



# Beyaz Net

## Penetrasyon Testi Sonuç Raporu



Bilişim partneriniz

Beyaz Bilgisayar ve Danışmanlık Hizmetleri LTD.ŞTİ.  
Tantavi Mahallesi İpekyolu Sokak No: 1  
Ümraniye / İSTANBUL  
T: 0216 557 72 72 – F: 0216 422 22 90  
01/06/2022 - 24/06/2022

Bu belge “**GİZLİ**” bilgiler içermektedir. Yetkili kişiler dışında okunması yasaktır.  
Bu belge yetkisiz bir şekilde elinize ulaştıysa lütfen [beyaz@beyaz.net](mailto:beyaz@beyaz.net) adresine bildiriniz

## Rapor Detayları

Rapor Başlığı	Beyaz Net Penetrasyon Testi Sonuç Raporu
Versiyon	v1.0
Yazan	Berkan AKIN
Test Ekibi	Berkan AKIN
Kontrol Eden	Eren ÇİFTCİ
Onaylayan	Mehmet Fatih ZEYVELİ (Genel Müdür)
Rapor Sınıfı	Gizli

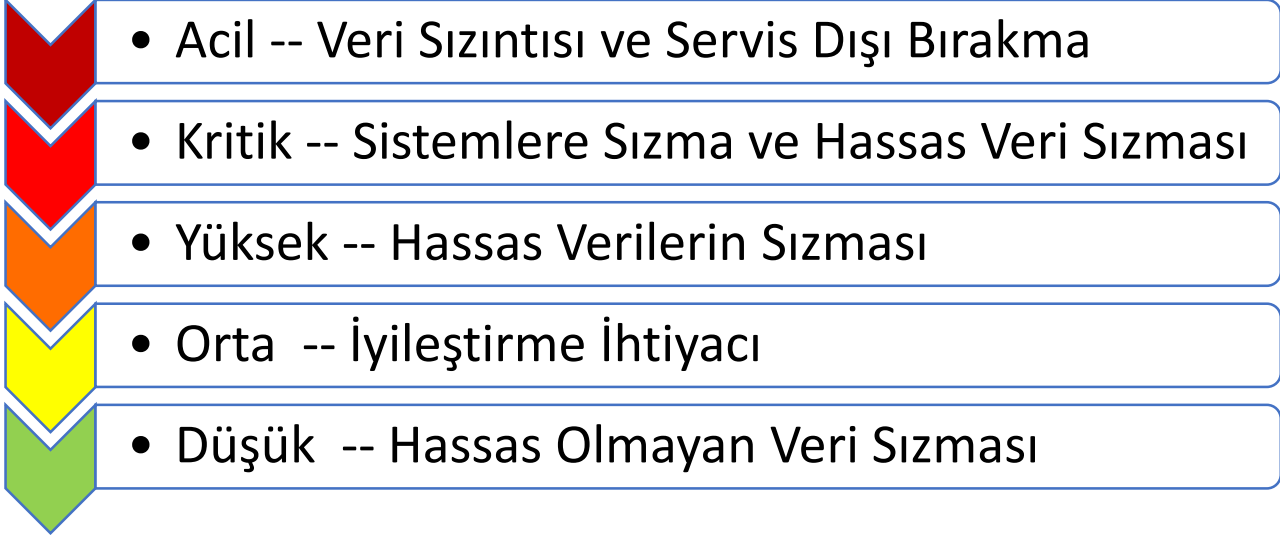
## Kurum Yetkilisi

Yetkili Adı ve Soyadı	Ünvanı	Kurum Adı
Beyaz Net	Beyaz Net	Beyaz Net

## Rapor Denetimi

Version	Tarih	Yazar	Tanım
v1.0	2022-05-04	Eren ÇİFTCİ	Final

## Güvenlik Seviyesi Derecelendirmesi



## Kurum Değerlendirmesi

**Yüksek**

- Hedef kuruma ait sistemler üzerinde yapılan çalışmalar neticesinde genel olarak Kritik, Yüksek ve Orta seviyeli bulgular tespit edilmiştir.
- Kurumun genel güvenlik seviyesi “ Yüksek Seviyede Riskli ” olarak belirlenmiştir.

## Yasal Sorumluluklar

Söz konusu raporun içeriği gizli olup, taraflar arasında yazılı mutabakat olmadan üçüncü kişilere basılı olarak (hardcopy) ya da elektronik ortamda (softcopy) paylaşılamaz, yayınlanamaz ve çoğaltılamaz.

Her ne kadar içeriğin doğruluğu ve hassasiyeti için maksimum çaba gösterildiyse de rapora ait muhtemel eksikliklerden dolayı Beyaz.Net sorumlu tutulamaz.

Rapora konu olan zafiyet analizi testleri Internet üzerinden gerçekleştirildiği durumlar için, üçüncü kişi ya da tüzel kişilerin (ISP'ler vs.) söz konusu trafiği dinleme ihtimali bulunmaktadır. Bu nedenle test aktivitelerinin üçüncü kişiler tarafından dinlenmesi veya gözlenmesi sonucunda ileride ortaya çıkabilecek kayıplardan Beyaz.Net sorumlu tutulamaz.

İşbu rapor içeriğinde yer alan test sonuçları firma veya kullanılan uygulamalar hakkında kritik bilgiler içerebilir. Dolayısıyla raporun dağıtılması konusunda gerekli hassasiyetin firma tarafında da gösterilmesi gerekmektedir. Beyaz.Net'ten kaynaklanmayan, raporun kaybolması veya üçüncü kişi /kişiler tarafından ele geçirilmesi sebebi ile kritik bilgilerin elde edilmesi durumunda meydana gelecek kayıplardan Beyaz.Net sorumlu tutulamaz.

İşbu rapor, tarama süresi içinde varlığı bilinen veya tarafımızdan tespit edilen güvenlik açıklarını içermektedir. Tarama işlemi bittikten sonra rapor teslim edilene kadar geçen süre içerisinde çıkabilecek yeni güvenlik açıklarına dair eksikliklerden dolayı Beyaz.Net sorumlu tutulamaz.

Raporlar içerisinde yer alan çözüm önerileri sadece tavsiye niteliği taşımaktadır. İlgili önerilerin tatbik edilmesi sebebi ile çıkabilecek problemlerden Beyaz.Net sorumlu tutulamaz. Öneriler başlığı altında sunulan değişiklikler, gerçekleştirilmeden önce mutlaka ilgili konu hakkında destek alınan yetkili firmalara veya uzman danışmanlara konu hakkında görüşleri sorulmalıdır.

İşbu raporu okuyan yetkili/yönetici vb. kişiler yukarıdaki maddeleri ihlal etmeyeceğini kabul ve taahhüt eder.

# İçindekiler

1	GİRİŞ .....	5
2	KAPSAM .....	6
3	YÖNETİCİ ÖZETİ .....	9
4	GENEL SIZMA TESTİ METODOLOJİSİ .....	11
4.1	BİLGİ TOPLAMA .....	12
4.2	AĞ HARİTALAMA .....	12
4.3	ZAFİYET TESPİTİ .....	12
4.4	SIZMA (PENETRASYON) SÜRECİ .....	13
4.5	HAK YÜKSELTME .....	13
4.6	ERİŞİMİN KORUNMASI .....	13
4.7	İZLERİN SİLİNMESİ .....	13
5	GERÇEKLEŞTİRİLEN GÜVENLİK TESTLERİ VE SONUÇLARI .....	14
6	SOSYAL MÜHENDİSLİK TESTİ .....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
6.1	GERÇEKLEŞTİRİLEN GÜVENLİK TESTİ İŞLEMLERİ .....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
6.1.1	Tespit Edilen Zafiyetler .....	Hata! Yer işareti tanımlanmamış.
6.1.1.1	Social Engineering .....	Hata! Yer işareti tanımlanmamış.
7	WEB UYGULAMA GÜVENLİK TESTLERİ .....	15
7.1	GERÇEKLEŞTİRİLEN GÜVENLİK TESTİ İŞLEMLERİ .....	15
7.1.1	Tespit Edilen Zafiyetler .....	17
7.1.1.1	Command Injection .....	17
7.1.1.2	Cross Site Scripting (XSS) .....	Hata! Yer işareti tanımlanmamış.
7.1.1.3	Broken Authentication Attack .....	Hata! Yer işareti tanımlanmamış.
7.1.1.4	Directory Listings .....	Hata! Yer işareti tanımlanmamış.
13	SIZMA TESTİNDE KULLANILAN ARAÇLAR & YAZILIMLAR .....	64
14	TERİMLER SÖZLÜĞÜ .....	65

## 1 Giriş

Bu rapor, Beyaz Bilgisayar ve Danışmanlık Hizmetleri Ltd. Şti tarafından “**Beyaz Net**” sistemleri üzerindeki güvenlik açıklarını ortaya çıkartmak amacı ile **01/06/2022 - 24/06/2022** tarihleri arasında gerçekleştirilen güvenlik ve sızma testlerinin (penetration test) detaylı sonuçlarını içermektedir.

Pentest çalışması kapsamında “**Beyaz Net**” altyapısı ve sunucularının çalışmasını olumsuz yönde etkileyecek araçlar ve yöntemler kullanılmamış, izinsiz ve yetkisiz bir şekilde hizmetin aksamasına neden olabilecek herhangi bir işlem gerçekleştirilmemiştir.

Rapor kapsam, yönetici özeti, öneriler ve kategorik olarak tespit edilen güvenlik zafiyetlerine ait detayları ve referansları içermektedir.

## 2 Kapsam

Sızma testinde ana amaçlardan biri tüm zafiyetlerin değerlendirilerek sisteme sızılmaya çalışılmasıdır. Bu amaç doğrultusunda gerçekleştirilecek sızma testlerinde kapsam pentest çalışmasının en önemli adımını oluşturmaktadır.

Sistem/ağ yöneticileri ile hackerların bakış açısı farklıdır ve sistem/ağ yöneticisi tarafından riskli görülmeyen bir sunucu/sistem, hacker için sisteme sızmanın ilk adımı olabilir.

Gerçekleştirilen denetimlerde “**Beyaz Net**” yetkilileri tarafından bildirilen ve Tablo 'da verilen sistemlere yönelik sızma testleri gerçekleştirilmiştir.

Test Başlığı	Detaylar
Web Uygulamaları	.../bwapp/...

