ASP.Net Uygulamalarında OASIS WAS EVDL Formatı Kullanan Uygulama Ateş Duvarı Gerçekleştirilmesi (DefApp v0.63)

Izzet Kerem KÜSMEZER[1]

[1]OWASP.org Türkiye Chapter Lideri

[1]e-posta: kerem.kusmezer@owasp.org

Anahtar sözcükler:Ateş Duvarı,Uygulama Güvenliği, ASP.Net, HttpModule, OWASP

ABSTRACT

This paper presents a new multithread Application Level Firewall, which implements the EVDL Protect Component Format (Purposed By The OASIS WAS (OASIS (Organization for the Advancement of Structured Information Standards Web Application Security*) Committee) as the attack signature format.*

The proposed Firewall gives the Administrators the ability to control the behavior of the validation and attack signature detection and more control then Microsoft's validate request ımplementatıon, whıch is designed by Microsoft for the ASP.Net Application since v1.1. It allows the Administrators to close the hole of the applications just using configuration files and without need of the source code of the running application. The solution is implemented as an Http Module.

The Project is a LGPL Licensed Project and supported by the Owasp.org (Open Web Application Security Project).

1. GİRİŞ

Günümüzde web uygulamalarındaki en önemli problem gerek proje tasarım aşamasında gerek proje teklif süreleri gerekse proje fiyatlandırmalarının düşürülmesinin sağlanması için , yazılım tasarımında güvenlik tasarımı ciddi anlamda gözardı edilmektedir.

Bu bağlamda geliştirilmiş olan uygulamalarda çok ciddi yazılım güvenliği ve doğrulama hataları bulunmaktadır. Bu açıkların kısaca bir özeti Tablo 1.1de verilmiştir. ASP.Net bu tarz saldırılara karşı otomatik CSS (Cross Site Scripting) Engelleme motoru içermektedir. Ama bu motorda aşağıda listenmiş olan güvenlik açıklarını içermekte ve kullanıcı tarafından filtrasyon tanımları konfigüre edilememektedir. Bu da ürünün kullanımını ciddi anlamda kısıtlamaktadır. DefApp bu sorunu gidermek ve OASIS WAS (Projesi tarafından geliştirilmiş olan EVDL formatının .Net üzerinde gerçekleştirilmesini sağlamak için C# kullanılarak geliştirilmiştir.

DefApp ASP.Net tarafından sunulan IHttpModule arayüzünün(interface'in) implemente edilmesi sonucunda , ASP.Net tarafından oluşturulan aşağıda listesi verilen olaylar yakalanarak (Tablo 1.2), uygulamanın girdileri üzerinde gerekli geçerlik denetimleri yapılmakta, uygulamanın bu istek sonucunda hata oluşturması durumunda bu hatalar yakalanarak bir günlük ortamına yazılmakta, ve kurallarda belirtildiyse istekte bulunan kişinin belirli bir eşik değerini aşması durumunda otomatik olarak IIS (İnternet Information Server) üzerinde bir kural yaratarak İnternet İletişim Kuralı (İnternet Protokol) bazlı erişimi kapatılmaktadır. Uygulama şu anda detayları aşağıdaki tabloda (Tablo 1.3) verilmiş olan standart tanımlı kural yapılarını desteklemektedir. Kural yapılarının başka geliştiricilerin ihtiyaçları doğrultusunda geliştirilebilmesi açısından geliştirilmiş bir Uygulama Programlama Arayüzü (API) ile birlikte sunulmaktadır.

