JOURNAL of SOCIAL and HUMANITIES SCIENCES RESEARCH (JSHSR)

Uluslararası Sosyal ve Beşeri Bilimler Araştırma Dergisi

Received/Makale Geliş 06.10.2021
Published /Yayınlanma 30.12.2021
Article Type/Makale Türü Research Article

Citation/Alıntı: Of, M. & Kılıçaslan, İ. (2021). XSS (CROS SITE SCRIPTING) web güvenliği açığının işletmeler açısından önemi. *Journal of Social and Humanities Sciences Research, 8*(77), 3144-3152. http://dx.doi.org/10.26450/jshsr.2837

Öğr. Gör. Mustafa OF
https://orcid.org/0000-0002-7924-9073
Kocaeli Üniversitesi, Kocaeli Meslek Yüksekokulu, Kocaeli / TÜRKİYE

Öğr. Gör. İsmail KILIÇASLAN
https://orcid.org/0000-0002-8443-9912
Kocaeli Üniversitesi, Ali Rıza Veziroğlu Meslek Yüksekokulu, Kocaeli / TÜRKİYE

Issue/Sayı: 77 Volume/Cilt: 8 jshsr.org ISSN: 2459-1149

XSS (CROS SITE SCRIPTING) WEB GÜVENLİĞİ AÇIĞININ İŞLETMELER AÇISINDAN ÖNEMİ

IMPORTANCE OF XSS (CROS SITE SCRIPTING) WEB SECURITY VULNERABLE FOR BUSINESSES

ÖZET

Web uygulamaları tüm dünya çapında internet oldukça yaygın bir şekilde kullanılmaktadır. Bu uygulamalar, günlük hayatımızı daha kolay ve etkileşimli bir hale getiren uygulamalardır. Web üzerinden oldukça geniş bir boyutta kişiye özel bilgiler paylaşılıyor. Web tabanlı uygulamalarının en temel önceliği bilgilerin bir bütün olarak sunulması ve gizli tutulmasıdır. Web tabanlı uygulamalar, siber saldırılara karşı oldukça savunmasızdır. XSS ve SQL enjeksiyonu en çok ortaya çıkan saldırılardan biridir. OWASP olarak anılan Açık Web Uygulama Güvenliği Projesi çevrimiçi topluluğu, 2021 yılın ilk 10 kritik güvenlik tehditleri sıralamasında XSS'i üçüncü sıraya koymuştur. Saldırganlar, web uygulamasına kötü amaçlı kodlar enjekte ederek, kişinin rızası olmadan büyük zararlara yol açabilir. Siteler arası komut dosyası çalıştırma (XSS) ve SQL enjeksiyon saldırıları, bir web uygulamasının karşılaştığı en çok karşılaşılan saldırı türleridir. XSS saldırısı, sunucuda çalışan ve kullanıcının farkında olmadığı ve felaketle sonuçlanabilecek bir web tarayıcısına kötü amaçlı kod yürütülmesinden kaynaklanır. Kötü amaçlı veya iyi huylu bir komut dosyasının tanınması, istenmeyen bir saldırının olmasını engelleyebilir ve web uygulamasının kullanıcıların verilerini gizli tutmasına yardımcı olabilir. Bu çalışmada en çok karşılaşılan güvenlik tehditleri açıklanacak, XSS saldırılarına ait ayrıntılı bilgiler verilecek, bu saldırılardan korunma teknikleri hakkında çarpıcı bilgiler sunulacaktır. İşletmelerin beyni olan bilişim uzmanlarının siber saldırılar hakkında daha dikkatli olma noktasında farkındalık sağlayacaktır.
Anahtar Kelimeler: Xss, Web Uygulamaları Güvenliği, OWASP.

ABSTRACT

Web applications are widely used all over the world. Applications are apps that make our daily life easier and more interactive. A wide range of personal information is shared over the web. The most basic priority of web-based applications is to present the information as a whole and to keep it confidential. Web-based applications are highly vulnerable to cyber-attacks. XSS and SQL injection is one of the most emerging attacks. The Open Web Application Security Project online community, known as OWASP, placed XSS in third place in their 2021 top 10 critical security threats. By injecting malicious code into the web application, attackers can cause great harm without the person's consent. Cross-site scripting (XSS) and SQL injection attacks are the most common types of attacks a web application encounters. An XSS attack is caused by executing malicious code into a web browser running on the server that the user is unaware of and can be catastrophic. Recognizing a malicious or harmless script can prevent an unwanted attack and help the web application keep users' data private. In this study, the most common security threats will be explained. Detailed information about XSS attacks will be given. Detailed information about the techniques of protection from these attacks will be presented. It will raise awareness of IT experts, who are the center of businesses, to be more careful about cyber-attacks.
Keywords: Xss, Web Applications Security, OWASP.