## Kategorilerine Göre Zafiyet Listesi

phishing	Social Engineering
web	Command Injection
web	Cross Site Scripting (XSS)
web	Path Traversal
web	Broken Authentication Attack
web	Directory Listings
web	Open Redirect
web	Authorization Attack
network	MS17-010 Remote Command Execution
network	Acme tthttpd < 2.26 Multiple Vulnerabilities
switch	ARP Cache Spoofing/Poisoning Attack
switch	SNMP Agent Default Community Name
database	Microsoft SQL Server Unsupported Version Detection
mobil	No Root Detection
mobil	Application Permissions

## Saldırı Noktasına Göre Zafiyet Listesi

External Web Application	Command Injection
External Web Application	Cross Site Scripting (XSS)
External Web Application	Path Traversal
External Web Application	Broken Authentication Attack
External Web Application	Directory Listings
External Web Application	Open Redirect
External Web Application	Authorization Attack
Internal Network	Wireless Network Testing
Internal Network	MS17-010 Remote Command Execution
Internal Network	ARP Cache Spoofing/Poisoning Attack
Internal Network	Acme tthttpd < 2.26 Multiple Vulnerabilities
Internal Network	SNMP Agent Default Community Name
Internal Database	Microsoft SQL Server Unsupported Version Detection
Mobile Application	No Root Detection
Mobile Application	Application Permissions



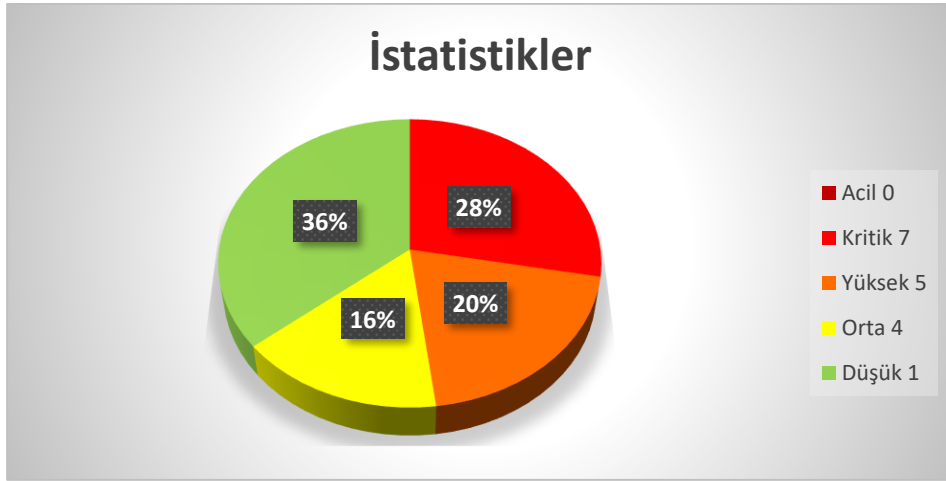
### 3 Yönetici Özeti

Bu rapor, Beyaz Bilgisayar ve Danışmanlık Hizmetleri Ltd. Şti. tarafından “Beyaz Net” bilişim sistemleri üzerindeki güvenlik açıklarını ortaya çıkartmak amacı ile **01/06/2022 - 24/06/2022** tarihleri arasında gerçekleştirilen sızma testleri (penetration test) ve güvenlik testleri çalışmalarının sonuçlarını içermektedir.

Testler, raporun devamında detayları verilen web uygulama, etki alanı/sunucu-istemci sistemleri, e-posta sunucuları, veritabanı testleri kapsamında gerçekleştirilmiştir.

Gerçekleştirilen testler esnasında bulunan zafiyet kategorilerinin risk dağılımları aşağıdaki gibidir.

Acil	Kritik	Yüksek	Orta	Düşük	Toplam
0	25	0	0	0	25



*Bir açıklığın birden fazla sistemde bulunması zafiyet sayısını etkilememektedir.*

Testlerimiz sırasında kullanılan başlıca yöntemler aşağıdaki gibidir;

- ❖ SQL Injection testleri
- ❖ Siteler arası istek sahtekarlığı
- ❖ Bellek taşması testleri
- ❖ Yerel ve uzaktan dosya okuma
- ❖ Basit parola denemeleri
- ❖ Güncelleştirme eksikliklerinden kaynaklanan kritik güvenlik testleri
- ❖ Yapılandırma eksikliği testleri
- ❖ Uygulama geliştirmedeki eksiklik taraması
- ❖ Ön tanımlı kullanıcı hesapları ile erişim denemeleri
- ❖ Yatayda ve dikeyde yetki erişim testleri
- ❖ Doğrudan nesne referansı
- ❖ Yansıtılan siteler arası script çalıştırma
- ❖ Tahmin edilebilir/ Ön tanımlı hesap bilgisi kullanımı testleri
- ❖ Hatalı oturum yönetimi ve captcha testleri

Beyaz.Net sızma testi kapsamında teknik raporda ayrıntıları (ekran görüntüleri, açıklama, çözüm önerisi, referans linkleri) verilen zafiyetler kullanılarak; web uygulaması testinde Host Header Injection zafiyeti kullanılarak saldırgan,web uygulamasını ziyaret eden kullanıcıları sahte bir web adresine yönlendirerek sosyal mühendislik saldırıları yapabilir.Web uygulamasında parametrelere özel karakter girilerek MySQL veritabanı hatası gözlemlenmiştir.Potansiyel olarak SQL Injection saldırısı uygulanabilir.Web uygulamasında Host Header Injection ve SQL Injection'a sebep olabilecek header ve parametreler filtrelenmelidir.Hata sayfasında bilgi ifşasına neden olabilecek bilgiler barınmaktadır bu yüzden özel hata sayfası oluşturulmalıdır.Web uygulama testinde bulunan diğer zafiyetler versiyon güncellemesi ile giderilebilir.Yerel ağ testinde; MS17-010 zafiyeti barındıran sunucuda uzaktan komut çalıştırılmıştır. Sunucuya yeni kullanıcı eklenip yerel yönetici grubuna eklenebilmiştir ancak oluşturulan kullanıcı ile giriş yapılamamıştır.PrintNightmare uzaktan komut çalıştırma zafiyetine sebep olan 'PrintSpooler'servisi kapatılmalıdır.SMB (dosya paylaşım protokolü) testinde genel olarak yapılandırma doğru konfigüre edilmiştir.Desteklenmeyen Windows sürümleri son versiyona güncellenmelidir

Testler sonucu en büyük güvenlik eksikliği, çalışan sistemlerin güvenlik standartlarına ve prosedürlerine uygun olarak kurulmaması ve kurulumdan sonra gereken güvenlik sıkılaştırmalarının yapılmaması veya eksik yapılmasından kaynaklandığı belirlenmiştir. Bu sebeple her bir işletim sistemi, ağ cihazı ve diğer cihazlar için bir kurulum prosedürünün hazırlanması, bütün kurulumların yazılı prosedürlere uygun olarak yapılması ve ürün ortamına alınmadan önce mutlaka güvenlik taramasından geçirilmesi önerilmektedir.

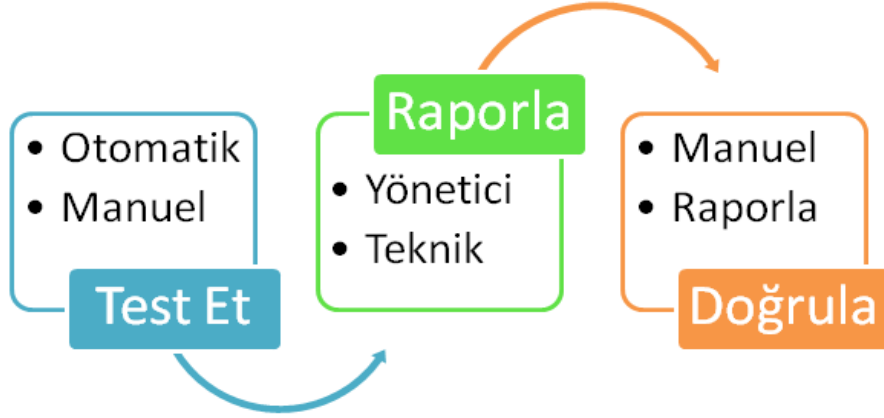
**Raporda her bir açıklığın hangi sistemlerde bulunduğu, zafiyetler ile ilgili alınması gereken önlemler detaylı olarak açıklanmıştır.** Testlerdeki güvenlik zafiyetlerinin kapatılması için gerekli çalışmalar yapılmalıdır. Zafiyetlerin kapatılmasında izlenecek sırayı belirlerken teknik raporda belirtilen zafiyet önem dereceleri öncelikli rol oynamalıdır. Düşük seviyeli zafiyetler raporda belirtilmemiştir.

## 4 Genel Sızma Testi Metodolojisi

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.

Pentest(sızma testleri) ve vulnerability assessment(zayıflık tarama) birbirine benzeyen fakat farklı kavramlardır. Zayıflık tarama, hedef sistemdeki güvenlik zafiyetlerinin çeşitli yazılımlar kullanarak bulunması ve raporlanması işlemidir. Pentest çalışmalarında amaç sadece güvenlik zafiyetlerini belirlemek değil, bu zafiyetler kullanılarak hedef sistemler üzerinde gerçekleştirilebilecek ek işlemlerin (sisteme sızma, veritabanı bilgilerine erişme) belirlenmesidir.

Zayıflık tarama daha çok otomatize araçlar kullanılarak gerçekleştirilir ve kısa sürer. Pentest çalışmaları, zayıflık tarama adımını da kapsayan ileri seviye tecrübe gerektiren bir süreçtir ve zayıflık tarama çalışmalarına göre çok daha uzun sürer.

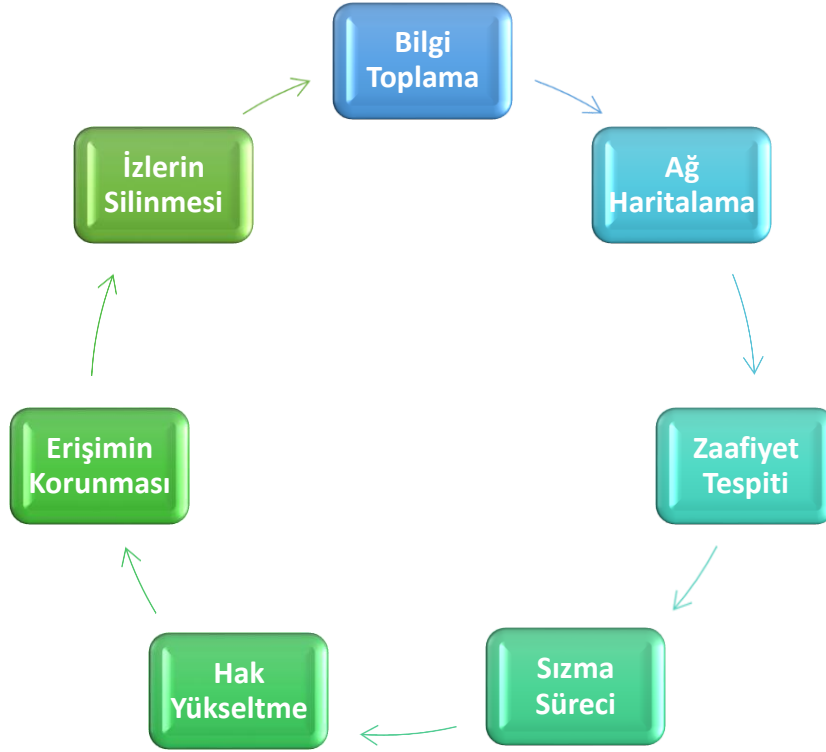


**Beyaz.Net** “Security Assessment Framework” hazırlanırken konu hakkındaki uluslararası standartlar incelenmiş ve azami ölçüde faydalanılmıştır. Aşağıda bu belgenin hazırlanmasında kaynak olarak kullanılan dokümanların isimleri yer almaktadır.