2. Yapılandırma Dosya Formatı

Yapılandırma Dosyası ASP.Net Uygulamalarının web.config dosyasına eklenir ve DefenceMainSettingHandler isimli bir SettingHandler implementasyonu tarafından işlenerek, uygulama bu doğrultuda konfigüre edilir. Bu kısımlar kısaca aşağıdaki bölümlerden oluşmaktadır.

a)Handler(İşleyici Bölümü)

```
<Handlers>
<HandleCookies Action="1"/>
<HandleFormFields Action="1"/>
<HandleQueryString Action="1"/>
</Handlers>
```

Yukarıda örnek olarak verilmiş olan bölümde istek içinde gelen hangi parametrelerin kontrol edileceği belirtilmektedir. Bu WAS EVDL Formatından bağımsız olarak uygulamanın kendi Filtrasyon motoru kullanımı durumlarında geçerli olmaktadır.

b) Kural Listesi(RuleList)

```
<RuleList>
<rule name="textchecks1" Action="deny"
Type="Textrule"
PlugInAlias="">&lt;SCRIPT&gt;</rule>
</RuleList>
```

Yukarıda verilmiş olan rulelistte tanımlanmış kural ile içinde <script> tagi bulunan bütün requestlerin uygulamaya ulaşmadan tespit edilerek durdurulması sağlanmıştır. Her kural listesi EVDL Formatı haricinde istediğiniz sayıda kural içerebilmektedir. Bu kuralların kontrolleri 've' mantıksal değimi kullanılarak değerlendirilmektedir.Plugin Alias tanımı yapılmış ise verilen filtre tanımı o isimle yüklenmiş eklentilerin içinde aranır.

c) Günlük Ayarları

```
<Activation Debug="true"
Active="true"
DebugFile="c:\\testdebug.log"/>
```

Günlük Oluşturulmasında Log4J Uygulamasının .Net Platformuna çevrilmiş hali olan Log4Net kullanılmıştır. Loglar yukarıda belirtilen konfigürasyon dosyası ayarları yardımıyla bir metin dosyasına yazdırılabilmektedirler. Günlük Dosyası yapılan isteklerin içeriklerini, istek başlık ve içerik bilgilerini içermekte, ve hangi kural tarafından ret edildiğiyle birlikte , istekte bulunan kişinin ip adresi gibi bilgilerini de içermektedir. (Tablo 1.4) Buradaki Debug Parametresi günlük oluşturulmasının açılmasını, Active Parametresi DefApp Uygulamasının web uygulaması yüklenirken aktive edilmesi, debugfile parametresi ise oluşturulan günlük içeriğinin belirtilen isimli dosyaya yazılmasını sağlamaktadır.

d) Eklenti ve Genişletme Ayarları

```
<Plugins>
<plugin pluginalias="" pluginname=""
assembly="" />
</Plugins>
```

Eklenti tanımlamaları yukarıdaki parametreler aracılığıyla yapılmaktadır. Burada PlugInAlias kural seçimlerinde bu eklenti kütüphanesinin kural tanımında burada belirtilmiş olan assemblynin içinden okunması sağlanır. Plugin Name bu assemblynin içinde yüklenecek Plugin Setinin ismini belirtmektedir. Burada belirtilmiş olan assembly reflection kullanılarak yüklenip , içerisindeki DefAppPlugin(Tablo 1.5) sınıfından türetilmiş sınıflar bulunarak, otomatik olarak gerekli eklenti sınıfları oluşturulmaktadır.

e) FoundStone Module Ayarları

```
<FoundStoneModule XMLRulesDatabase=""
ValidatorFormMappings="" ValidatorRules=""
Active="true" />
```

Bu ayar bölümü FoundStone Firması tarafından geliştirilmiş ve geliştiricisi Dinis Cruz ve benim tarafımdan DefApp uygulamasını eklenmiş olan Validasyon Motorunun Konfigürasyonun Yapılması için gerekli ayarları içermektedir. Detaylı Konfigürasyon Formatı www.foundstone.com adresinden Validator.Net uygulamasının dokümantasyonunda bulunmaktadır.

f) Henüz İmplementasyon Aşamasında Olan Özellikler

a)EVDL-0.5-Protect Schema Validasyonu
b)EVDL Standardında tanımlanmış olan aşağıdaki istek normalizasyon tipleri.
c)EVDL Formatındaki Rule Tipleri
d)EVDL Formatındaki RuleSet Tanımlamaları
Bu kısımların detaylı bilgisini OWASP WAS Commitesinden bulabilirsiniz. Bu formatların iç implemantasyonları bitirilmiş olup konfigürasyon modüleri tamamlanacaktır. Bu spesifikasyonu http://www.yazilimguvenligi.com adresinden indirebilirsiniz.

