Otomatik Form Doldurma Saldırıları ve Bu Tür Saldırılara Karşı Güvenlik Önerileri

[1]Hüseyin Kutlu ve *[2]Engin Avcı ve ve *[3]Erkan Tanyıldızı *[4]Ali İhsan Çelik
[1]Besni Meslek Yüksekokulu, Bilgisayar Kullanımı Bölümü Adıyaman Üniversitesi, Türkiye
*[2]Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü Fırat Üniversitesi, Türkiye
*[3]Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü Fırat Üniversitesi, Türkiye
*[4]Besni Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü Adıyaman Üniversitesi, Türkiye

Özet

Bu çalışmada web sayfaları üzerinde sağ tıklayıp sayfa bilgilerini elde ederek yapılan web saldırıları hakkında bilgi verilmiştir. Söz konusu saldırıların nasıl önlenebileceği üzerine önerilerde bulunulmuştur. Bu tür saldırılarda ve saldırılara karşı alınan önlemlerde kullanılan CAPTCHA, Dom nesnesi, HtmlElement kavramlarına değinilmiştir. Çalışma kapsamında yapılan uyulamalarda Web sayfalarında otomatik form doldurma işlemi ile saldırı gerçekleştirme örneği gerçekleştirilmiştir. Bu tür saldırılara önlem olarak kullanılan CAPTCHA aşılmıştır. Bu tür saldırılara karşı alınması tavsiye edilen önlemlerden HTML kodları sayfaya kriptolu olarak gönderilmesi, Sayfa üzerinde sağ tık engellenmesi, Web sayfasına kullanıcı ve adı şifre ile erişim gibi önlemler uygulanmıştır. Bu tür saldırılara karşı çeşitli güvenlik önerileri sunulmuştur.

Anahtar Kelimeler: HtmlElement; DOM; Document Object Model; html şifreleme

Abstract

This study is related to web attacks which is made by viewing the page source by right-clicking on a web page. Suggestions were made on how to prevent these attacks. The concept has been referred to CAPTCHA, Dom Object, HtmlElement that used in the measures taken against in this type of attack.
At applications made under study Attack with automatic form filling process Web pages were performed. CAPTCHA is used as a measure of this type of attack is exceeded. Recommended that measures be taken against such attacks have been applied such as HTML code to be sent encrypted in page, prevent the right-click on the page to, accessing the web page with user name and password Several safety recommendations were presented against this type of attack.

Key words: HtmlElement; DOM; Document Object Model; html encription

1. Giriş

Bilişim teknolojilerin deki interaktif uygulamalar insan hayatını kolaylaştırdığı gibi, çeşitli istismarları da beraberinde getirmektedir. İnternette istemci yerine programlanmış robot (bot) saldırılarını engellemek ve istemcinin insan mı yoksa robot mu olduğunu test eden Carnegie Mellon School of Computer Science tarafından geliştirilen CAPTCHA, bot yazılımcıları

*İlgili Yazar: Hüseyin Kutlu, Adres: Besni Meslek Yüksekokulu, Bilgisayar Kullanımı Bölümü, Adıyaman Üniversitesi, Adıyaman Türkiye. E-mail: hkutlu@adiyaman.edu.tr, Phone: +904163110422 Fax: +904163110424