- ❖ OWASP Testing Guide v3
- ❖ OSSTM
- ❖ ISSAF
- ❖ NIST
- ❖ PTES
- ❖ Fedramp

Gerçekleştirilen testler uluslararası standart ve yönetmeliklere (PaymentCardIndustry (PCI), ISO 27001) tam uyumludur.

Beyaz Bilgisayar Danışmanlık Hizmetleri tarafında gerçekleştirilecek olan penetrasyon testinin adımları aşağıdaki gibidir.



Şekil 1: Penetrasyon Testi Test Aşamaları

#### 4.1 Bilgi Toplama

Bilgi toplama ilk ve en önemli adımdır. Bilgi toplama aşamasında hedef sistem hakkında ne kadar çok bilgi elde edilebilirse sızma testi aşaması o kadar kapsamlı ve kesin sonuç elde edilebilir. Bu sebeple bu aşamada ki amaç hedef sistem üzerinde mümkün olduğunca fazla bilgi toplamaktır. Bu bilgiler firma hakkında olabileceği gibi firma çalışanları hakkında da olabilir.

#### 4.2 Ağ Haritalama

Ağ haritalama aşamasında amaç ağ yapısını detaylıca belirlemektir. Açık sistemler ve açık portları, servisler ve servis üzerinde çalışan yazılımlar ve bu yazılımların sürümleri, sistemdeki Firewall, IPS cihazlarının belirlenmesi, sunucu sistemler ve versiyonları gibi bileşenlerin tamamı belirlendikten sonra hedef sisteme ait ağ haritası çıkartılır. Ağ haritalama esnasında hedef sistemde IPS, WAF ve benzeri savunma sistemleri olup olmadığı tespit edilmeli ve teste buna göre devam edilmelidir.

#### 4.3 Zafiyet Tespiti

Hedef tespiti aşamasında amaç belirlenen hedef sistemdeki açıklıkları ortaya çıkarmaktır. Kapsam dahilinde belirtilen IP aralığındaki canlı sistem tespiti, işletim sistemi versiyonları, service pack sürümleri gibi bilgiler elde edilmeye çalışılır. Birden fazla zafiyet tarama aracıyla sistem taranarak oluşabilecek false/positif oranı minimize edilmeye çalışılır. Bu aşamada tarama, hedef sisteme zarar vermeden gerçekleştirilir. Sonuçlar uzmanlar tarafından yorumlanarak rapora eklenir.

#### 4.4 Sızma (Penetrasyon) Süreci

Sızma sürecinde, daha öncesinde belirlenen açıklıklar, bu açığa ait exploit veritabanında exploit yazılmışsa bulunan bu exploit ile, eğer yazılmamışsa ve yeteri kadar süre tanınmışsa o açığa ait sıfırdan bir exploit yazılarak mevcut açıklara ait kritik veriler elde edilmeye çalışılır. Kritik süreçlerde öncelikle firma onayı alınır ve Exploit edilen sistemlerin çalışmasını kesintiye uğratmamak için büyük özen gösterilir, servis dışı bırakmaya yönelik exploitler ve yasa dışı exploitler kullanılmaz.

Bu adımda dikkat edilecek husus, denenecek exploitlerin onayının alınması ve daha önceden lab ortamında denenmiş bir exploit kullanılmalıdır.

#### 4.5 Hak Yükseltme

Hak yükseltme sürecindeki amaç, exploit edilen herhangi bir sistem hesabı ile tam yetkili kullanıcı moduna geçmektir. Bu aşama için mevcut yerel hak yükseltme exploitleri kullanılabilir. Bir sonraki sürece katkı sağlaması açısından bu süreçte, elde edilen yetkili makinelerde rhost,ssh dosyaları veya history gibi komutlarla kapsamlı bilgiler elde edilmeye çalışılır. (Örneğin ilgili sistem üzerinde root haklarına sahip olursa bile yerel kullanıcı veritabanının kopyasının alınması ve bu kullanıcıların şifrelerinin kırılarak kapsam dahilindeki diğer sunuculara erişim için kullanılması gibi.)

#### 4.6 Erişimin Korunması

Bu aşamada, sisteme girildiğinin başkaları tarafından farkedilmemesi için önlemler alınır. Bunlar giriş loglarının silinmesi, çalıştırılan ekprocesslerin saklı tutulması, dışarıya erişim açıksa gizli kanalların kullanılması, backdoor, rootkit yerleştirilmesi gibi.

#### 4.7 İzlerin Silinmesi

Bu aşamada, hedef sisteme bırakılan backdoorlar, test için yazılmış scriptler, sızma testi aşamasında eklenen tüm bilgiler not alınarak test bitiminde silinir. Sistemin penetrasyon testine başlamadan önceki haliyle bırakıldığına dair kurumdan onay alınır.

## 5 Gerçekleştirilen Güvenlik Testleri Ve Sonuçları

Sızma test sonuçlarının raporlanması temelde iki farklı şekilde yapılmaktadır. Bunlardan ilki bileşen bazlı raporlama, diğeri de hedef bazlı raporlama. Hedef bazlı raporlamada her bir zafiyet ayrı bir başlık olarak yazılmaktadır, bileşen bazlı raporlamada aynı kategorideki (kapatılması aynı aksiyona bağlı, aynı açıklığın farklı sistemlerde bulunması) zafiyetler tek bir başlık altında yazılarak bulgu içerisinde ayırım yapılmaktadır.

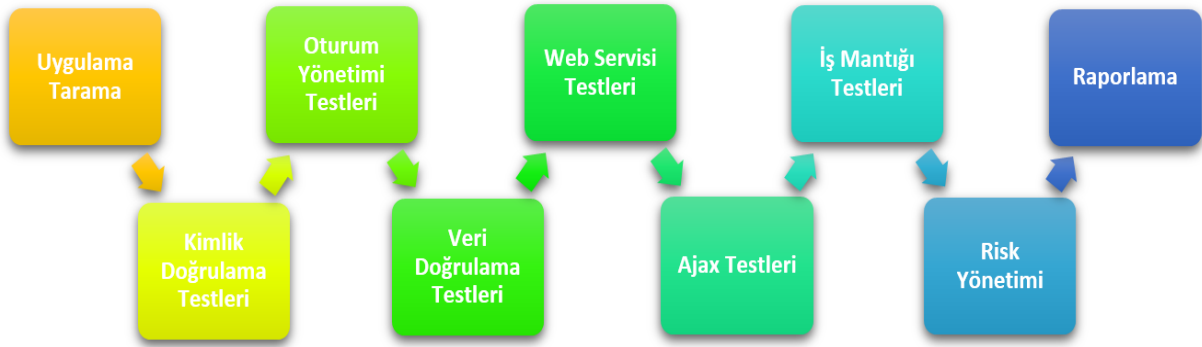
Raporun okunurluğu ve sadeliği açısından “**Beyaz Net**” için gerçekleştirilen sızma testi çalışmasında bileşen bazlı raporlama tercih edilmiştir. Aşağıda gerçekleştirilen testler ve testlere ait çıktılarına yer verilmiştir.

## 6 Web Uygulama Güvenlik Testleri

### 6.1 Gerçekleştirilen Güvenlik Testi İşlemleri

Gerçekleştirilen güvenlik testi işlemleri web uygulamalarına yapılan testler sisteme zarar vermeyecek şekilde, internet üzerinden ve yerel ağdan gerçekleştirilmiştir. Sunucular üzerinde çalışan servislerin ve işletim sisteminin bilinen zafiyetlerinin araştırılmasının yanında, sistemdeki uygulamalara has güvenlik zafiyetleri de araştırılmıştır.

Yapılan güvenlik testleri bileşen tabanlı ele alınmıştır. Bu testlerde ilk olarak Beyaz.Net tarafından derlenen Test Prosedürleri adımları uygulanmıştır. Test prosedürleri ile tespit edilemeyen zafiyetler ise ticari tarama araçları yardımıyla bulunmaya çalışılmıştır. Bu araçların birçok yanlış alarm (false positive) verebileceği hususu göz önünde bulundurularak, tespit edilen zafiyetler detaylı olarak incelenmiştir.



Bu kapsamda aşağıda detaylandırılan test adımları gerçekleştirilmiştir:

- ❖ Uzaktan genel tarama araçları ile sunucuların açık olan servisleri, yama eksiklikleri ve yapılandırma hataları aranmıştır.
- ❖ Uygulama girdisi kontrol testleri (Siteler Ötesi Betik Çalıştırma, Parametre Enjeksiyonu ve Manipülasyonu) uygulanmıştır.
- ❖ Parametre bütünlüğü güvenlik kontrolleri denetlenmiştir.
- ❖ Sistem hakkında bilgi açığa çıkarmaya yönelik testler uygulanmıştır.
- ❖ Oturum yönetiminde bulunabilecek bazı zafiyetler araştırılmıştır.
- ❖ Yetkilendirme (URL tabanlı) süreçlerinde bulunabilecek bazı zafiyetler araştırılmıştır.
- ❖ Uygulamanın bulunduğu sunucu üzerinde konuşlanmış diğer servisler kullanılarak bilgi edinilmeye çalışılmıştır.
- ❖ İlgili veri tabanlarına erişim sağlanmaya çalışılarak, uygulamada yetkili kullanıcı hesapları edinilmeye çalışılmıştır.
- ❖ Şifre politikaları incelenmiştir.
- ❖ Ayrıca aşağıda verilen test başlıkları da manuel olarak, açık kaynak ve ticari araçlar kullanılarak test edilmiştir.