3. UYGULAMANIN ÇALIŞTIRILMASI

Uygulama yukarıda detaylarıyla anlatılan konfigürasyon parametrelerinin web.Config dosyasına (Tablo 1.7) şekilde eklenmesi sonucunda ve DefAppModule'une ait dll dosyasının GAC veya web uygulamasının bin/ klasörüne konularak çalışır hale getirilir. Uygulama Debug modunda çalıştırıldığında konfigürasyon dosyalarının okunması , gerekli kuralların yaratılmasıyla ilgili olan bütün işlem adımları belirtilmiş olan günlük dosyası üzerinden takip edilebilmekte , saldırganca kullanılmış bütün saldırı desenleri izlenebilmektedir. Aynı zamanda uygulama logları gerçek zamanlı olarak defapp.aspx sayfasının çağrılmasıyla web üzerinden de takip edilebilmektedir. DefApp.aspx dosyası şu anda şifresiz olarak istenebilmektedir, ama bunun yetkilendirme yönetimiyle ilgili geliştirme bitirilmiş olup, bir dahaki sürümde aktive edilecektir.Şayet isteklerin içinde kurallar tarafından onaylanmayan bir desen ile karşılaşılırsa, uygulamanın çalışması durdurularak kullanıcıya bir hata mesajı dönülmektedir.

Şekiller ve Tablolar:

Olay Adı	Yapılan İşlem	Olay İşleyici Metod ve Açıklaması
BeginRequest	İstemci Tarafından Gönderilen İsteğin ASP.Net Uygulamasına Ulaşmadan Önce Tanımlanmış Olan Filtrasyon Metodlarınca Denetimden Geçirildiği Bölüm.	Main.MainHandlingSection
PreSendRequestHeaders	İstemci Tarafından Gönderilen İsteğe ASP.Net Uygulaması Tarafından Gerekli İçerik Oluşturulduktan Sonra HTTP (Hiper Metin Aktarım İletişim Kuralı) Başlık Bilgileri Gönderilmeden Çalıştırılan Bölüm	Main.HeaderHandlingSection
PreSendRequestContent	İstemci Tarafından Gönderilen İsteğe ASP.Net Uygulaması Tarafından Gerekli İçerik Oluşturulduktan Sonra HTTP (Hiper Metin Aktarım İletişim Kuralı) İçerik Bilgileri Gönderilmeden Çalıştırılan Bölüm	Main.ContentHandlingSection
EndRequest	İstemci Tarafından Gönderilen İsteğe ASP.Net Uygulaması Tarafından Gerekli İçerik Oluşturulduktan Sonra HTTP (Hiper Metin Aktarım İletişim Kuralı) İstekle İlgili İşlemlerin Bitirildiği Bölüm.	Main.EndHandlingSection
Error	Uygulamanın Çalışması Esnasında Kural Dışılık İşlemesi Yapılmamış Durumların Yakalandığı Modül	Main.ErrorHandlingSection