tarafından geliştirilen programlar veya OCR uygulamaları ile aşılmaktadır. Bununla birlikte dom (document object model) ile resim ve html nesnelerine erişilmektedir. Sayfa kaynak kodları bilgileriyle saldırı gerçekleştirecek yazılımcının yaptığı browser ile otomatik form doldurma işlemi gerçekleşmektedir. Bu yöntemleri kullanarak DOS ve DDOS saldırısı gerçekleştirmek web sayfasındaki resimlere otomatik erişen bir yazılım geliştirmek mümkündür. Bu çalışmada bu tür saldırıların nasıl gerçekleştirildiği uygulamalarla anlatılmış ve bu saldırılara karşı alınması gereken önlemler uygulamalar ile açıklanmıştır.
Bu çalışmanın amacı aslında bir resim olan ve istemcinin insan mı yoksa robot mu olduğunu test eden CAPTCHA resmine dom (document object model) kullanarak erişip resmi OCR uygulamaları ile okuyup sayfa kaynak kodlarından alınan bilgiler ile otomatik olarak sisteme giriş yapan veya bu yöntemle siteyi meşgul eden bir bot yazılımı geliştirilebilir ligini uygulamalarla göstermek ve bu tür saldırılara karşı alınabilecek yöntemleri uygulamalı olarak göstermektir.
Bu çalışma kapsamında CAPTCHA yöntemlerinden, OCR API'lerinden, dom (document object model) modelinden, sayfa kaynak kodlarından otomatik form doldurma teknikleri kısaca anlatılmıştır. Çalışma kapsamında yapılan örnek uygulama anlatılmıştır. CAPTCHA aşma ve otomatik form doldurma saldırılarına karşı alınacak önlemler kısaca anlatılmış ve kaynak kodlarıyla verilmiştir.

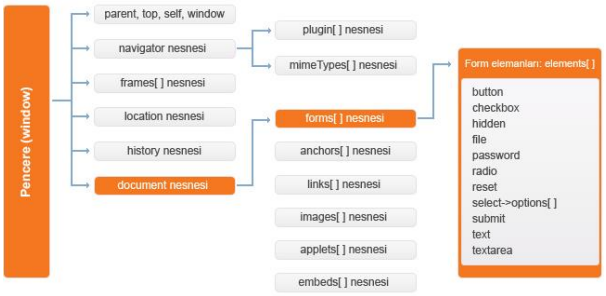
2. CAPTCHA ve CAPTCHA TÜRLERİ

İnternet kullanımının yaygınlaşması web sitelerinin çoğalması ile birlikte web yazılımlarının gelişmesine de neden olmuştur. Web Sitelerinin geniş kitlelerce kullanılması beraberinde güvenlik problemlerini doğurmuştur. Yazılım geliştiriciler güvenlik problemlerine çözüm üretmeye çalışırken kötü amaçlı yazılımcılar ise kodlar yardımıyla insanmış gibi davranan robot kullanıcılar(bot) oluşturarak şifre ile ilgili bütün ihtimalleri ve alternatifleri deneyerek kırma (Brute Force saldırısı), spam üretme, reklam ve zararlı link gibi istenmeyen paylaşımlar yapma, ağ trafiğini meşgul etme, sistemlere zarar verme ve benzeri amaçlar ile robot yazılımlar geliştirmişlerdir. Bu robot yazılımlara çözüm olarak istemcinin normal kullanıcı olan bir insan mı yoksa kötü amaçlı olan bir bilgisayarları mı olduğunu ayırt etmeye yarayan CAPTCHA ortaya çıkmıştır. Aslında bir resim olan CAPTCHA görüntü işleme algoritmalarıyla aşıldıkça farklı türleri gelişmiştir. Bu türlere Text tabanlı CAPTCHA, Re-CAPTCHA, Mat-CAPTCHA, Assira, KittenAuth, Motion CAPTCHA örnek olarak verilebilir. CAPTCHAlar amacına uygun geliştikçe, CAPTCHAları aşan robot yazılımlar görüntü işleme algoritmalarının hızla gelişmesiyle tüm CAPTCHAların üstesinden gelebilecek hale gelmişlerdir. CAPTCHA uygulaması kullanılmaz veya aşılabilirlerse rastgele kullanıcı adı ve şifre kombinasyonları deneyen bir robot yazılım DOM ve HtmlElement nesneleri yardımıyla sisteme giriş yapabilir.

3. DOM ve DOM Nesnesi

Web sitesi tasarımını bitirilip, html kodları web sayfası javascript açısından birer "document" nesnesi haline gelir.
Web sayfasındaki her bir nesneye "DOM" ara yüzü ile müdahale edebilir, istenilen değişiklikler yapabilir, istenilen olaylar ilgili nesnelere atanabilir.