1. GİRİŞ

Bilişim dünyası artık eskisi gibi sakin bir dünya değil. Her an insanların bilişim güvenliği noktasındaki zayıf taraflarını iyi bilen ve biraz da teknik bilgiye sahip olan ve adına hacker veya bilgisayar korsanı denilen kişiler tarafından bir saldırıya maruz kalınabilir. Saldırı sadece şifreleri çalma vb. veri hırsızlığı yönünde olmayabilir. Aksine keyfi olarak da bir sunucunun vermiş olduğu hizmeti sekteye uğratma yönünde olabilir. Artık eskisi gibi ülkeler arasında sıcak savaşlar gerçekleşmiyor. Savaşlar, bilgisayar dünyasında yapılıyor. Ülkeler siber güvenlik ordularını oluşturuyor. Geçenlerde Türkiye Bilgi Teknolojileri ve iletişim Kurumu (BTK), bir yarışma düzenleyerek dereceye girenlerden bazılarını BTK'da Ulusal Siber Olaylara Müdahale Merkezi'nde (USOM) işe aldı. Bu kişiler ülkenin siber güvenlik ordusunda görev yapmaya başladılar. Bilişim sistemlerinin kapasitesi büyüdükçe aynı oranda güvenlik açığı da genişlemektedir. Sistemler, yazılım paketlerinden meydana geldikleri için bu yazılımları geliştiren programcıların güvenlik konusunda gösterdikleri dikkat ve hassasiyet oldukça önemli olmaktadır (URL 3). Genellikle yazılım geliştiriciler güvenlik noktasında gerekli hassasiyete sahip değildirler. Sonuç olarak güvensiz yazılımlar birçok güvenlik açığı ortaya çıkarmaktadırlar. Bu duruma kullanılan işletim sisteminin güven derecesi eklendiğinde sonuç oldukça vahim olmaktadır. Kuzey Kore'nin yerel işletim sistemi olan Red Star OS'da bile güvenlik açıkları bulundu. Özellikle kapalı bir alanda kullanılan bir işletim sistemi olmasından dolayı bu işletim sistemi en güvenli olanların arasındaydı. (URL 5) Güvenlik açığı olmayan bir sistem geliştirmek oldukça zordur. Gerekli olan sistem geliştiricilerin ve aynı zamanda son kullanıcıların siber güvenlik konusunda gerekli bilgi ve farkındalığa sahip olmalarıdır. ABD merkezli siber güvenlik kuruluşu olan Cybersecurity Ventures'a göre siber saldırıların 2021 yılında küresel ekonomiye zararının 6 trilyon dolara ulaşması bekleniyor. (URL 8)

Bilinen sıcak savaş çeşitleri artık yerini siber savaşa bırakmıştır. Siber savaş, bilgisayar ve iletişim teknolojilerini güçlü bir şekilde kullanarak hedef olarak seçilen ortama veya topluma farklı amaçlara yönelik olarak zarar verme amacını gütmektedir. Ülkelerin önemli bilgi ve belgelerine siber casusluk teknikleri ile ulaşılabildiği birçok örnek mevcuttur. Siber casusluk, bilişim teknolojilerinin ilerlemesiyle ortaya çıkmış ve her geçen gün önemi artan bir konudur. Kullanıcıların zafiyeti, bilgisayar sistemlerinin kontrol edilmesi güç olan güvenlik açıkların bulunması, güvenlik kontrolü yapabilecek uygulamaların sistemlerde yeteri kadar kullanılmaması sonucunda birçok belge ve bilgi yetkisi olmayan kişilerin eline geçmiştir (Bayraktar, 2015: 51).

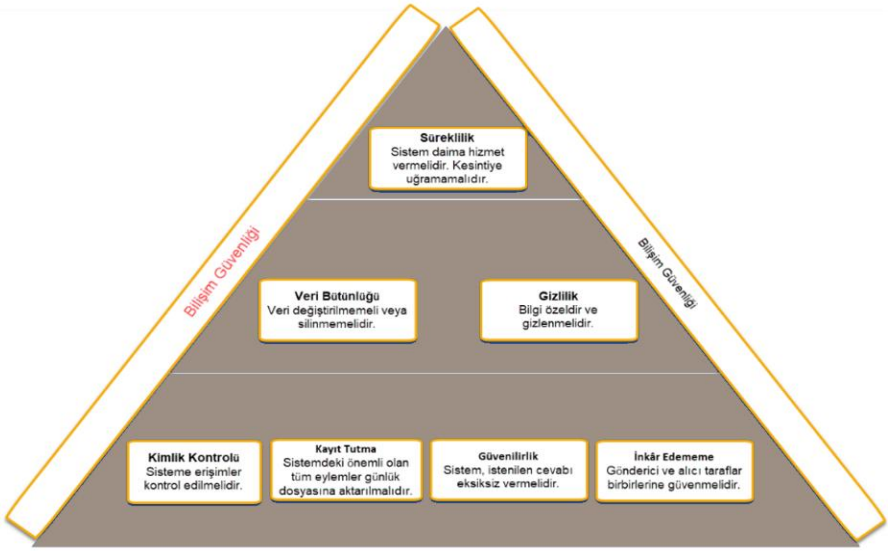
Bu çalışmada, çalışma hayatında kullanılan sistemlerin kontrolünü sağlayan yazılımlar ile veri alışverişinin güvenliği üzerinde durulacaktır. Alınması gerekli olan önlemlerin neler olduğu anlatılmaya çalışılacaktır. En zararlılar listesinde ilk 10'da bulunan XSS'in tanımı yapılacak ve korunma metotları izah edilecektir. (URL 6)