Tablo 1.2.Yakalanan Olaylar

Güvenlik Açığı İsmi	Güvenlik Açığı Açıklaması	DefApp Tarafından Sağlanan Korunma Metodu
Teyit Edilmemiş Girdi Enjeksiyon Kusurları	Kullanıcıdan gelen talebin, web uygulamasında kullanılmadan önce girdi olarak kabul edilmeye uygunluğu kontrol edilmez. Saldırganlar bu tanımlanmamış girdi açıklarını kullanarak web uygulaması üzerinden arka plandaki bileşenlere saldırabilirler. Web uygulamaları dış sistemlere veya yerel işletim sistemine eriştiklerinde, değişkenleri kabul ederler. Eğer bir saldırgan zararlı komutlarını bu değişkenlerin içine saklayabilirse, dış sistem bu komutları web uygulaması adına çalıştırır.	Regular Expression Kuralı ile tanımlanmış, aynı zamanda Metin Kuralı ile tanımlanmış kurallarla gerekli koruma BeginRequest olayında kontrol edilebilir. Aynı zamanda içerdiği FoundStone HttpModule implemantasyonuyla webform bazında da input validasyon tanımlanabilir.
Çiğnenmiş Doğrulama ve Oturum Yönetimi	Hesap güven belgeleri ve oturum "token"ları doğru korunmaz. Şifreler, anahtarlar, oturum kurabiyeleri ve diğer "token"larla oynayabilen saldırganlar doğrulama kusurlarını aşabilir ve başka kullanıcıların kimliklerine bürünebilirler.	BegınRequest olayında ve PreSendRequestHeaders methodlarında cookieler ve başlık bilgileriyle ilgili gerekli denetimler yapılmaktadır.
Çapraz Site Sorgu (XSS) Açıkları	Web uygulamaları bir saldırıyı son kullanıcıya browser'ı vasıtasıyla iletmek için bir araç olarak kullanılabilir. Başarılı bir saldırı son kullanıcının oturum izini afişe edebilir, yerel makineye saldırabilir veya kullanıcıyı kandırmak için içeriği taklit edebilir.	Çapraz Site Sorguları DefApp Tarafından otomatik olarak BeginRequest olayında yakalanmaktadır. Bu esnada gelen istek üzerinde ASP.Net uygulamasında tanımlı codepage'e bağlı olarak yüksek UTF-8 karakterleri eş değer normal karakter değerlerine çevrilerek, ASP.Net 1.1 deki doğrulama motorunun içerdiği bir hatada giderilmektedir.
Uygunsuz Hata İşleme	Normal koşullarda oluşan hata durumları doğru ele alınmaz. Eğer saldırgan web uygulamasının başa çıkamayacağı hatalara neden olabilirse, detaylı sistem bilgisi elde edebilir, servisi engelleyebilir, güvenlik mekanizmalarının başarısız olmasını sağlayabilir veya sunucuyu çökertebilir.	Uygulama tarafından implemente edilmiş olan hata işleme modülü hata mesajlarının son kullanıcıya ulaşmasını engellemekte, ve bilinçli olarak belirli hataların sürekli oluşturulması halinde saldırganın uygulamaya erişimi engellenmektedir.
Servis Engelleme	Web uygulamaları bilgileri ve güven belgelerini korumak için, sıkça kriptografik fonksiyonlar kullanırlar. Bu fonksiyonlar ve entegre etmek için kullanılan kodun zorluğu, sık sık düşük seviye koruma ile sonuçlanır.	Uygulama tarafından implemente edilmiş olan hata işleme modülü hata mesajlarının son kullanıcıya ulaşmasını engellemektedir. Uygulama bir adresten normalin üzerinde istek aldığında saldırganın uygulamaya erişimi engellenmektedir.

Tablo 1.1.Yazılım Güvenliği Güvenlik Açıkları

Kural İsmi	Kural Özellikleri
Textrule (Metin Kuralı)	İsteğin içinde belirtilmiş objelerde tanımlanmış olan metni arar ve uygun olarak belirtilmiş aksiyonu gerçekleştirir.
RegExRule(Regular Expression Kuralı)	İsteğin içinde belirtilmiş Regular Expression desenini arar ve uygun olarak belirtilmiş aksiyonu gerçekleştirir.
Ntextrule(Metin Kuralı)	İsteğin içinde belirtilmiş objelerde tanımlanmamış olan metni arar ve uygun olarak belirtilmiş aksiyonu gerçekleştirir.
LtRule(Küçüktür Kuralı)	İsteğin içinde belirtilmiş objelerde tanımlanmamış olan degerden küçük değeri arar ve uygun olarak belirtilmiş aksiyonu gerçekleştirir.
GtRule(Büyüktür Kuralı)	İsteğin içinde belirtilmiş objelerde tanımlanmamış olan degerden büzük değeri arar ve uygun olarak belirtilmiş aksiyonu gerçekleştirir.