#### Yapılandırma Yönetim Testleri

- ❖ SSL/TLS versiyon, algoritma ve sertifika geçerlilik testleri - OWASP-CM-001
- ❖ Hedef uygulamada kullanılan yönetim panelinin belirlenmesi - OWASP-CM-00
- ❖ Dosya uzantısı yönetimi testleri - OWASP-CM-005
- ❖ Yedek, kopya, test veya eski sürümlerden kalma sayfa ve uygulamaların belirlenmesi- OWASP-CM-00
- ❖ Sunucu tarafından desteklenen metodların ve XST belirlenmesi - OWASP-CM-008

#### Kimlik Doğrulama Testleri

- ❖ Hassas bilgilerin şifreli/şifresiz kanallardan aktarımı - OWASP-AT-001
- ❖ Hedef uygulama üzerinde kullanıcı adı belirleme/doğrulama çalışmaları - OWASP-AT-002
- ❖ Hedef uygulama üzerinde tanımlı kullanıcıların belirlenmesi - OWASP-AT-00
- ❖ Hedef uygulama üzerinde yetkili kullanıcılara yönelik brute force parola denemeleri - OWASP-AT-00
- ❖ Kimlik doğrulama aşamasını atlatma denemeleri - OWASP-AT-005
- ❖ Parola hatırlatma ve parola sıfırlama özelliklerinin testleri - OWASP-AT-006
- ❖ Browser ön bellek yönetimi ve “Log out” fonksiyonlarının testleri - OWASP-AT-007
- ❖ CAPTCHA güvenlik testleri - OWASP-AT-008

### **Oturum Yönetimi Testleri**

- ❖ Oturum yönetimi zayıflıkları, oturum yönetimi bypass testleri - OWASP-SM-001
- ❖ Detaylı cookie güvenlik testleri - OWASP-SM-002
- ❖ Oturum sabitleme (session fixation) testleri - OWASP-SM-003
- ❖ Oturum değerleri tahmin saldırıları - OWASP-SM-004
- ❖ CSRF(Cross site request forgery) testleri - OWASP-SM-005

### **Yetkilendirme Testleri**

- ❖ Dizin atlatma/gezme(Directory Treversal) testleri - OWASP-AZ-001
- ❖ Yetkilendirme atlatma, yetkilendirme geçiş testleri - OWASP-AZ-002
- ❖ Yetki yükseltimi testleri - OWASP-AZ-003

### **İş Mantığı Denetim Testleri**

- ❖ Uygulamanın işleyişinin belirlenmesini takiben uygulamanın işleyişine yönelik teknik olmayan atakların denenmesi.

### **Veri Doğrulama Testleri**

- ❖ Yansıtılan XSS testleri - OWASP-DV-001
- ❖ Depolanmış XSS testleri - OWASP-DV-002
- ❖ DOM tabanlı XSS testleri - OWASP-DV-003
- ❖ XSF (Flash XSS) testleri -OWASP-DV-004
- ❖ SQL enjeksiyonu testleri - OWASP-DV-005
- ❖ LDAP enjeksiyonu testleri - OWASP-DV-006
- ❖ Xpath enjeksiyonu testleri - OWASP-DV-010
- ❖ Kod enjeksiyonu testleri - OWASP-DV-012
- ❖ İşletim sistemi komut enjeksiyonu testleri - OWASP-DV-013
- ❖ Bellek taşması (buffer overflow) testleri - OWASP-DV-014
- ❖ Http response splitting testleri - OWASP-DV-016

### **Web Servisi ve Ajax Testleri**

- ❖ Web servisi bilgi toplama çalışmaları - OWASP-WS-001
- ❖ WSDL testleri - OWASP-WS-002
- ❖ XML yapı testleri - OWASP-WS-003

### **Web Uygulama Güvenlik Sistemlerinin Testleri**

- ❖ Web uygulama güvenlik duvarı keşif testleri
- ❖ Network IPS keşif testleri
- ❖ IPS/Web uygulama güvenlik duvarı atlatma testleri



### 6.1.1 Tespit Edilen Zafiyetler

Kritik	6.1.1.1 Command Injection
Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-
Bulgu Açıklaması	

#### HTML Injection Reflected(Get)

Kullanıcıdan alınan input verilerini sunucuya filitrelenmeden alınması ve tekrar kullanıcılara girilen inputların aynısının cevap olarak dönmesi kullanıcıların tarayıcısı üzerinde zararlı bir kod çalıştırılabilir hale getirmektedir

**Zafiyetin bulunduğu url:** .../ /bWAPP/htmli\_get.php

Get isteği ile belirtilen url üzerinden alınan veriler urlin parametrelerinin manipile edilerek html kodlarının çalışmasına imkan vermektedir.

**Zafiyet Kanıtı:**



Örnek Ekran Görüntüsü – HTML Injection Reflected(Get)

**Açıklığı Barındıran Sistemler:**

.../bWAPP/htmli\_get.php

**Çözüm Önerileri:**

- ❖ Input verileri filitrelenmeli ve HTML etiketlerini input olarak kabul edilmemelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/11-Client-side\\_Testing/03-Testing\\_for\\_HTML\\_Injection](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/03-Testing_for_HTML_Injection)

**Kritik**

## 6.1.1.2 Command Injection

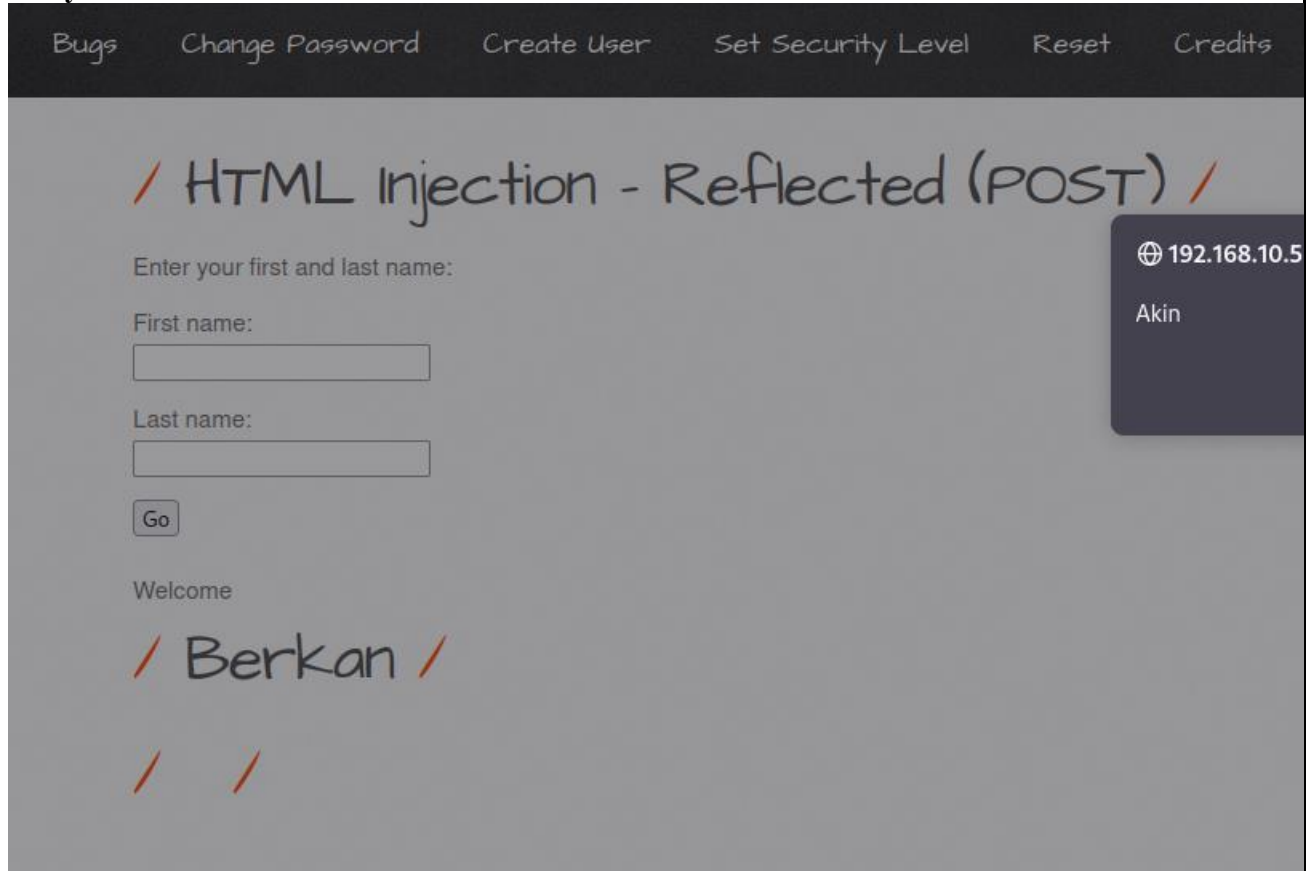
Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****HTML Injection – Reflected(Post)**

Kullanıcıdan alınan input verilerini sunucuya filitrelenmeden alınması ve tekrar kullanıcılara girilen inputların dönmesi ile kullanıcının tarayıcısı üzerinde zararlı bir kod çalıştırılabilir hale getirmektedir.

**Zafiyetin bulunduğu url:** .../bWAPP/htmli\_post.php

Post isteği ile alınan veriler araya proxy araçları ile girilerek manüple edilip html kodlarının çalışmasına imkan vermektedir.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – HTML Injection – Reflected(Post)

**Açıklığı Barındıran Sistemler:**

- ❖ .../bWAPP/htmli\_post.php

**Çözüm Önerileri:**

- ❖ HTML etiketlerini input olarak kabul edilmemelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/11-Client-side\\_Testing/03-Testing\\_for\\_HTML\\_Injection](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/03-Testing_for_HTML_Injection)

**Kritik**

## 6.1.1.3 Command Injection

Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****HTML Injection reflected(Url)**

Kullanıcıdan alınan input verilerini sunucuya filitrelenmeden alınması ve tekrar kullanıcılara girilen inputların dönmesi ile kullanıcının tarayıcısı üzerinde zararlı bir kod çalıştırılabilir hale getirmektedir.

**Zafiyetin bulunduğu url:** .../bWAPP/htmli\_post.php

Get isteği ile belirtilen url üzerinden alınan veriler urlin parametrelerinin manüple edilerek html kodlarının çalışmasına imkan vermektedir.



Örnek Ekran Görüntüsü – HTML Injection reflected(Url)

**Açıklığı Barındıran Sistemler:**

- ❖ .../ /bWAPP/htmli\_current\_url.php

**Çözüm Önerileri:**

- ❖ Tarayıcıdan gelen veriler filtreye tabi tutulup sonrasın da web sunucusunda işleme alınmalıdır.
- ❖ **Referanslar:**

- ❖ [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/11-Client-side\\_Testing/03-Testing\\_for\\_HTML\\_Injection](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/03-Testing_for_HTML_Injection)

**Kritik**

## 6.1.1.4 Command Injection

Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****HTML Injection - Stored (Blog)**

Kullanıcıdan alınan input verilerini sunucuya filitrelenmeden alınması ve tekrar kullanıcılara aynı input verilerinin dönmesi tarayıcı üzerinde zararlı bir kod çalıştırılabilir hale getirmektedir.