2. BİLİŞİM GÜVENLİĞİ VE YAZILIM OLGUNLUK MODELLERİ

Yazılım güvenliği, yazılımın kendisinden meydana gelen, çalıştığı işletim sistemi ile etkileşimden oluşan veya farklı kullanım çeşitleri durumunda yazılımı kullanan kişi veya sistemlerin karşılaşabileceği güvenlik ataklarına karşı durabilmek için geliştirilen tedbirlerin tamamına verilen isimdir. Güvenlik kavramı ile ilgili başvurulan her türlü tedbirde uygulanan kurallar topluluğudur.

Bilişim güvenliği, aşağıdaki temel kavramlardan meydana gelmektedir;

- Süreklilik: Sistem daima hizmet vermelidir. Kesintiye uğramamalıdır.
- Veri Bütünlüğü: Veri değiştirilmemeli, silinmemeleri, orijinali korunmalıdır.
- Gizlilik: Bilgi özeldir ve gizlenmelidir.

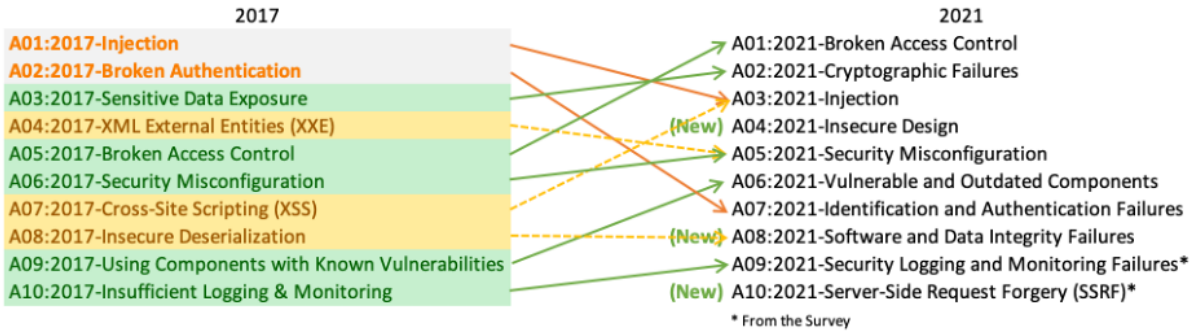


Şekil 1: Bilişim Güvenliğinin Temel Kavramları

Kullanıcıların isteklerine ve taleplerine bağlı olarak oluşturulan yazılımlarda en çok gözden kaçırılan kısım "Güvenilir" olma özelliğidir. Eksiksiz ve sorunsuz çalışan bir uygulama normal şartlar altında çalışabildiği gibi güvenlik seviyesi düşük olan durumlarda da aynı tepkiyi verebilecek şekilde tasarlanmalıdır. Yazılımın belirli aşamalarda daha güvenli bir şekilde oluşturulabilmesine rehberlik edebilecek yazılım olgunluk modelleri geliştirilmiştir. OpenSAMM (Open Software Assurance Maturity Model http://www.opensamm.org) bunlardan biridir. Yazılım Güvencesi Olgunluk Modeli (SAMM), kuruluşların karşı karşıya kaldıkları özel risklere göre uyarlanmış bir yazılım güvenliği stratejisi oluşturmalarına ve uygulamalarına yardımcı olan açık bir çerçevedir. Bir OWASP (Open Web Application Security Project) projesi olarak geliştirilmiştir. Açık Web Uygulaması Güvenlik Projesi (OWASP), yazılım güvenliğini arttırmaya odaklanmış, dünya çapında kâr amacı gütmeyen bir kuruluşudur.

3. GÜVENLİK TEHDİTLERİ SIRALAMASI

OWASP (Open Web Application Security Project), 2017 yılında karşılaşılan ve 2021 yılında ortaya çıkan en kritik olarak ilk 10 web uygulamalarına ait güvenlik zafiyetlerini ortaya çıkartmıştır. Aşağıdaki görselde kritik uygulamalarının değişen şartlara göre yeni isimlerle anıldığı görülmektedir.