Tablo 1.3.Standart Tanımlanmış Kurallar

120 [2840] DEBUG Owasp.DefApp.Utility.GeneralUtilities - HttpMethod------>POST
RequestType------>POST
ContentType------>application/x-www-form-urlencoded
ContentLength------>127
ContentEncoding------>System.Text.CodePageEncoding
AcceptTypes------>System.String[]
IsAuthenticated------>False
IsSecureConnection------>False
Path------>/DefAppTestWeb/testforms.aspx
FilePath------>/DefAppTestWeb/testforms.aspx
CurrentExecutionFilePath------>/DefAppTestWeb/testforms.aspx
PathInfo------>
PhysicalPath------>C:\Inetpub\wwwroot\DefAppTestWeb\testforms.aspx
ApplicationPath------>/DefAppTestWeb
PhysicalApplicationPath------>C:\Inetpub\wwwroot\DefAppTestWeb\
UserAgent------>Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322)
UserLanguages------>System.String[]
Browser------>System.Web.Mobile.MobileCapabilities
UserHostName------>127.0.0.1
UserHostAddress------>127.0.0.1
RawUrl------>/DefAppTestWeb/testforms.aspx
Url------>http://localhost/DefAppTestWeb/testforms.aspx
UrlReferrer------>http://localhost/DefAppTestWeb/testforms.aspx
QueryString------>
Form------>__VIEWSTATE=dDwtMzg4MDA0NzA7Oz7HvuPk%2fv8ojwi6jQf2vq%2fL6yJFqA%3d%3d&TextBox1=%3cscript
Headers------>Cache-Control=no-cache&Connection=Keep-Alive&Content-Length=127&Content-Type=application%2fx-www-f
excel%2c+application%2fmsword%2c+*%2f*&Accept-Encoding=gzip%2c+deflate&Accept-Language=en-us%2ctr%3bq%3d0.5
Agent=Mozilla%2f4.0+(compatible%3b+MSIE+6.0%3b+Windows+NT+5.1%3b+.NET+CLR+1.1.4322)
ServerVariables------>ALL_HTTP=HTTP_CACHE_CONTROL%3ano-cache%0d%0aHTTP_CONNECTION%3aKeep-Alive%0
shockwave-flash%2c+application%2fvnd.ms-powerpoint%2c+application%2fvnd.ms-excel%2c+application%2fmsword%2c+*%2
us%2ctr%3bq%3d0.5%0d%0aHTTP_COOKIE%3aASP.NET_SessionId%3d42zrl455iqwzrfye4q3r0r45%0d%0aHTTP_HOST%3
AW=Cache-Control%3a+no-cache%0d%0aConnection%3a+Keep-Alive%0d%0aContent-Length%3a+127%0d%0aContent-Type
excel%2c+application%2fmsword%2c+*%2f*%0d%0aAccept-Encoding%3a+gzip%2c+deflate%0d%0aAccept-Language%3a+en
Agent%3a+Mozilla%2f4.0+(compatible%3b+MSIE+6.0%3b+Windows+NT+5.1%3b+.NET+CLR+1.1.4322)%0d%0a&APPL_M
T_COOKIE=&CERT_FLAGS=&CERT_ISSUER=&CERT_KEYSIZE=&CERT_SECRETKEYSIZE=&CERT_SERIALNUMBE
urlencoded&GATEWAY_INTERFACE=CGI%2f1.1&HTTPS=off&HTTPS_KEYSIZE=&HTTPS_SECRETKEYSIZE=&HTTPS
tpub%5cwwwroot%5cDefAppTestWeb%5ctestforms.aspx&QUERY_STRING=&REMOTE_ADDR=127.0.0.1&REMOTE_HOS
ERVER_SOFTWARE=Microsoft-IIS%2f5.0&URL=%2fDefAppTestWeb%2ftestforms.aspx&HTTP_CACHE_CONTROL=no-ca
xbitmap%2c+image%2fjpeg%2c+image%2fpjpeg%2c+application%2fx-shockwave-flash%2c+application%2fvnd.ms-powerpoint
us%2ctr%3bq%3d0.5&HTTP_COOKIE=ASP.NET_SessionId%3d42zrl455iqwzrfye4q3r0r45&HTTP_HOST=localhost&HTTP_
Cookies------>System.Web.HttpCookieCollection

```
TotalBytes------>127
120 [2840] DEBUG Owasp.DefApp.DefenceMainSettingHandler - Kyrillisch (Windows)
120 [2840] DEBUG Owasp.DefApp.DefenceMainSettingHandler - windows-1251
120 [2840] DEBUG Owasp.DefApp.DefenceMainSettingHandler - 1251
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Begin With The Form Handling
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Checking Form Object __VIEWSTATE-->dDwtMzg4MDA0Nz
120 [2840] DEBUG Owasp.DefApp.Convertors.OutputConvertors - Begin With CleanUp
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Checking Form Object TextBox1--><script>
120 [2840] DEBUG Owasp.DefApp.Convertors.OutputConvertors - Begin With CleanUp
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Validation Failure has been found invalidation of rule:textchecks
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Invalid Objects has been found according to rule textchecks
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Checking Form Object TextBox2--><script>
120 [2840] DEBUG Owasp.DefApp.Convertors.OutputConvertors - Begin With CleanUp
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Validation Failure has been found invalidation of rule:textchecks
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Invalid Objects has been found according to rule textchecks
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Checking Form Object Button1-->Button
120 [2840] DEBUG Owasp.DefApp.Convertors.OutputConvertors - Begin With CleanUp
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - End Of The Form Handling
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Begin With The Querystring Handling
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - End With The Querystring Handling
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - Begin With The Cookie Handling
120 [2840] INFO  Owasp.DefApp.DefenceMainSettingHandler - The Cookie Object To Be Check :ASP.NET_SessionId--->42zrl4
```