HTML verilerinin doğrudan sunucuya alınmasından ve sunucudan kullanıcıya iletilmesinden dolayı tarayıcılarda kod çalıştırabilme imkanı vermektedir.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – HTML Injection Stored (Blog)

**Açıklığı Barındıran Sistemler:**

❖ ... /bWAPP/htmli\_stored.php

**Çözüm Önerileri:**

- ❖ HTML etiketlerini input noktalarından içeriye alınmaması gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/11-Client-side\\_Testing/03-Testing\\_for\\_HTML\\_Injection](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/11-Client-side_Testing/03-Testing_for_HTML_Injection)



**Kritik**

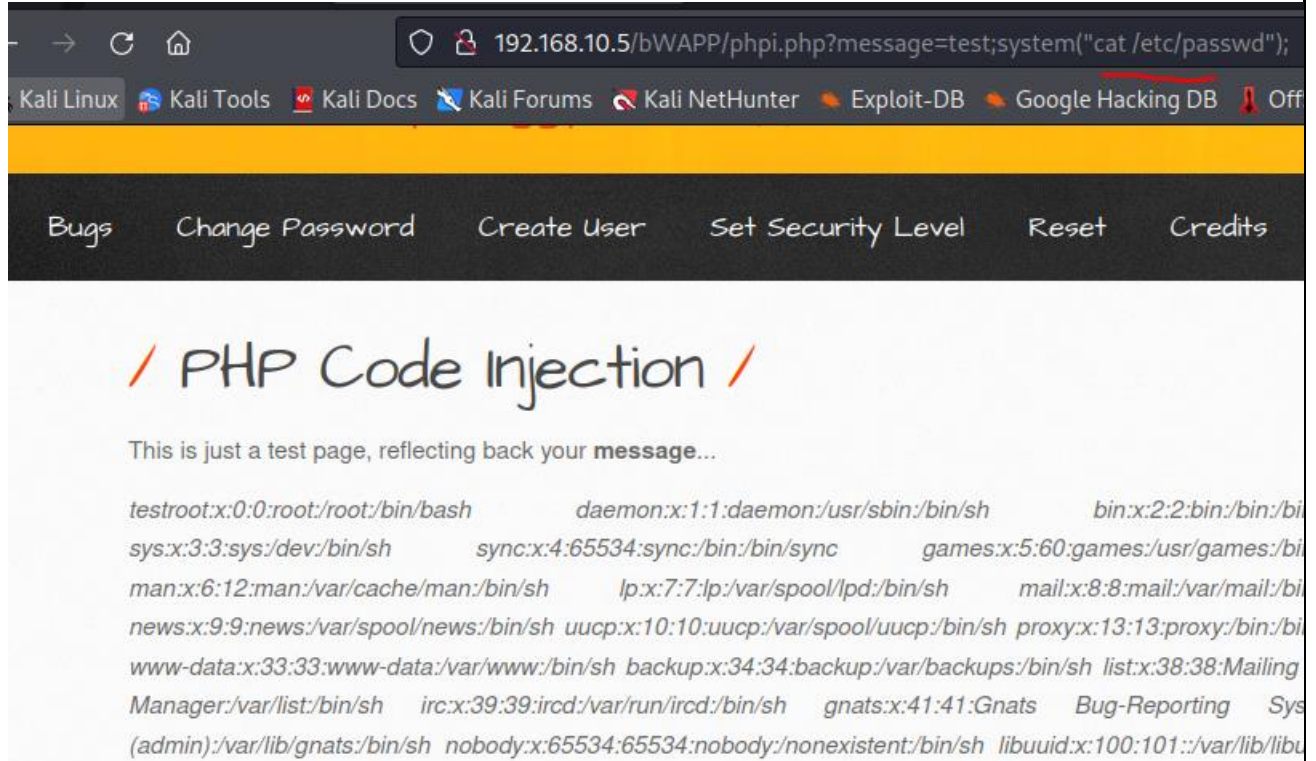
## 6.1.1.5 Command Injection

Açığın Etkisi	Sunucu İşletim Sisteminde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****PHP Code Injection**

Tarayıcıdan gelen veriler ile web sunucusunda işlem yapılması gerekebilir. Böyle durumlarda gelen veriler web sunucusu tarafından işleme sokulabilir. Bu durum saldırganlar tarafından bir fırsat olabilir. Uygulamada url parametreleri üzerinden testler yapılmıştır.

Sistemde url üzerinden çeşitli parameterlerin denenmesi sonucunda php kodlarının çalıştığı gözlemlenmiştir. Sunucu tarafında işletim sisteminin /etc/passwd dosyası okunmuştur. Bu zafiyet uzaktan kod çalıştırmaya imkan vermektedir. Web uygulamasında **PHP Code Injection zafiyeti tespit** edilmiştir.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – Command Injection

**Açıklığı Barındıran Sistemler:**

- ❖ <https://beyaz.net/command>
- ❖ <https://admin.beyaz.net>

**Çözüm Önerileri:**

- ❖ PHP programlama ile kodlanmış programlarda `evil()` fonksiyonun olabildiğince kullanmamak ve bu fonksiyonun parametrelerine dikkat etmek gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**

## 6.1.1.6 Command Injection

Açığın Etkisi	Web Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****Server Side Include Injection**

Server side include programın birden çok dosyasını bir dosya üzerinde birleştirmeye yarar. Bu ssi'di parametreleri dışarıdan kullanıcı inputlarıyla birleştirilmesi sonucu manüple edilebilir.

Saldırgan kendi komutlarını işletim sistemi seviyesinde çalıştırabilir.

Görüldüğü üzere tarayıcının kullanıcı input alanına ssi kodu girilip programın kod akışı bozulmuştur. Girilen kod sayesinde /etc/passwd dosyası okunabilmiştir.

**Zafiyet Kanıtı:**

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

```
Hello root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/s
/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var
/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing List Manager:/var/li
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var/lib/libuuid:/bin/sh dhcp
system user,,:/var/run/hplip:/bin/false avahi-autoipd:x:105:113:Avahi autoip daemon,,:/var/lib/aval
/run/pulse:/bin/false messagebus:x:108:119::/var/run/dbus:/bin/false avahi:x:109:120:Avahi mDNS
haldaemon:x:111:123:Hardware abstraction layer,,:/var/run/hald:/bin/false bee:x:1000:1000:bee,,,
dovecot:x:114:126:Dovecot mail server,,:/usr/lib/dovecot:/bin/false smmta:x:115:127:Mail Transfer
/neo:/bin/sh alice:x:1002:1002::/home/alice:/bin/sh thor:x:1003:1003::/home/thor:/bin/sh wolverine
postfix:x:117:129::/var/spool/postfix:/bin/false proftpd:x:118:65534::/var/run/proftpd:/bin/false ftp:
```

Your IP address is:

**192.168.10.22**

Örnek Ekran Görüntüsü – Server Side Include Injection

**Açıklığı Barındıran Sistemler:**

❖ .../bWAPP/ssii.php

**Çözüm Önerileri:**

❖ Puts() fonksiyonuna verilen parametreler dikkatlice kontrol edilip verilmelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**

## 6.1.1.7 Command Injection

Açığın Etkisi	Veri Tabanı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (GET/Search)**

Kullanıcıdan alınan inputlar filitrelenmeden sunucuya alınıp veri tabanı sorgusuna katılırsa SQL injection zafiyeti oluşur.

Tarayıcı üzerinde input bölgelerine ataklar denenmiş. Ataklar sonucunda veritabanı kullanıcılarının kullanıcı isimleri, şifre hashleri ve mail gibi kritik bilgileri ele geçirilmiştir. Belirtilen sayfada verilen inputlar ile sunucuda **Union Based SQL Injection zafiyeti tespiti** edilmiştir.

Örnek sorgu kodu

**Zafiyet Kanıtı:**

/ SQL Injection (GET/Search) /

Search for a movie: ' and 1=0 union all select 1,login,pass' Search

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	<a href="#">Link</a>
bee	6885858486f31043e5839c735d99457f045affd0	bwapp-bee@mailinator.com	Any bugs?	<a href="#">Link</a>
berkan	50d3dfcd142da75c1a9ae655d02f0ad8aba2cb13	basipa9674@exoacre.com	Any bugs?	<a href="#">Link</a>

Örnek Ekran Görüntüsü – SQL Injection (Get/Search)

**Açıklığı Barındıran Sistemler:**

❖ .../bWAPP/sqli\_1.php

**Çözüm Önerileri:**

❖ Alınan veriler filitrelenmelidir.

---

❖ Referanslar:

- ❖ [https://owasp.org/www-community/attacks/Command Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**

## 6.1.1.8 Command Injection

Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (GET/Select)**

Tarayıcıdan alınan veriler kontrol edilmeden veri tabanı sorgusuna eklenmesi sonucu saldırganlar veritabanında sorgular çalıştırabilir.

Sunucuya veri gönderen atak yüzeyine çeşitli denemelerle saldırılar yapılmıştır. Atak noktasından alınan movie sayısına ek sql sorguları yazıldığında veri tabanı sunucusunda sorgu çalıştığı görülmektedir. Saldırıda veritabanı kullanıcı adları ve şifreleri ele geçirilmiştir. Ele geçirilen bilgilerin bulunduğu örnek resim aşağıdaki gibidir.

**Zafiyet Kanıtı:**

/ SQL Injection (GET/Select) /

Select a movie:

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	<a href="#">Link</a>

Örnek Ekran Görüntüsü – SQL Injection (GET/Select)

**Açıklığı Barındıran Sistemler:**

- ❖ .../bWAPP/sqli\_2.php

**Çözüm Önerileri:**

- ❖ Girdi denetimin kontrol altına alınması hemen hemen her uygulama güvenliği problemi için ciddi manada çözüm sunmaktadır.
- ❖ Girdiler için beyaz liste hazırlanması mümkün ise kullanılması önerilir.

- ❖ Komut enjeksiyonu probleminin çözümünde kullanıcıdan alınan girdilerin pozitif girdi denetiminden geçirilmesi gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)



**Kritik**6.1.1.9 *Command Injection*

Açığın Etkisi	Veri Tabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (POSTSearch)**

Kullanıcı tarafından girilen input noktalarından zararlı kodlar sisteme enjekte edilebilmektedir. Bu noktalara zararlı kodlar saldırgan tarafında enjekte edilirse kod akışını bozmakta ve programın çalışması aksayabilmekte veya saldırgan sistemden önemli verileri sızdıra bilmektedir.

Atak yüzeylerine çeşitli inputlar ile zafiyetli olup olmadığı araştırılmıştır. Araştırmalar sonucunda Sql sorgularını manüpile edilip veritabanından veri çıkarılabildiği ortaya çıkmıştır. Programda **Union Based SQL Injection** zafiyeti keşfedilmiştir.

**Zafiyet Kanıtı:**

/ SQL Injection (POST/Search) /

Search for a movie:

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	<a href="#">Link</a>
bee	6885858486f31043e5839c735d99457f045affd0	bwapp-bee@mailinator.com	Any bugs?	<a href="#">Link</a>
berkan	50d3dfcd142da75c1a9ae655d02f0ad8aba2cb13	basipa9674@exoacre.com	Any bugs?	<a href="#">Link</a>

Örnek Ekran Görüntüsü – SQL Injection (POSTSearch)

**Açıklığı Barındıran Sistemler:**

❖ .../bWAPP/sqli\_6.php

**Çözüm Önerileri:**

- ❖ Veri tabanına sorgu yapan input girdilerini veri tabanı sorgularıyla birleştirmeden iyice dikkat edilmesi gerekir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

**Kritik**

## 6.1.1.10 Command Injection

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (POSTSelect)**

Tarayıcı üzerinden sunucuya gönderdiğimiz veriler filitrelenmezse saldırganlar burdaki oluşacak zafiyeti sömürebilirler. Sql sorgularını yaptığımız web uygulamızda tarayıcıdan gelen veriler ile doğrudan birleşmesi gerekir.