Şekil 1: OWASP 2021 yılı ilk 10 kritik uygulama listesi, **Kaynak:** URL1

OWASP'ın belirlediği güvenlik tehditleri uygulamalarını kısaca aşağıdaki şekilde açıklayabiliriz.

1. **Injection**
   Kullanıcıdan alınan verileri çeşitli şekillerde değiştirmeye dayalı tehditlerdir. Sql Injection, Xml Injection, Code Injection, bunlardan sadece birkaçıdır.

2. **Broken Authentication and Session Management**

   Oturumun kontrol dışı olarak yönetilmesinden kaynaklı tehditlerdir. Oturum sabitleme (Session Fixation) ve oturumu tahmin etme (Session Prediction), olarak örnek verilebilir.

3. **Sensitive Data Exposure**

   Hassas ve önemli olan verilere erişimin çok kolay olması sonucunda oluşan tehditlerdir. Veri tabanı yönetim sistemine erişim yetkisinin birçok kullanıcıda olması örnek olarak verilebilir.

4. **Xml External Entity (XEE)**

   XML dış varlık enjeksiyonu (XXE), bir saldırganın bir uygulamanın XML verilerini işlemesine müdahale etmesine izin veren bir web güvenlik açığıdır. Genellikle bir saldırganın uygulama sunucusu dosya sistemindeki dosyaları görüntülemesine ve uygulamanın kendisinin erişebileceği herhangi bir arka uç veya harici sistemle etkileşime girmesine izin verir.

5. **Broken Access Control**

   Bazı durumlarda yetkilendirme açığı olarak adlandırılan erişim kontrolü, bir web uygulamasının içeriğe erişim sağlamasına izin vermesidir. Geliştiricilerin güvenlik erişimlerine yeteri kadar önem vermemesi sonucunda ortaya çıkmaktadır. Güvenlik erişimlerinin daha sıkı kontrollerden geçmesi sonucunda bu açığın etkisi azalacaktır.

6. **Security Misconfiguration**

   Çalışan sistemlerin genel ayarlarının ilk başlangıç ayarları şeklinde bırakılması veya hatalı olarak düzenlenmesi sonucunda ortaya çıkan tehditlerdir. Apache veya IIS web sunucu yazılımlarının genel ayarlarının organizasyona veya işletmeye göre değiştirilmemesi veya hatalı olarak düzenlenmesi örneği verilebilir.

7. **Cross-Site Scripting (XSS)**

   Kullanıcının web tarayıcısında JavaScript veya istemci taraflı script kodlarını çalıştırmaya imkân sağlayan tehditlerdir. Reflected, Stored ve DOM türleri bulunmaktadır.

8. **Insecure Deserialization**

   Güvensizlik Serileştirme, bir uygulamanın mantığını kötüye kullanmak, hizmet reddi (DoS: Denial of Service) saldırısı uygulamak veya seri hale getirildikten sonra isteğe bağlı bir kod yürütmek için güvenilmeyen veriler kullanıldığında ortaya çıkan bir güvenlik açığıdır.

9. **Using Components with Knows Vulnerabilities**

   Kullanılan yazılıma eklenmiş farklı kişiler tarafından yazılmış olan eklentilerden ortaya çıkan bir tehdit türüdür. Hazır içerik sistemleri (WordPress, Joomla vb.) içerisine eklenen farklı amaçlara yönelik eklentiler (Plug in) örnek olarak verilebilir.

10. **Insufficient Logging & Monitoring**

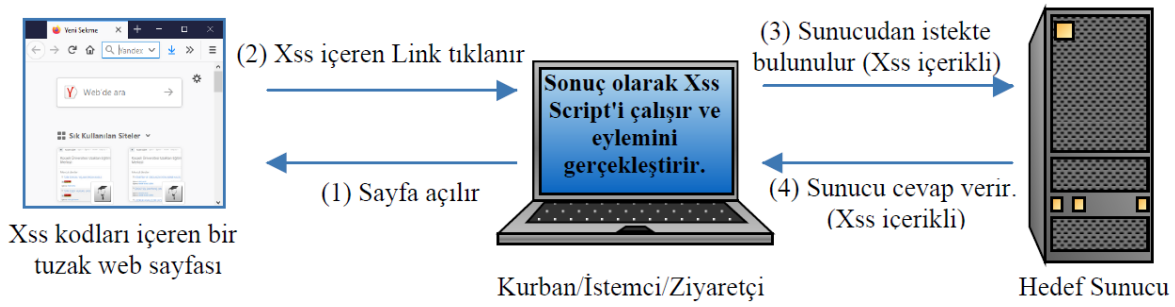
    Yetersiz Kayıt ve İzleme yukarıdaki risklerden biraz farklıdır. Doğrudan bir saldırıya yol açmasa da bu risk, saldırıya ait bilgileri zamanında tespit edemeyecektir. Atağa ait bilgiler ne kadar geç elde edilirse o kadar uzun süre sistem devre dışı kalacaktır. (URL 7)