DOM yapısını öğrenmek; javascript ve dolaylı olarak jQuery kullanımında olayın mantığını kavramak açısından çok yararlı olacaktır.
DOM ("Document Object Model"), HTML ve XML belgeler ("documents") için "Program Uygulama Arayüzü (API)"dir. Bu uygulama sayesinde HTML ve XML belgeler üzerinde istenilen yerde değişim, ekleme, silme gibi işlemler yapılabilmektedir.
Şekil 1'de javascript nesnelerinin birbirleriyle olan ilişkileri gösterilmektedir. En üst nesnemiz "window" nesnesidir. Örnek olarak "window->document->forms->elements[]" yapısı gösterilmiştir."



Şekil 1. Javascript nesnelerinin birbirleriyle olan ilişkileri

Bu yapıları bilindiğinde, web site sayfalarınızdaki müdahale etmeniz gereken nesnelere "zincirleme" yöntemiyle ulaşıp, istediğiniz düzenlemeler yapılabilir.
XSS Saldırısı (Cross Site Scripting Attack) diğer adı ile Çapraz Site Betik Saldırısı, Asp, PHP, ASP.NET gibi birçok web programlama dilinde meydana gelen, bir betik kod (HTML, JavaScript v.s) saldırısıdır. Saldırganın (Attacker) hedefi, web uygulamasına çeşitli kod betikleri yazarak XSS saldırısının çalışmasını sağlamaktır. XSS; HTML, JavaScript v.b betikler kullanarak gerçekleştirilir. Tüm web programlama dilleri çalışma anında (Run Time), son kullanıcı (End User)'ya HTML ile geri dönüş yapar. ASP.NET (.aspx) ile programlanmış olan bir web uygulaması, son kullanıcı tarafından çalıştırıldığında, web uygulaması HTML olarak görüntülenir. Bu özelliğinden dolayı HTML global bir görüntüleme dili haline gelmiştir. Yapılacak olan saldırı global HTML-JavaScript kullanılarak gerçekleştirilecektir.

DOM (Document Object Model – Belge Nesnesi Modeli): Internet tarayıcıları (Internet Explorer, Mozilla Firefox, Opera, Safari v.s) girmiş olduğumuz her web sayfasını bir belge olarak kabul eder. Girdiğimiz web sayfasında bulunan tüm materyaller (Image, buton, textbox v.s) ise bir nesne olarak kabul edilir. Örneğin web sitesinde bulunan bir resim, bir buton v.s birer nesnedir. DOM sayfada bulunan nesnelere müdahale etmemizi sağlar. Bunun için JavaScript gibi bazı script dillerinin kullanılması gerekir. Zararlı olarak kullanılabilecek DOM betikleri;

- HTML Yazma, Örnek:
  - document.write(…)
  - document.writeln(…)
  - document.body.innerHtml=…

- Doğrudan DOM Değiştirme (Dahili DHTML olayları), Örnek:
  - document.forms[0].action=… (ve çeşitli koleksiyonları)
  - document.attachEvent(…)
  - document.create…(…)

- document.execCommand(…)
- document.body. … (DOM ana nesnesi üzerinden erişen)
- window.attachEvent(…)

- Belge URL Değişimi, Örnek:

- document.location=… (yer, alan ve host atama)
- document.location.hostname=…
- document.location.replace(…)
- document.location.assign(…)
- document.URL=…
- window.navigate(…)

- Pencere Açma Değiştirme, Örnek:
- document.open(…)
- window.open(…)
- window.location.href=… (yer, alan ve host atama)

- Doğrudan Script Çalıştırma, Örnek:
- eval(…)
- window.execScript(…)
- window.setInterval(…)
- window.setTimeout(…)