Belirtilen Ugulamada tarayıcı üzerinden web sunucuya veriler gönderilmiştir. Bu gönderilen movie parametresine eklemeler yapılarak sorgular manipüle edilebildiği tespit edilmiştir. Uygulamada **Union Based Sql Injection** zafiyeti tespit edilmiştir.

**Zafiyet Kanıtı:****/ SQL Injection (POST/Select) /**Select a movie:  

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	<a href="#">Link</a>

Örnek Ekran Görüntüsü – SQL Injection (POSTSelect)

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_13.php

**Çözüm Önerileri:**

❖ Tarayıcı tarafından alınan inputlar doğrudan sql sorgusuna sokulmaması gerekmektedir.

**Referanslar:**❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

❖ <https://portswigger.net/web-security/os-command-injection>

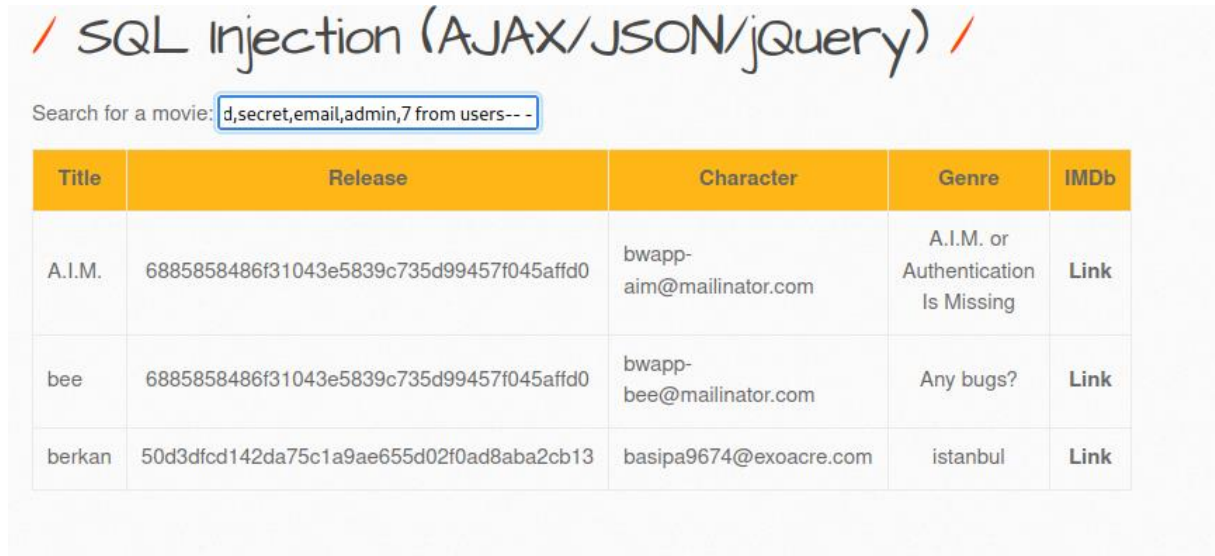
**Kritik**6.1.1.11 *Command Injection*

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (AJAX/JSON/JQuery)**

Ajax yapısını kullanan sistemler input noktasını herhangi bir girdi tetiklemesi durumunda sunucuya eş zamanlı çalışarak istek yapabilmektedir. Ajax İstekleride aynı şekilde filitrelenmeden veritabanı sorgusuna katılması programın istenmeyen kod akışına girmesine sebep olabilmektedir.

Test edilen uygulamada ajax istekleri bulunması sırasında **Union Based Sql Injection** zafiyeti tespit edilmiştir. Bulunan zafiyet veritabanı kullanıcı adı şifre e-mail gibi bilgilerin ele geçirilmesi ile sömürülmüştür.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – SQL Injection (AJAXJSONjQuer)

**Açıklığı Barındıran Sistemler:**

- ❖ /bWAPP/sqli\_10-1.php
- ❖ /bWAPP/sqli\_10-2.php

**Çözüm Önerileri:**

- ❖ Girdi denetimin kontrol altına alınması hemen hemen her uygulama güvenliği problemi için ciddi manada çözüm sunmaktadır.

- ❖ Girdiler için beyaz liste hazırlanması mümkün ise kullanılması önerilir.
- ❖ Komut enjeksiyonu probleminin çözümünde kullanıcıdan alınan girdilerin pozitif girdi denetiminden geçirilmesi gerekmektedir.
- ❖ Veritabanına gönderilecek parametreler iyice süzülüp gönderilmesi gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command Injection](https://owasp.org/www-community/attacks/Command_Injection)

**Kritik**

## 6.1.1.12 Command Injection

Açığın Etkisi	Veri Tabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (CAPTCHA)**

Tarayıcı Üzerinden doğrudan filitrelenmeden yapılan istekler programın kod akışını bozabilmektedir.

Test edilen uygulamada Sql sorgularının yapısı bozularak sorgu dışında veritabanında veri getirilmesi sağlanmıştır. Veritabanından çıkarılan bilgiler arasında veritabanı kullanıcı isimleri şifreleri mailer gibi bilgiler bulunmaktadır. **Uygulamada Union Based Sql Injection** zafiyeti tespit edilmiştir.

**Zafiyet Kanıtı:**

## / SQL Injection (CAPTCHA) /

Search for a movie:

Title	Release	Character	Genre	IMDb
A.I.M.	6885858486f31043e5839c735d99457f045affd0	bwapp-aim@mailinator.com	A.I.M. or Authentication Is Missing	<a href="#">Link</a>
bee	6885858486f31043e5839c735d99457f045affd0	bwapp-bee@mailinator.com	Any bugs?	<a href="#">Link</a>
berkan	50d3dfcd142da75c1a9ae655d02f0ad8aba2cb13	basipa9674@exoacre.com	Any bugs?	<a href="#">Link</a>

Örnek Ekran Görüntüsü – SQL Injection (CAPTCHA)

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_9.php

**Çözüm Önerileri:**

- ❖ Veritabanına gönderilen Sql parameterelerini filitrelemeden sorguyu çalıştırmamak gerekir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)



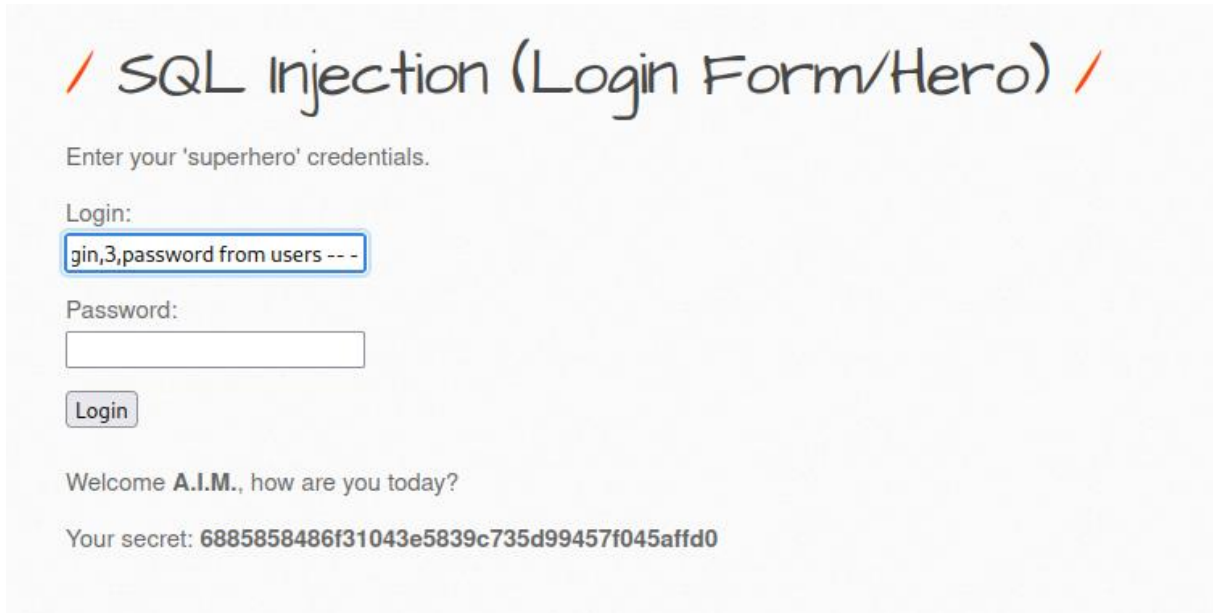
**Kritik**6.1.1.13 *Command Injection*

Açığın Etkisi	Yetkisiz Erişim
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (Login Form/Hero)**

Kullanıcı adı ve password alanları saldırganlar için birer saldırı yüzeyi oluşturmaktadır. Bu login sayfaları bazen bypass edilip yetkisiz erişim kazanılabilmektedir.

Uygulamanın Testinde bahsedilen sayfada çeşitli ataklar denenmiştir. Atakların sonucunda sayfaya hem yetkisiz erişim hem veritabanı kullanıcı bilgileri ele geçirilmiştir. Aşağıda ele geçirilen örnek verilerin olduğu fotoğraf bulunmaktadır.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – Command Injection

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_3.php

**Çözüm Önerileri:**

- ❖ Yetkisiz erişime sql sorgusunu filitrelemeden eklenmesi katılıyor. Sorguyu filitrelemeden web sunucuda birleştirmemelidir.