www.internetlivestats.com (URL2) web sitesinin vermiş olduğu güncel verilere göre 2021 yılının sonlarına doğru göre tüm dünyada yaklaşık olarak 1,910,413,489 adet web sitesi bulunmaktadır. Bu siteler, faklı geliştiriciler tarafından çeşitli web teknolojileri ile oluşturulmuş sitelerdir. Bir kısmında güvenlik denetimleri oldukça yüksek iken bir kısmında çok daha düşük tutulmuş olabilir. Wordpress, Joomla, Drupal vb. Php (Personal Home Page) tabanlı içerik yönetim sistemlerinin yaygınlaşması ile az bir web teknolojisi bilgisine sahip kişiler bile web sayfalarını yönetmeye başladılar. Bu durum ortaya çok büyük bir güvenlik açığı meydana getirdi. Geniş bir kitle tarafından kullanılan, dinamik web sayfalarını geliştirme için tasarlanmış ve HTML içine gömülebilen bir script (Betik) dili olan Php ile yazılan bu içerik yönetim sistemleri, (CMS: Content Management System) Php'nin de güvenlik bakımından eksik yönleri ile birleşince kapıların ardına kadar açık olduğu korumasız web siteleri topluluğu olarak ortaya çıkmıştır. (URL 4)

4. XSS

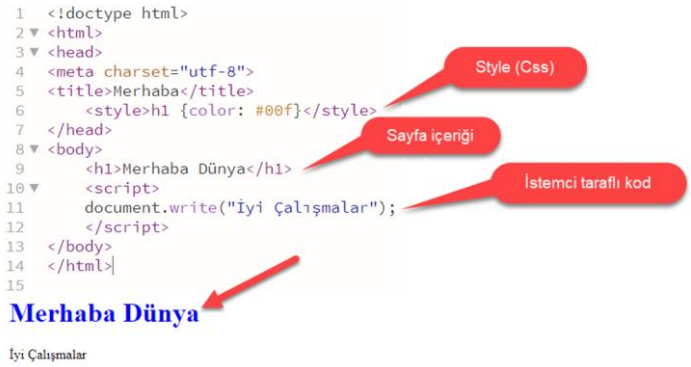
Kısaca XSS olan ve açılımı "Siteler Arası Komut Dosyası Yazma" (Cross Site Scripting), web uygulamalarının güvenliğini tehdit eden tehdit çeşitlerinden biridir. Genellikle JavaScript kodlarının web sayfalarına enjekte edilmesi ile ortaya çıkmaktadır. JavaScript, özellikle web tarayıcı yazılımlarının veya farklı uygulamaların yorumlayabildiği bir script (Betik) programlama dilidir. Web uygulaması

olarak en çok kullanılan web tarayıcılarında (Chrome, Mozilla, Edge, Opera, Safari, Internet Explorer) tehdidi ortadan kaldırmak için etkili ve pratik bir yol bulmak için araştırma ve geliştirmeler halen devam etmektedir.



Şekil 2: Xss'in Çalışma Mantığı

Şekil 2'de gösterildiği gibi herhangi bir ziyaretçi bir web sayfasını açar. İçindeki bir bağlantıyı veya adresi yükler. Bu sayfa, bir tuzak sayfasıdır. İstek sunucuya gönderilir. Sunucu bu isteğe cevap olarak sayfanın Html/Css/JavaScript şekli ile cevap verir. Sonuç olarak Xss kodları devreye girer ve amacını gerçekleştirmek için kodlarını yürütür. JavaScript kodlarının ağırlıkla kullanıldığı Xss ataklarını gerçekleştiren korsanlar; ziyaretçinin bilgisayarındaki çerez (Cookie) dosyaları aracılığı ile şifrelerini çalabilir, oturum (Session) bilgilerini elde edebilir, web sitesinin tahrif edilmesine neden olabilir, istemci bilgisayara solucan (Worm) yüklenmesine neden olabilir. Bunun yanında sayfayı farklı bir adrese yönlendirebilir, klavyeden girilen tuşları okuyabilir, bilgi giriş formları üzerinden veri gönderimi yapabilir. Çeşitli senaryolarla ziyaretçinin güvenliğini tehdit edebilir. Çerez dosyaları küçük çaplı (4 Kbyte) dosyalardır. Ziyaretçilerin daha önce gezdikleri sayfaya ait kullanıcı adı vb. bilgileri depolayan metin tabanlı dosyalardır. Çeşitli sitelerde ise ziyaretçiye özel daha kapsamlı bilgiler bulunmaktadır. Xss kodları ile bu bilgiler güvenliği delen kişiler tarafından çalınabilir. Bu yüzden siteler, çerez oluşturma durumunu kullanıcıdan onay alarak yapmaktadırlar.



