklama:** Zafiyet neden oluşuyor?
  - **Kanıtlar (PoC):**
    - Ekran görüntüleri
    - Komut çıktıları
    - Payload örnekleri
  - **Risk Etkisi:**
    - Gizlilik, bütünlük, erişilebilirlik üzerindeki zarar
  - **Çözüm Önerileri:**
    Teknik, uygulanabilir, net adımlar
  - **Referanslar:**
    MITRE, OWASP, NIST bağlantıları
- Bu format uluslararası pentest standartlarında kullanılır.

SKORLAMA: CVSS 3.1

- CVSS'in en önemli bölümü **Base Metrics** kısmıdır; çünkü bir zafiyetin teknik ciddiyeti büyük ölçüde bu metriklere göre hesaplanır. Base Metrics, zafiyetin sömürülmesi için gereken koşulları ve istismar sonucunda sistemde oluşacak etkiyi tanımlar. Kurum ortamından bağımsızdır ve tamamen zafiyetin "ham teknik doğasını" ölçer.
- **CVSS Base Metrics şu alt başlıklardan oluşur:**

- **1) Attack Vector (AV)**
  Zafiyetin istismar edilmesi için saldırganın hedefe ne kadar yakın olması gerektiğini belirtir.
  Network (N): İnternet üzerinden sömürülebilir → en tehlikeli
  Adjacent (A): Aynı LAN/VLAN gerekir
  Local (L): Makineye yerel erişim gerekir
  Physical (P): Fiziksel temas gerekir

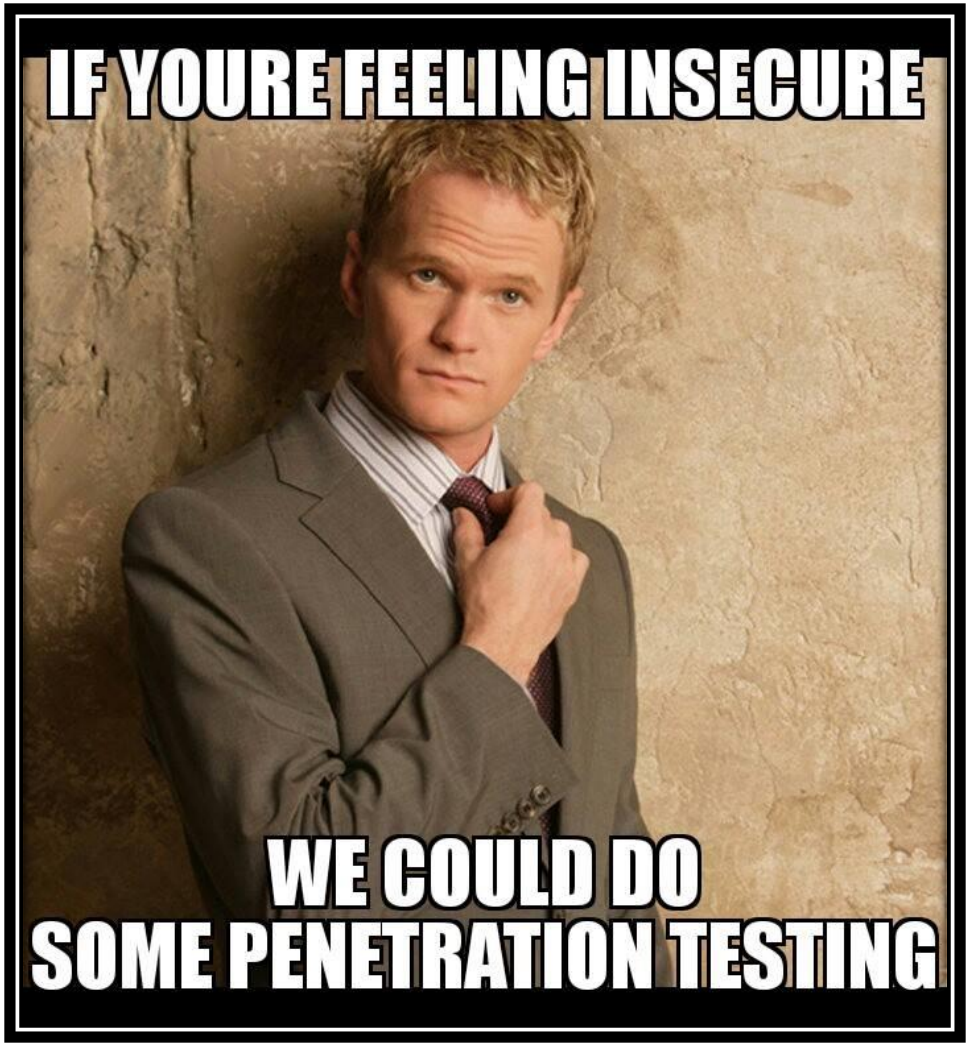
2) Attack Complexity (AC)
İstismar için özel koşullar gerekip gerekmediğini ölçer.
Low (L): Ek şarta ihtiyaç yok, doğrudan çalışır
High (H): Timing, özel konfigürasyon veya ön hazırlık gerekir

3) Privileges Required (PR)
Zafiyeti kullanmak için saldırganın hangi başlangıç yetkisine sahip olması gerektiğini ifade eder.
None (N): Hiçbir yetki gerekmez → en riskli
Low (L): Standart kullanıcı
High (H): Yönetici / root yetkisi

4) User Interaction (UI)
Zafiyetin tetiklenmesi için kullanıcı eylemi gerekip gerekmediğini belirtir.
None (N): Kullanıcı etkileşimi gerekmez (RCE vb.)
Required (R): Kullanıcı tıklar/dosya açar

5) CIA Impact Metrics (Confidentiality, Integrity, Availability)
Zafiyetin etkisini üç temel güvenlik bileşeni üzerinden ölçer:
Confidentiality (Gizlilik): Veriye yetkisiz erişim
Integrity (Bütünlük): Verinin değiştirilmesi
Availability (Erişilebilirlik): Servisin çökmesi/engellenmesi
Her biri **None (N) – Low (L) – High (H)** olarak değerlendirilir.

DİNLEDİĞİNİZ İÇİN TEŞEKKÜRLER



KAYNAKÇA

1. Wikipedia (2025). *Sızma testi*. Wikimedia Foundation. Erişim tarihi 17 Kasım 2025, https://tr.wikipedia.org/wiki/S%C4%B1zma_testi
2. PTES – Penetration Testing Execution Standard PTES. (2014). *Penetration Testing Execution Standard*. https://www.pentest-standard.org
3. NIST SP 800-115 – Technical Guide to Information Security TestingScarfone, K., Souppaya, M., & Cody, A. (2008). *Technical Guide to Information Security Testing and Assessment (NIST SP 800-115).* https://csrc.nist.gov/publications/detail/sp/800-115/final
4. OWASP Web Security Testing GuideOWASP Foundation. (2023). *OWASP Web Security Testing Guide*. https://owasp.org/www-project-web-security-testing-guide/
5. Common Vulnerability Scoring System (CVSS 3.1)FIRST. (2019). *Common Vulnerability Scoring System v3.1: Specification Document.* https://www.first.org/cvss/v3.1/specification-document
6. Alhassan, M., & Abomhara, M. (2020). *Penetration testing: A systematic literature review*. IEEE Access, 8, 165130–165144. https://doi.org/10.1109/ACCESS.2020.3022554
7. Sharma, R., Sahay, R., & Kumar, R. (2021). *A comprehensive survey on penetration testing methodologies and tools*. Journal of Information Security and Applications, 58, 102727. https://doi.org/10.1016/j.jisa.2021.102727