4. Html Element Nesnesi

Web browserınızda açılan sayfada, istediğiniz html elemanını yönetme imkanını veren HtmlElement classıdır. Örneğin web browserınızdan açılan bir forma değer göndermek isteniyorsa veya yine browserınızda açılan bir sayfada, resimler çekilmek isteniyorsa bu işlemleri HtmlElement classı yapabilir.
Kullanış amaçları ve yerleri çoktur. Örneğin bir sms programı yapılacak ise GSM operatörünün Web SMS özelliğini desteklendiğini var sayarak. Kısa bir kod yazarak, SMS atılacak sayfaya textboxlarımızdan girilen değerleri gönderip, formu submit ettirilebilir. Yine örnek verilecek olursa. Bir sayfadan sürekli sorgulama yaparak istenilen veriler alınmak isteniyorsa (web sitenin rss desteği ve webserviceleri yoksa) HtmlElement classı oluşturulur.
Web browserınızdaki elemanları DOM teknolojisi ile istediğinizi yaptırabilirsiniz veya verileri alabilirsiniz.

```
[ad#icerik-ad]
foreach (HtmlElement item in webBrowser1.Document.Links)
{
MessageBox.Show(item.GetAttribute("href").ToString());
} // Bu kodla sayfadaki tüm linkleri alıp message box ile ekrana uyarı verdirebiliriz. GetAttribute komutu javascript kodudur. Gelen elemanın Attribute'unu almayı sağlar. linkin hrefi verilen örnekle alınmıştır.
webBrowser1.Document.GetElementById("q").InnerText = "Hüseyin KUTLU";
webBrowser1.Document.Forms[0].InvokeMember("submit");
```

Webbrowserımızda google açık olsun. bu komut çalıştığında q elemanın textine(valueside olabilir burada value) "Hüseyin KUTLU" stringini yerleştirir sonra sayfanın ilk formunu submit eder.

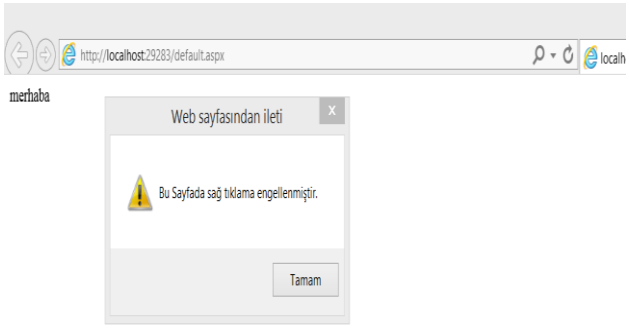
5. Proje Kapsamında Yapılan Uygulama

Bu çalışmada aslında bir resim olan ve istemcinin insan mı yoksa robot mu olduğunu test eden CAPTCHA resmine dom (document object model) kullanarak erişilmiştir. Elde edilen CAPTCHA resmi bilgisayara kaydedilmiştir. Kaydedilen CAPTCHA resim OCR uygulamaları ile okunmuş ve bir değişkene atanmıştır. Sayfa kaynak kodlarından alınan bilgiler ile kontrollere erişilmiştir ve ilgili kontrollere ilgili değerler atanmıştır. HtmlElement classı ile submit butonu tıklanmıştır. Böylelikle otomatik olarak sisteme giriş yapan veya bu yöntemle siteyi meşgul eden bir bot yazılımı geliştirilebilir ligini gösterilmiştir.

Çalışmayı gerçekleştirmek amacıyla öncelikle Visual Studio 2010 ortamında VB dilinde bir web browser tasarlanmıştır. Tasarlanan Web browserda görüntülen sayfada CAPTCHA resmine DOM nesnesiyle erişimiştir. CAPTCHA nesnesinin sayfadaki index değeri bulunmuştur. CAPTCHA resmi bilgisayara kaydedilmiş ve Web browserdaki bir yordamla okunup bir değişkene atanmıştır. Sayfa kaynağından alınan ID değerleri ile ilgili kontrollere Htmlelement yardımı ile değer atayan bir fonksiyon yazılmıştır. Htmlelement yardımı ile sisteme login olunmuştur. CAPTCHA yanlış okunması durumunda yazılım yeniden başlatılarak ziyaret edilen site meşgul edilmiştir.

Çalışma kapsamında yapılan çalışma Web güvenliğinde Captcha kullanılıyorsa iyi bir OCR API'si ile aşılabileceğini ve DOM ve HtmlElement nesneleri yarımı ile formların otomatik olarak doldurulabileceğini göstermiştir. Yapılan uygulamaya benzer bot yazılımlarıyla bütün ihtimalleri ve alternatifleri deneyerek kırma (Brute Force saldırısı), DOS, DDOS saldırıları, spam üretme, şifre bloke etme, reklam ve zararlı link gibi istenmeyen paylaşımlar yapma, ağ trafiğini meşgul etme, sistemlere zarar verme ve benzeri saldırılar gerçekleştirilebilir. Bu ihtimaller sonucu bir web sitesine Bu ihtimal dahilinde yapılan öneriler aşağıdaki gibidir.