Şekil 3: Web sayfası farklı web teknolojilerine ait kodları içerebilir

Xss, bir adres satırında bile başlayabilir.
http://www.siteadresi.com/mdunya.php?isim=<script>alert('Hack Edildin');</script>

Sunucudan istemciye gelen kodlar şöyle olacaktır:
<body>Merhaba <script>alert('Hack Edildin');</script></body>

Yukarıdaki gibi bir kaynak kod, istemcinin tarayıcısına gelirse bu kodlar, web tarayıcı tarafından çalıştırılacaktır.

Ses getirmiş Xss ataklarından bazıları şöyledir;

- Myspace Worm Samy (2004)

  Myspace dosya paylaşım sitesini hedef almıştır. "Samy is my Hero" mesajı ile ortaya çıkmıştır. Kendi profiline eklenen kodlar, kişinin diğer arkadaşlarının profiline de eklenerek yayılma sağlamıştır. 20 saatte 1.000.000 bulaşma gerçekleşmiştir.

- TweetDeck XSS Worm (2011)

  derGeruhn adlı bir kullanıcı, TweetDeck uygulamasında depolanmış bir XSS güvenlik açığından yararlanmış ve 82.136 Twitter kullanıcısını etkileyen ve belirli bir mesajı yeniden tweet yapmaya zorlayan bir solucan meydana getirmesini sağlamıştır.

4.1. Xss Çeşitleri

Anlık Xss (Reflected Xss): Anlık olarak, zararlı kod (Payload), bir internet adresine yerleştirilir. Hedef, zararlı kodun olduğu adresi ziyaret eder. Kalıcı değildir. Ziyaretçinin bir form giriş alanına veya bir internet adresine (URl: Uniform Resource Locator) istenilen verileri girmek yerine web tarayıcısının çalıştırabileceği JavaScript kodlarının girilmesi ile gerçekleşebilir.

Kayıtlı Xss (Stored Xss): Veri tabanlarında kayıtlı olan kalıcı zararlı kodlardır (Payload). Kalıcı bir Xss çeşididir. Fark edilmese sürekli çalışmaya açıktır. Her veri tabanından ilgili alan yüklendikçe Xss kodları çalışacaktır.

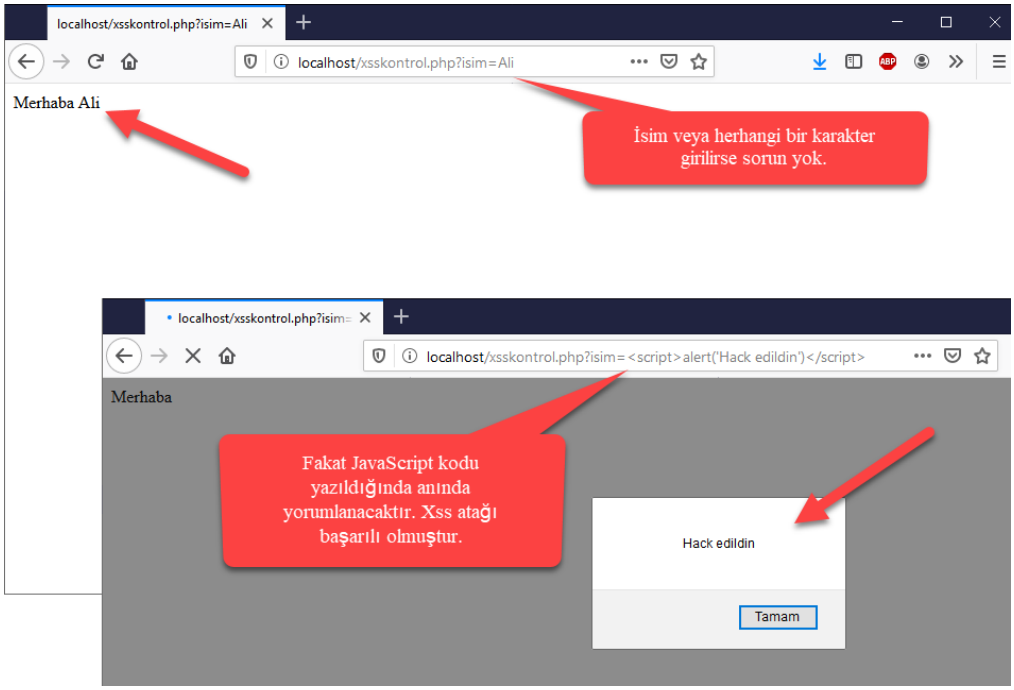
DOM tabanlı XSS (DOM Based Xss): DOM (Document Object Model) ortamının değişmesine bağlı olarak zararlı kod çalıştırılır. Kalıcı veya geçici olabilir. DOM, web sayfasını oluşturan HTML ile programlama dilleri arasında bir standart meydana getirip, HTML yapısından bilgi alışverişinde bulunmasına yardımcı olur. DOM, Nesneler ve özelliklerden meydana gelir. (Marashdih, 2017)

