Zafiyet Testi Örnekleri

Zafiyet Testi Metodolojileri

Zafiyet testi için çeşitli metodolojiler mevcuttur. En popüler olanlardan bazıları şunlardır:

- **OWASP Zafiyet Testi Kılavuzu**: OWASP tarafından sunulan kapsamlı bir zafiyet testi metodolojisi.
- **Penetration Testing Execution Standard (PTES)**: Sızma testi için kapsamlı bir standart sağlayan bir metodoloji.
- **NIST SP 800-115**: NIST tarafından yayınlanan ve bilgisayar sistemlerinin güvenlik testi için kılavuz sağlayan bir belge.
- **ISSAF**: Bilgi Güvenliği Sızma Testi Çalışma Grubu'nun (Information Systems Security Assessment Framework) metodolojisi.

OWASP Tarafından Belirlenen Web Uygulaması Zafiyetleri

OWASP (Open Web Application Security Project) tarafından belirlenen ve sıkça karşılaşılan web uygulaması zafiyetlerinden bazıları şunlardır:

- SQL Injection: Kötü niyetli kullanıcıların veritabanına kötü kod enjekte etmesine izin veren bir zafiyet.
- Cross-Site Scripting (XSS): Kötü niyetli kullanıcıların web uygulaması üzerinden kullanıcılara zararlı kodlar göndermesine izin veren bir zafiyet.
- Cross-Site Request Forgery (CSRF): Kötü niyetli kullanıcıların yetkili bir kullanıcı oturumunu kötüye kullanarak, izinsiz işlemleri gerçekleştirmesine izin veren bir zafiyet.
- Insecure Deserialization: Güvensiz deserializasyon işlemleri sonucu kötü niyetli kodların yürütülmesine olanak tanıyan bir zafiyet.
- Broken Authentication: Kullanıcı kimlik doğrulama ve oturum yönetimi işlemlerindeki zafiyetler nedeniyle yetkisiz erişimlerin gerçekleşmesine olanak tanır.

Test Örnekleri 1

Senaryo: E-ticaret Web Sitesinde SQL Injection Saldırısı

Amaç: Web sitesindeki bir SQL Injection zafiyetini tespit etmek ve gidermek.

Adımlar:

1. **Hedef Belirleme**: E-ticaret web sitesinin belirlenmesi ve zafiyet testinin bu site üzerinde yapılacağının kararlaştırılması.
2. **Bilgi Toplama**: Web sitesinin yapısı ve veritabanı bağlantıları hakkında bilgi toplanması. Örneğin, site URL'si, formlar, veritabanı motoru vb.
3. **Zafiyet Taraması**: Bir zafiyet tarama aracı kullanılarak web sitesinde SQL Injection zafiyeti taraması yapılması.
4. **Manuel Denetimler**: Otomatik araçlarla tespit edilemeyen zafiyetlerin manuel olarak incelenmesi. Örneğin, formların elle test edilmesi ve giriş alanlarına özel SQL sorguları enjekte edilmesi.
5. **Zafiyetin Analizi ve Sınıflandırılması**: Eğer web sitesinde SQL Injection zafiyeti bulunursa, zafiyetin potansiyel etkilerinin ve risklerinin değerlendirilmesi.
6. **Raporlama**: Zafiyetin bulunduğuna dair detaylı bir rapor hazırlanması. Raporda, zafiyetin nedenleri, etkileri ve çözüm önerileri bulunmalıdır. Ayrıca, zafiyetin derecesi ve öncelik seviyesi belirtilmelidir.

- **Örnek**: E-ticaret web sitesindeki kullanıcı giriş formu URL'sindeki "username" parametresine bir SQL Injection saldırısı gerçekleştirilerek, siteye giriş yapmak için kullanılan SQL sorgusuna zararlı bir komut enjekte edilir. Örneğin, "username" parametresine "' OR 1=1 --" değeri gönderildiğinde, SQL sorgusunun doğru olduğunu belirten bir koşul eklendiği için giriş başarılı olur. Bu, kullanıcının giriş yapmadan önce gerekli kimlik doğrulamasını atlamasına neden olabilir.

Test Örnekleri 2

Senaryo: E-ticaret Web Sitesinde Cross-Site Scripting (XSS) Saldırısı

Amaç: Web sitesindeki bir XSS zafiyetini tespit etmek ve gidermek.