**Referanslar:**

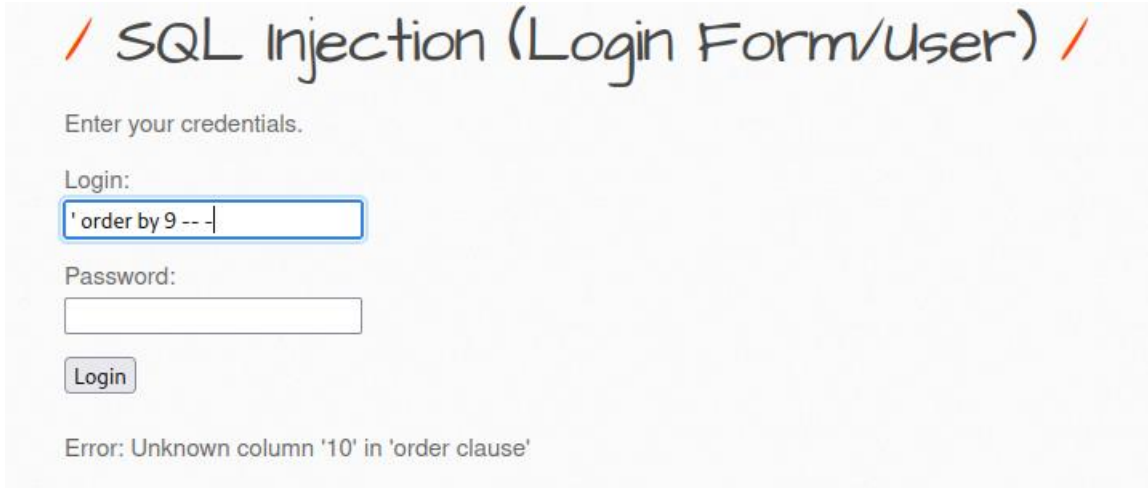
- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**6.1.1.14 *Command Injection*

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (Login Form/User)**

Sistemde Order by komutunun çalışmasına izin veriyor.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – SQL Injection (Login Form/User)

**Açıklığı Barındıran Sistemler:**

- ❖ <https://beyaz.net/command>
- ❖ <https://admin.beyaz.net>

**Çözüm Önerileri:**

- ❖
- ❖ **Referanslar:**
  - ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
  - ❖ <https://portswigger.net/web-security/os-command-injection>



**Kritik**

## 6.1.1.15 Command Injection

Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection (SQLite)**

Uygulamalarda kullanıcı inputlarının olduğu noktalar programların kod akışını değiştirebilecek atak vektörüne sahip olabilmektedir. Uygulama güvenlik testleri için input noktalarına çeşitli ataklar yapılmıştır.

Yapılan ataklar sonucunda uygulamanın SQLite veritabanı sunucunu kullandığı tespit edilmiş veritabanına giden sorgular manipüle edilerek sistemden kullanıcı adı password hashi ve email gibi çeşitli veriler ele geçirilmiştir. Örnek fotoğraf aşağıda gösterildiği gibidir.

**Zafiyet Kanıtı:**

## / SQL Injection (SQLite) /

Search for a movie:   (requires the PHP SQLite module)

Title	Release	Character	Genre	IMDb
1	A.I.M.		6885858486f31043e5839c735d99457f045affd0	<a href="#">Link</a>
2	bee		6885858486f31043e5839c735d99457f045affd0	<a href="#">Link</a>
G.I. Joe: Retaliation	2013	Cobra Commander	action	<a href="#">Link</a>
Iron Man	2008	Tony Stark	action	<a href="#">Link</a>
Man of Steel	2013	Clark Kent	action	<a href="#">Link</a>
Terminator Salvation	2009	John Connor	sci-fi	<a href="#">Link</a>
The Amazing				

Örnek Ekran Görüntüsü – SQL Injection (SQLite)

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_11.php

**Çözüm Önerileri:**

- ❖ Tarayıcıdan gelen verileri filitrelemeden veritabanı sorgu söz dizimine dâhil edilmemesi gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**6.1.1.16 *Command Injection*

Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Stored (Blog)**

Tarayıcıdan aldığımız Bazı inputlar veritabanına kaydolup tekrar karşımıza çıkabilir. Veritabanında bu sorgular kayıtlı olarak kalır.

Uygulamanın input noktalarında yapılan saldırgan vektörleri ile message box kısmında yazılan verilerin veri tabanına kaydedildiği tespit edilmiştir. Bu alanda yazılan verilerin veritabanı sorgusuna eklendiği gözlemlenmiştir. Bu veriler değiştirilerek veritabanı sorgusu manipüle edilmeye çalışılmıştır. Elde edilen verilerde veritabanı kullanıcılarının kullanıcı adı, password hashleri gibi bilgiler ele geçirilmiştir.

**Zafiyet Kanıtı:**

10 A.I.M.,bee,berkan

11 6885858486f31043e5839c735d99457f045affd0,6885858486f31043e5839c735d99457f045affd0,50d3dfcd14

Örnek Ekran Görüntüsü – SQL Injection - Stored (Blog)

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_7.php

**Çözüm Önerileri:**

- ❖ Girdi denetimin kontrol altına alınması hemen hemen her uygulama güvenliği problemi için ciddi manada çözüm sunmaktadır.
- ❖ Girdiler için beyaz liste hazırlanması mümkün ise kullanılması önerilir.
- ❖ Komut enjeksiyonu probleminin çözümünde kullanıcıdan alınan girdilerin pozitif girdi denetiminden geçirilmesi gerekmektedir.
- ❖ Tarayıcıdan veri tabanına giden verileri doğrudan veritabanı sorgusuna katmamalıyız.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

❖ <https://portswigger.net/web-security/os-command-injection>



**Kritik**6.1.1.17 *Command Injection*

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Stored(SQLite)**

Tarayıcıdan alınan verilerin veritabanlarına kaydetmemiz gerekebilir. Veriyi doğrudan veritabanı sunucusuna kaydetmek sunucuda istenmeyen kodların çalışmasına ve bilgi ifşasına neden olabilir.

Uygulama testlerinde sunucuya giden veriler manipüle edilerek gönderilmiştir. Çeşitli atakların sonucunda sistemdeki veritabanı sunucusunda sql komutları çalıştırılıp sistemden veriler elde edilmiştir. Aşağıda örnek ele geçirilen veriler.

**Zafiyet Kanıtı:**

9	CREATE TABLE "blog" ( "id" int(10) NOT NULL , "owner" varchar(100) DEFAULT NULL, "entry" varchar(500) DEFAULT NULL, "date" datetime DEFAULT NULL, PRIMARY KEY ("id" )	2022-06-30	1
10	blog	2022-07-02	1
11	berkan	2022-07-02	1

Örnek Ekran Görüntüsü – Com

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_12.php

**Çözüm Önerileri:**

- ❖ Uygulama input noktalarından alınan verileri filitrelemeden veritabanı sorgusuna eklenip çalıştırılmamalıdır.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**

## 6.1.1.18 Command Injection

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Stored (User-Agent)**

HTTP parametrelerinden olan User-Agent parametresi bir web sunucusu isteğinin nasıl bir cihazdan ve hangi uygulamadan atıldığı bilgisini web sunucusuna taşır. Web sunucusu bu log kayıtlarını sistemde veritabanına yazabilir.

Test edilen uygulamada user-agent parametresi ile veri göndrildiğinde veritabanı sunucusuna kayıt edildiği gözlemlenmiştir. Gönderilen veriler manipüle edilip tekrar gönderildiğinde veritabanından bilgi ifşası olabileceği ve veri tabanının kullanıcı adı ve password hashlerinin ele geçirilebileceği gözlemlenmiştir.

**Zafiyet Kanıtı:**

## SQL Injection - Stored (User-Agent)

Your IP address and User-Agent string have been logged into the database! ([download log file](#))

An overview of our latest visitors:

Date	IP Address	User-Agent
2022-07-02 09:26:31	6885858486f31043e5839c735d99457f045affd0,688585848	1
2022-07-02 09:26:03	192.168.10.16	Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
2022-07-02 09:25:16	A.I.M.,bee,berkan	1

Örnek Ekran Görüntüsü – SQL Injection - Stored (User-Agent)

**Açıklığı Barındıran Sistemler:**

❖ /bwAPP/sqli\_17.php

**Çözüm Önerileri:**

- ❖ Veritabanında depolanacak verileri filitrelemeden geçirip veri tabanına kaydetmek gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**

## 6.1.1.19 Command Injection

Açığın Etkisi	Tarayıcı Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Stored (XML)**

**Not:** Sql hatası alıyorum ama verdiğim inputları string gibi algılıyor örnek aşağıdaki resimdeki gibi

Kullanıcıdan alınan input verilerini sunucuya filitrelenmeden alınması ve tekrar kullanıcılara girilen inputların dönmesi ile kullanıcının tarayıcısı üzerinde zararlı bir kod çalıştırılabilir hale getirmektedir.

**Zafiyet Kanıtı:**

```
POST /bWAPP/sqli_8-2.php HTTP/1.1
Host: 192.168.10.14
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0)
Gecko/20100101 Firefox/91.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-type: text/xml; charset=UTF-8
Content-Length: 70
Origin: http://192.168.10.14
Connection: close
Referer: http://192.168.10.14/bWAPP/sqli_8-1.php
Cookie: PHPSESSID=f453704336acb60db8af4e604b0cfaa9;
security_level=0

<reset>
<login>
' or 1=1 -- -
</login>
<secret>
Any bugs?
</secret>
</reset>
```

```
1 HTTP/1.1 200 OK
2 Date: Sun, 03 Jul 2022 09:58:13 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2 mod_fastcgi
  PHP/5.2.4-2ubuntu5 with Suhosin-Patch mod_ssl/2
  OpenSSL/0.9.8g
4 X-Powered-By: PHP/5.2.4-2ubuntu5
5 Expires: Thu, 19 Nov 1981 08:52:00 GMT
6 Cache-Control: no-store, no-cache, must-revalid
  post-check=0, pre-check=0
7 Pragma: no-cache
8 Content-Length: 39
9 Connection: close
10 Content-Type: text/html
11
12 ' or 1=1 -- - 's secret has been reset!
```

**Örnek Ekran Görüntüsü – SQL Injection - Stored (XML)****Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_8-1.php

**Çözüm Önerileri:**

❖

**Referanslar:**

❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)

❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**6.1.1.20 *Command Injection*

Açığın Etkisi	Veri tabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Blind - Boolean-Based**

Tarayıcı alanlarına girilen veriler veri tabanı sorgusuna doğrudan bağlana bilmektedir. Veritabanı sorgusu manipüle edilip veri çekilebilmektedir.

Test edilen uygulamaya çeşitli inputlar verilip denendiğinde input bölgesinden çeşitli komutların çalıştığı tespit edilmiştir. **Uygulamada Blind Based based Sql Injection** açığı tespit edilmiştir. Veri tabanı isimleri elde edilmiştir.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – Command Injection

**Açıklığı Barındıran Sistemler:**

- ❖ /bWAPP/sqli\_4.php

**Çözüm Önerileri:**

- ❖ Input alanlarındaki verileri filitrelemeden doğrudan veri tabanı sorgusu ile birleştirilmemelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**6.1.1.21 *Command Injection*

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection – Blind – Time-Based**

Kullanıcılardan alınan veriler ile veritabanında sorgular yapmamız gerekebilir. Bu durumda input noktalarından alınan veriler ile sql sorgularını web sunucusunda birleştirmemiz gerekebilir. Bu birleştirme verileri manüplasyonu ile sistemde kötü durumlara yol açabilir.

Uygulamada input noktalarına denenen atak vektörleri ile belirtilen input noktasında sql sorgularının çalıştırılabildiği saptanmıştır. Input noktasına sorgu parameteresinin yanınada sql sorgusunda gönderildiğinde çalıştırıldığı saptanmıştır. Uygulamada **Blind Time-Based SQL Injection** Zafiyeti saptanmıştır.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – SQL Injection – Blind – Time-Based

**Açıklığı Barındıran Sistemler:**

- ❖ .../bWAPP/sqli\_15.php

**Çözüm Önerileri:**

- ❖ Input noktalarından alınan veriler SQL sorusuna katılacaksa flitrelenip kontrol edilmelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>



**Kritik**6.1.1.22 *Command Injection*

Açığın Etkisi	Web Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Blind (SQLite)**

Son kullanıcılardan alınan veriler ile web sunucularında bazı işlemler gerçekleştirmek gerekebilir. Bu verileri web sunucusuna gönderirken gerekli güvenlik önlemleri alınmadığı durumda istismara açık olabilir.