Aşağıda Php kodları kullanılarak oluşturulmuş bir Xss atağı gösterilmiştir.

xsskontrol.php

```
1  <!DOCTYPE html>
2  <html>
3  <meta charset="utf-8">
4  <body>
5
6  <?php
7  $isim=$_GET["isim"];
8  echo "Merhaba " . $isim;
9  ?>
10
11 </body>
12 </html>
```

Şekil 4: xsskontrol.php kodları



Şekil 5: Xss atağı örneği

Yukarıdaki örnekte 8. Satırdaki kodların yerine şu kodlar getirilirse Xss atağı çalışmayacaktır.

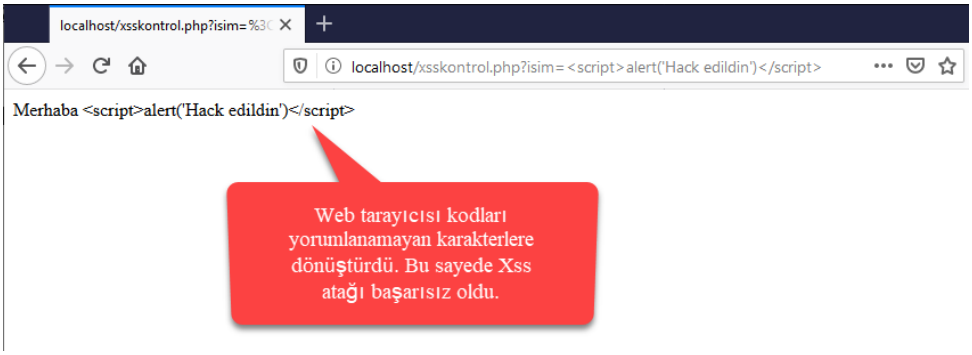
echo "Merhaba " . htmlentities($isim);

veya

echo "Merhaba " . htmlspecialchars($isim);

htmlentities ve htmlspecialchars Php fonksiyonu ile "<" ve ">" karakterleri sadece gösterilebilen karakterlere dönüşmüştür. İstemciye dönen kodlar şu şekildedir.

Merhaba **<**script**>**alert('Hack edildin')**<**/script**>**



Şekil 6: htmlentities fonksiyonu kullanılarak Xss atağı bertaraf edildi.

4.2. Xss Ataklarından Korunma Yolları

Tam anlamıyla bu ataklardan korunmak oldukça zordur. Çünkü açığı kapattıkça atağı yapanlar yeni bir açık arama peşinde olacaktır. Fakat etkili bazı korunma metotları şu şekilde sıralanabilir.

- İstemcilerin JavaScript yorumlamaları kapatılabilir. Böyle bir durumda sitelerin belirli bir kısmı JavaScript kodlarını çalıştırdığı için sayfalar tam istenildiği gibi çalışmayabilir. Kesin bir çözümdür fakat uygulanması zordur.
- Veri giriş alanlarının zararlı kod kontrolü yapılmayan sistemlerde bu tür kodları ayrıştıracak fonksiyonlar kullanılmalıdır. Örneğin Php için htmlentities, htmlspecialchars gibi fonksiyonlar kullanılabilir. Asp.Net için Server.HtmlEncode("<ifadeler>") fonksiyonu kullanılabilir.
- Web sayfası içerisindeki kodlar tekrar gözden geçirilmelidir. Ataklara imkân verebilecek kodlar filtrelenmelidir. Gereksiz veriler veri tabanlarına dahil edilmemelidir.
- Şirketin veya kurumun web programlama bölümünde çalışanların XSS vb. veri güvenliği eğitimi almaları sağlanmalıdır. Çünkü internet gibi bir okyanusa açılan her içerik korsan saldırısına maruz kalabilecek bir gemi pozisyonundadır.
- Bir şirket ağından İnternete erişim varsa güvenlik duvarı (Firewall) cihazı veya uygulaması kullanılmalıdır.
- Web içeriklerinde kullanılan karakterler, web tarayıcı tarafından yorumlanamayacak hale getirilmelidir. Örneğin "<" karakteri, < ve ">" karkateri, > karakterlerine dönüştürülmelidir.
- Bir enjeksiyon sadece çalışabilen bir alanda değil bir Css tasarım dosyasında da bulunabilir. Kodların dikkatli bir şekilde gözden geçirilmesi gereklidir. Örneğin: <h1 style="background:url('javascript:alert(1234)')">
- Kaynak kodların gözden geçirilme işlemi için güvenilir siteler kullanılmalıdır.