- Kullanılan CAPTCHA kolay okunmamalıdır.
- CAPTCHA girişi bir sayaç yardımıyla sayılarak bot yazılımı olduğu anlaşılabilir ve web sayfası belirli aralıklarla erişim engelli hale getirilebilir.
- Sayfa üzerinde mause sağ tuşu ile ulaşılan kaynağı görüntüle sekmesi <body oncontextmenu="return false;"> komutu ile kilitlenebilir. Aşağıdaki kodlarla farklı bir yötemle mause sağ tık menüsü kapatılmıştır.



Şekil 2. Sayfada Sağ Tık Engelleme

SAYFADA SAĞ TIK ENGELLEME SCRİPT KODLARI

```
<script type="text/javascript">
        var msg = "Bu Sayfada sağ tıklama engellenmiştir.";
      function clickIE() { if (document.all) { alert(msg); return false; } }
      function clickNS(e) {
        if
(document.layers || (document.getElementById && !document.all)) {
          if (e.which == 2 || e.which == 3) { alert(msg); return false; }
        }
      }
      if (document.layers) {
        document.captureEvents(Event.MOUSEDOWN);
        document.onmousedown = clickNS;
      }
      else { document.onmouseup = clickNS; document.oncontextmenu = clickIE; }
      document.oncontextmenu = new Function("return false")
</script>
```

- HTML komutları kriptolanabilir.
- DOM nesnesinin ve HtmlElement'in çalışmaması amacıyla HTML kodlarınız kripolanabilir. Bu amaçla Javascript kodlarıyla aşağıda bir uygulama gerçekleştirilmiştir.

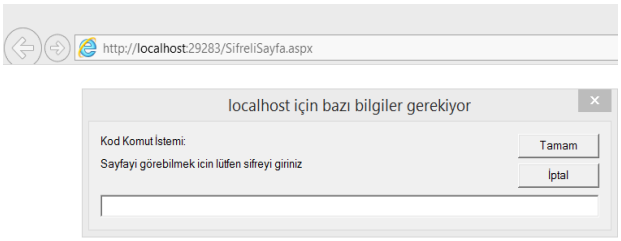
Şekil 3. HTML kodlarının kriptolanması

SAYFA KODLARINI ŞİFREYEN SCRİPT KODLARI

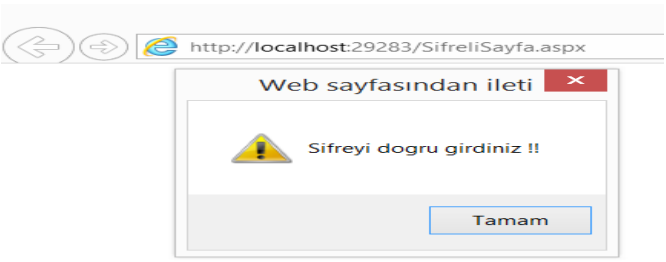
```
<script type="text/javascript">
      function encodestr() {
        var s = document.getElementById('HTMLCodeTB').value;
        alert(s);
       var k = "23";
```

```
            var sl = s.length;
            var kl = k.length;
            for (encodestr = '', i = 0; i < sl; i++)
            {
               var encodedChar = s.charCodeAt(i) ^ k.charCodeAt(i % kl);
               encodestr += String.fromCharCode((encodedChar & 0x0F) + 97) +
String.fromCharCode((encodedChar >> 4) + 97);
            }
            //document.getElementById('sifre').innerHTML = encodestr;
            //document.write(encodestr);
            //decodestr(encodestr, "23");
            document.getElementById('SonCodeTB').value = encodestr;
         }
   </script>
   <script type="text/javascript">
      function decodestr()
       {
         var s = document.getElementById('SonCodeTB').value;
         var k = "23";
         var sl = s.length;
         var kl = k.length;
         for (decodestr = '', i = 0, j = 0; i < sl; i += 2, j++) {
            decodestr += String.fromCharCode(((s.charCodeAt(i) - 97) + ((s.charCodeAt(i + 1) - 97) << 4)) ^
k.charCodeAt(j % kl));
         }
         document.getElementById('EDHTB').value = decodestr;
         return decodestr; }</script>
```