Adımlar:

1. **Hedef Belirleme**: E-ticaret web sitesinin belirlenmesi ve zafiyet testinin bu site üzerinde yapılacağının kararlaştırılması.
2. **Bilgi Toplama**: Web sitesinin yapısı ve kullanılan teknolojiler hakkında bilgi toplanması. Özellikle, kullanıcı giriş alanları, form alanları ve veri gösterim alanlarının belirlenmesi önemlidir.
3. **Zafiyet Taraması**: Web sitesinde XSS zafiyetlerini tespit edebilecek otomatik araçlar kullanılarak bir zafiyet taraması yapılması.
4. **Manuel Denetimler**: Otomatik araçlarla tespit edilemeyen zafiyetlerin manuel olarak incelenmesi. Özellikle, form alanlarına zararlı kodlar enjekte edilerek XSS zafiyetlerinin tespit edilmesi önemlidir.
5. **Zafiyetin Analizi ve Sınıflandırılması**: Bulunan XSS zafiyetlerinin potansiyel etkilerinin ve risklerinin değerlendirilmesi. Özellikle, kullanıcıların tarayıcılarında zararlı kodların çalıştırılması sonucunda ne gibi güvenlik açıkları oluşabileceğinin analiz edilmesi önemlidir.
6. **Raporlama**: Zafiyetlerin bulunduğuna dair detaylı bir rapor hazırlanması. Raporun içerisinde zafiyetlerin nedenleri, etkileri ve çözüm önerileri bulunmalıdır. Ayrıca, zafiyetlerin derecesi ve öncelik seviyesi belirtilmelidir.

- **Örnek**: E-ticaret web sitesinin bir ürün yorumu bölümünde kullanıcılar tarafından girilen metinlerin doğrudan sayfaya yansıtıldığı ve bu metinlerin bir JavaScript kodu içerdiği fark edilir. Bir saldırgan, bu metin alanına zararlı bir JavaScript kodu ekleyerek, diğer kullanıcıların tarayıcılarında bu kodun çalışmasını sağlayabilir ve kötü niyetli işlemler gerçekleştirebilir.

Test Örnekleri 3

Senaryo: Bir Şirketin İç Ağındaki Bir Sunucuda Sistem Güvenlik Ayarlarının Eksikliği

Amaç: Şirketin iç ağındaki bir sunucuda sistem güvenlik ayarlarının eksikliğini tespit etmek ve gidermek.

Adımlar:

1. **Hedef Belirleme**: Şirketin iç ağındaki sunucuların ve sistemlerin belirlenmesi ve güvenlik testinin bu sistemler üzerinde yapılacağının kararlaştırılması.
2. **Bilgi Toplama**: Sunucuların ve sistemlerin yapılandırma bilgilerinin toplanması. Özellikle, işletim sistemi versiyonları, güncellemelerin durumu, açık portlar ve hizmetlerin listesi gibi bilgilerin toplanması önemlidir.
3. **Zafiyet Taraması**: Güvenlik tarama araçları kullanılarak sunucuların ve sistemlerin zafiyet taraması yapılması. Bu tarama sırasında, güvenlik açıkları ve eksiklikler tespit edilmeye çalışılır.
4. **Manuel Denetimler**: Otomatik araçlarla tespit edilemeyen zafiyetlerin manuel olarak incelenmesi. Özellikle, sistem ayarlarının ve konfigürasyonlarının güvenlik açısından kontrol edilmesi önemlidir.
5. **Zafiyetin Analizi ve Sınıflandırılması**: Bulunan güvenlik açıklarının ve eksikliklerin potansiyel etkilerinin ve risklerinin değerlendirilmesi. Özellikle, bu açıkların kötü niyetli kullanıcılar tarafından nasıl kötüye kullanılabileceğinin analiz edilmesi önemlidir.
6. **Raporlama**: Bulunan güvenlik açıklarına dair detaylı bir rapor hazırlanması. Raporun içerisinde açıkların nedenleri, etkileri ve çözüm önerileri bulunmalıdır. Ayrıca, açıkların derecesi ve öncelik seviyesi belirtilmelidir.

- **Örnek**: Şirketin iç ağındaki bir sunucuda güncel güvenlik yamalarının eksik olduğu ve bu nedenle sunucunun dışarıdan erişilebilir olduğu tespit edilir. Bir saldırgan, bu sunucuya dışarıdan erişerek yetkisiz işlemler gerçekleştirebilir veya hassas verilere erişebilir.

Kaynakça :

- Open Web Application Security Project: https://owasp.org/
- Penetration Testing Execution Standard: http://www.pentest-standard.org
- National Institute of Standards and Technology: https://www.nist.gov
- BIL 114 YAZILIMDA GÜVENLİK 7.HAFTA SUNUMU