  OWASP HTML Sanitier Project

  https://owasp.org/www-project-java-html-sanitizer/

  Microsoft Web Protection Library

  System.Web.Security.AntiXss (Microsoft AntiXSS Library)

  https://www.owasp.org/index.php/.NET_AntiXSS_Library

- Kodları arıtan filtre kütüphaneleri kullanılmalıdır. Örneğin Php için HTMLPurifier, PHP ile yazılmış standartlara uygun bir HTML filtre kütüphanesidir. HTML Purifier (Arıtıcı), güvensiz ve izin verilmeyen Xss kodlarını kaldırmanın yanında aynı zamanda web belgelerinin web teknolojisi standartlara uygun olmasını da sağlar. Güvenlik derecesi önemli olan veri giriş alanlarında XSS saldırılarına karşı yüksek güvenlik sağlayan kimlik doğrulama kütüphanesi HTMLPurifier kullanılmalıdır.

Örnek HTML Purifier kullanımı:

www.htmlpurifier.org adresinden güncel kütüphane indirilir. Kitaplık dosyaları örnek Php dosyasının ulaşabileceği bir klasöre açılır. Aşağıda "Htmlpurifierdeneme.php" adlı örnek bir kullanım ve çalışmış hali bulunmaktadır.

Şekil 7: HtmlPurifier kitaplığının kullanımı ve çalıştırılması

5. SONUÇLAR

İnternet dünyasında haberli veya habersiz birçok veri başkalarının eline geçmektedir. Dikkat edilmediği taktirde bu veriler bir süre sonra kişilere zarar verebilecek boyutlara ulaşmaktadır. Siber güvenlik, gelecek nesillerde üzerinde önemle durulması gerekli olan bir konudur. Siber güvenlikle ilgili gerekli alt yapılar sağlanmamışsa ve eğitimler alınmamışsa büyük mali zararlara sebep olabilecek sonuçların meydana gelmesi her an beklenmelidir. Şirketlerin siber güvenlik üzerine uzmanlaşmış personellere ihtiyaçları her geçen gün artmaktadır. Ülkelerini eğitimden sorumlu olan bakanlıkları veya üniversiteleri siber güvenlik uzmanlarının yetiştirilmesi görevi işini yeteri kadar ciddiye almalıdırlar. Ülkemizde en azından üniversitelere bağlı meslek yüksekokullarında siber güvenlik teknikerleri yetiştirilmelidir. Bu çalışmada Xss ataklarının oluşma şekilleri incelenmiştir. Ataklara karşı alınması gerekli olan tedbirler izah edilmiştir. Daha güvenli bir internet dünyası için zararlı kodları arıtılmış ve filtrelenmiş web sayfalarına ihtiyaç vardır. Geliştiricilerin bu konuda yeteri kadar eğitim almaları gerekmektedir. Bu çalışma bize bunu da hatırlatma görevini yerine getirmiştir.

KAYNAKÇA

BAYRAKTAR, G. (2015), *Siber Savaş ve Ulusal Güvenlik Stratejisi*. İstanbul: Yeni Yüzyıl Yayınları.

A. W. MARASHDIH & Z. F. ZAABA, (2017) Cross Site Scripting: Removing Approaches in Web Application, *4th Information Systems International Conference 2017, ISICO 2017, 6-8*

URL1: https://owasp.org/Top10, E. Tarihi: 21.09.2021

URL 2: Total number of Websites, https://www.internetlivestats.com/total-number-of-websites/, E. Tarihi: 10.09.2021

URL 3: https://www.haberturk.com/ekonomi/teknoloji/haber/1463070-turkiyenin-siber-ordusu, E. Tarihi: 10.09.2021

URL 4: Php Nedir?, https://www.php.net/manual/tr/intro-whatis.php#intro-whatis, E. Tarihi: 11.10.2021

URL 5: Dünyanın en güvenli işletim sistemi, hacklendi! https://www.teknolojioku.com/guvenlik/dunyanin-en-guvenli-isletim-sistemi-hacklendi-5a28fbea18e540630d1df808, E. Tarihi: 10.10.2021

URL 6: 2017'de siber güvenlik için 86 milyar dolar harcandı, https://www.sigortacigazetesi.com.tr/2017 de-siber-guvenlik-icin-86-milyar-dolar-harcandi, E. Tarihi: 08.10.2021

URL 7: What is the Document Object Model? https://www.w3.org/TR/DOM-Level-2-Core/introduction, E. Tarihi: 11.10.2021

URL 8: 6 trilyon dolarlık siber kayıp, https://www.dunya.com/sektorler/teknoloji/6-trilyon-dolarlik-siber-kayip-haberi-628334, E. Tarihi: 09.10.2021