Test edilen uygulamada belirtilen input noktasına çeşitli saldırı tipleriyle zafiyet araştırması yapılmıştır. Yapılan araştırmada belirtilen input noktasına sql sorguları gönderildiğinde web sunucusu cevap verdiği saptanmıştır. Uygulamada Blind Based sql injection

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – SQL Injection - Blind (SQLite)

**Açıklığı Barındıran Sistemler:**

- ❖ .../bWAPP/sqli\_14.php

**Çözüm Önerileri:**

- ❖ Kullanıcı tarafından web sunucuya giden verileri filitrelemek gerekmektedir.
- ❖ Sunucudan giden ve gelen veriler için **ansenkron** bir yapıda olması gerekmektedir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

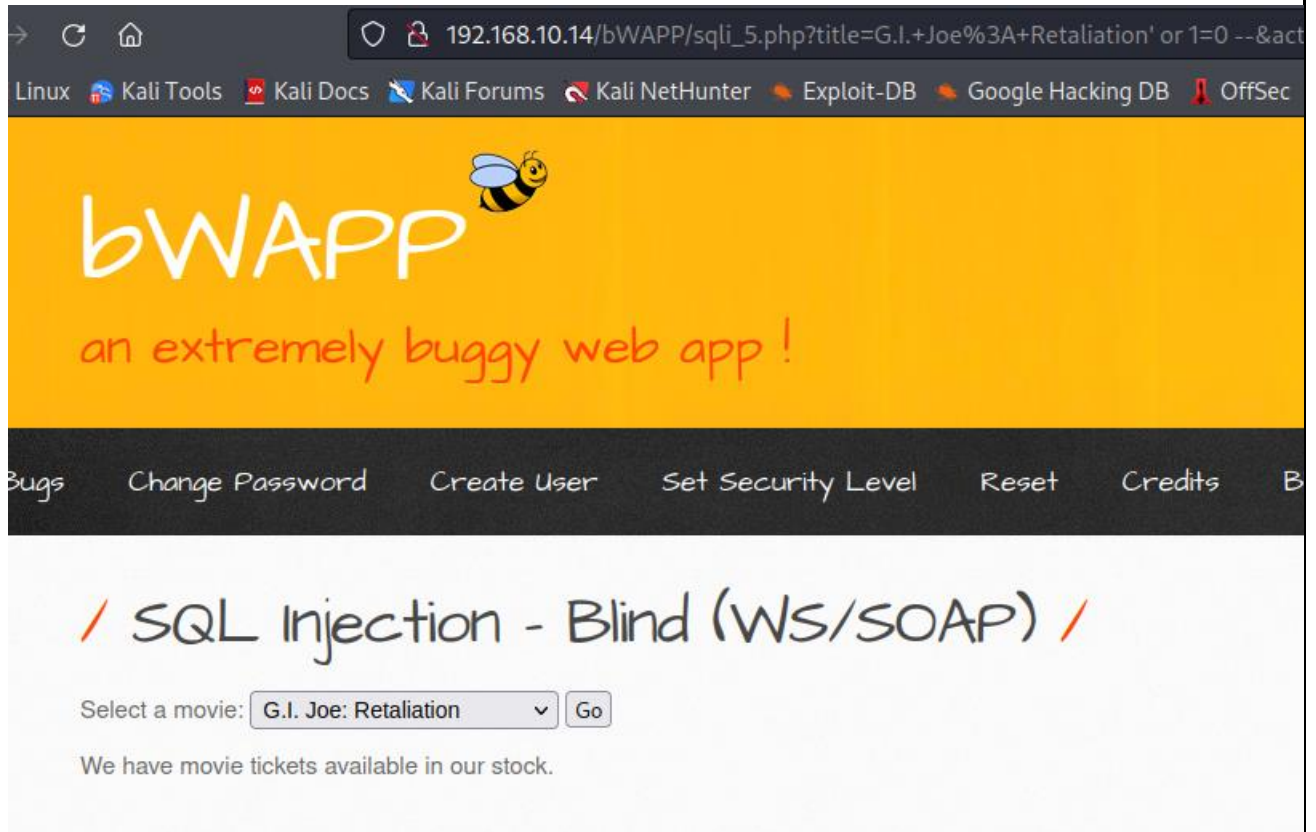
**Kritik**6.1.1.23 *Command Injection*

Açığın Etkisi	Veritabanı Sunucusu Üzerinde Kod Çalıştırılabilir
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****SQL Injection - Blind (WSSOAP)**

Tarayıcıdan veri gönderen input noktaları web sunucusunda bir takım işlemler için kullanılabilir. Özellikle sql sorgularına katılıp veri tabanı işlemlerinde kullanılabilir.

Test edilen uygulamada Get methodu üzerinden veri gönderdiği tespit edilmiştir. Buradan yola çıkarak çeşitli saldırılar yapılmıştır. Saldırıların sonucunda url üzerinde title parametresine sql sorguları eklendiğinde veri tabanında çalıştırılabildiği gözlemlenmiştir. Uygulamada **Blind Based Sql Injection** Keşfedilmiştir.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü – SQL Injection - Blind (WSSOAP)

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/sqli\_5.php

**Çözüm Önerileri:**

- ❖ Sql sorgularını gönderildiği parametreleri doğrudan sql sorgusu ile birleştirilmemelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**6.1.1.24 *Command Injection*

Açığın Etkisi	Web Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****XML/XPath Injection (Login Form)**

Bazı durumlarda veriler xml formatında web sunucusunda saklanabilir. Web sunucusu bu verilere ekleme çıkarma ve değiştirme gibi işlemler yapabilir. Bu durumlarda gerekli önlemler alınmazsa saldırganlar için fırsat oluşturabilir.

Uygulamada input noktalarından login formu test edilmiştir. Burada gönderilen veriler manipüle edilerek login formu geçilmiştir. Uygulamada **XML/Xpath Injection** bulunmuştur.

**Zafiyet Kanıtı:**

**/ XML/XPath Injection (Login Form) /**

Enter your 'superhero' credentials.

Login:

Password:

Login

Welcome **Neo**, how are you today?

Your secret: **Oh why didn't I took that BLACK pill?**

Örnek Ekran Görüntüsü – XML/XPath Injection (Login Form)

**Açıklığı Barındıran Sistemler:**

- ❖ /bWAPP/xmli\_1.php

**Çözüm Önerileri:**

- ❖ Input noktalarından veri alınacağı zaman dikkatlice filitrelenmelidir.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

**Kritik**6.1.1.25 *Command Injection*

Açığın Etkisi	Web Sunucusu Üzerinde Kod Çalıştırabilme
Erişim Noktası	İnternet
Kullanıcı Profili	Standart Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler
Bulgu Portu	8080
Referans Kodu	-

**Bulgu Açıklaması****XMLXPath Injection (Search)**

Web uygulamaları verilerini depolamak için farklı çözümler geliştirebilirler. Xml formatında veri tutulmasında bu çözümlerden bir tanesidir. Web uygulamaları xml formatında veri tutup bu format üzerinden veri manipilasyonu yapabilir. Kullanıcılardan veri alınıp uygulama içinde kullanılması durumunda zafiyet oluşturabilmektedir.

Test yapılan uygulamada alınan verilere xpath sorgularının eklenmesi ile uygulamadan veri ifşası gerçekleşmiştir. Uygulamada XML/XPath Injection zafiyeti tespit edilmiştir.

**Zafiyet Kanıtı:**

Örnek Ekran Görüntüsü –

**Açıklığı Barındıran Sistemler:**

❖ /bWAPP/xmli\_2.php

**Çözüm Önerileri:**

- ❖ Kullanıcıdan alınan input noktaları verileri dikkatlice filitrelenip web sunucusuna alınmalıdır.

**Referanslar:**

- ❖ [https://owasp.org/www-community/attacks/Command\\_Injection](https://owasp.org/www-community/attacks/Command_Injection)
- ❖ <https://portswigger.net/web-security/os-command-injection>

## 7 Sızma Testinde Kullanılan Araçlar & Yazılımlar

Kali	Birçok Sızma Testi aracını içeren Linux dağıtımıdır.
Nessus	Ticari zafiyet tarama aracıdır.
Acunetix	Web uygulama ve web servis zafiyet tarama yazılımıdır.
Netsparker	Web uygulama ve web servis zafiyet tarama yazılımıdır.
Burp Suite Pro	Web uygulama zafiyet tarama ve proxy yazılımıdır.
Nmap	Gelişmiş yetenekleri zafiyet tespitine kadar ulaşabilen ağ yapısı tespiti aracıdır.
Metasploit Framework	Gelişmiş özelliklere sahip exploit frameworktür.
Nikto	Web uygulama ve sunucu zafiyet tarama yazılımıdır.
Sqlmap	Web uygulama veri tabanı zafiyet tarama yazılımıdır.
Hydra	Pek çok servisi destekleyen kimlik doğrulama bilgileri kırma aracıdır.
DirBuster	Web uygulama ve sunuculardaki dizinleri ve dosya isimlerini taramaya yarayan bir Java yazılımıdır.
WPScan	WordPress üzerindeki zafiyetleri taramaya yarayan bir araçtır.
John the Ripper	Pek çok servisi destekleyen kimlik doğrulama bilgileri kırma aracıdır.



## 8 Terimler Sözlüğü

Hardcopy	Fiziksel Ortamda Çoğaltmak, Kopyalamak
Softcopy	Elektronik Ortamda Çoğaltmak, Kopyalamak
ISP	İnternet Servis Sağlayıcı
Dos/Ddos	Denial Of Service – Servis Dışı Bırakma
VPN	Virtual Private Network – Sanal Özel Ağ
Reflected	Yansıtılmış
Stored	Kalıcı
Xss	Cross Site Scripting – Siteler Arası Komut Çalıştırma
Payload	Kod Parçacığı
Client	İstemci
Server	Sunucu
Shell	Komut Ara yüzü
SMB	Server Message Block – Sunucu İleti Bloğu
Encrypt	Şifrelemek
MİTM	Man İn The Middle – Ortadaki Adam Saldırısı
SSH	Secure Shell – Güvenli Veri İletimi İçin Ağ Protokolü
SNMP	Simple Network Management Protocol - Basit Ağ Yönetim Protokolü
Meterpreter	Metasploit Üzerinde Bulunan Bir Payload.
Exploit	Sömürmek, İstismar Etmek
Auxiliary	Modüller İçin Geliştirilen Ek Programlar, Yardımcı Araçlardır.