- Sayfaya erişim kullanıcı adı ve şifre ile gerçekleştirilir

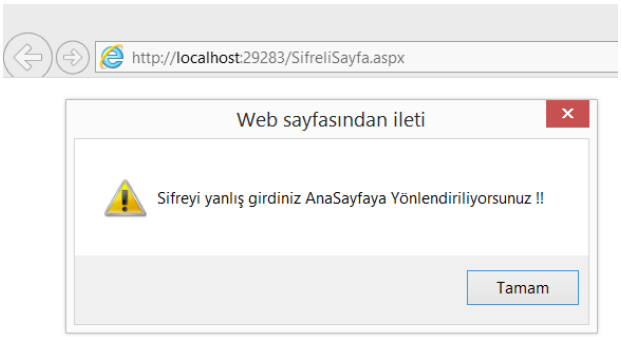


a) Kullanıcıdan Şifre İstemi



b) Kullanıcı Şifreyi Doğru Girdiyse Görünecek Uyarı

c) Kullanıcının Şifreyi Doğru GirmesiHalinde Yönlendirileceği Sayfa



d) Kullanıcı Şifreyi Yanlış Girdiyse Görünecek Uyarı

Şekil 3. Sayfa Erişimine Şifre Koyma

SAYFA KODLARINA ERİŞİMİ ENGELLEYEN SCRİPT KODLAR

```
<script>
   var password;
   var pass1 = "htmlpaylas";
   password = prompt('Sayfayi görebilmek icin lütfen sifreyi giriniz', '');
   if (password == pass1) { alert('Sifreyi dogru girdiniz !!'); }
   else { window.location = ""; alert('Sifreyi yanlış girdiniz AnaSayfaya Yönlendiriliyorsunuz !!'); }
   //-->
</script>
```

- Web sayfası Flash gibi resim tabanlı olarak tasarlanabilir
- Kopyalanmamasını istediğiniz veriler ajax ile çağırılabilir

Kaynaklar

[1] Joss Crow "http://www.josscrowcroft.com/projects/motioncaptcha-jquery-plugin"
[2] https://www.bilgiguvenligi.gov.tr/teknik-yazilar-kategorisi/captcha-kullanimi.html
[3] http://umutzafer.wordpress.com/2011/12/08/captcha-nedir-cesitleri-nelerdir-nerelerde-kullanilir/
[4] http://www.teakolik.com/captcha-nedir-kittenauth-nedir-ne-ise-yarar/
[5] http://makaleci.com/php-guvenlik-kodu-uygulamasi-captcha.html
[6] http://www.teknotrik.com/hack-1-brute-force-saldirisi.html
[7] http://www.captcha.net/
[8] http://www.csharpnedir.com/articles/read/?id=1129
[9] http://support.google.com/a/bin/answer.py?hl=tr&answer=1217728
[10] Çelikbilek,İ.,2012 Geleceğin web standartları HTML5.Kodlab Yayıncılık,457s.,İstanbul
[11] Kutlu H, Avci E,"Görüntü İşleme Teknikleriyle CAPTCHA aşma " ,ISDFS 2013 ,ELAZIĞ
[12] http://osmankurt.net/post/C-da-Webbrowser-Kullanc4b1mc4b1-Textboxc4b1-Otomatik-Doldurma-Ve-Butona-otomatik-Basma.aspx
[13] http://cybercomo.50megs.com/html_Encryption.htm
[14] http://www.pinusart.com/yazilar-34-dom-document-object-model-nedir.html
[15] http://msdn.microsoft.com/en-us/library/hf9hbf87(v=vs.110).aspx
[16] http://wmaraci.com/forum/araclar/html-kod-sifreleyici-28.html
[17] http://www.sifrex.com/web-site-araclari/html-kod-sifreleme