Tablo 1.4 DefApp Günlük Dosyası Örneği

```
      public abstract class DefAppPlugin
      {
            public abstract string PluginName();
            public abstract ArrayList DefinedFilters();
            public abstract ArrayList DefinedRules();
            public abstract Rule GetRuleByName(string rulename);
            public abstract Filter GetFilterByName(string filtername);
            private static readonly ILog log = LogManager.GetLogger(typeof
(DefAppPlugin));
            public static ArrayList XmlToPlugin(string AssemblyName, int
maxCount, ArrayList ary)
            {
                  ArrayList plugIns = null;
                  if (ary == null)
                        new ArrayList();
                  else
                        plugIns = ary;
                  Assembly assembly = Assembly.LoadFile(AssemblyName);
                  Type[] types = null;
                  int Count = 0;
                  try
                  {
                        types = assembly.GetTypes();
                  }
                  catch (Exception ex)
                  {
                        Console.WriteLine(ex.Message);
                  }
                  for (int i = 0; i < types.Length; i++)
                  {
                        if (types[i].IsClass && types[i].IsPublic &&
types[i].IsSubclassOf(typeof (DefAppPlugin)))
                        {
                              DefAppPlugin.log.Info(types[i].Namespace +
"." + types[i].Name);
                              Object obj =
```

```
Activator.CreateInstance(types[i], null);
                              if (!GeneralUtilities.IsNull(obj))
                              {
                                    plugIns.Add(obj);
                                    Count++;
                              }
                              if (Count >= maxCount)
                                    return plugIns;
                        }
                  }
                  return plugIns;
            }
            public static DefAppPlugin XmlToPlugin(string AssemblyName)
            {
                  Assembly assembly = Assembly.LoadFile(AssemblyName);
                  Type[] types = null;
                  try
                  {
                        types = assembly.GetTypes();
                  }
                  catch (Exception ex)
                  {
                        Console.WriteLine(ex.Message);
                  }
                  for (int i = 0; i < types.Length; i++)
                  {
                        if (types[i].IsClass && types[i].IsPublic &&
types[i].IsSubclassOf(typeof (DefAppPlugin)))
                        {
                              DefAppPlugin.log.Info(types[i].Namespace +
"." + types[i].Name);
                              Object obj =
Activator.CreateInstance(types[i], null);
                              if (!GeneralUtilities.IsNull(obj))
                              {
                                    return (DefAppPlugin) obj;
                              }
                        }
                  }
                  return null       }
                                          }
```

Tablo 1.5 DefAppPlugin Sınıfı

```
<configuration>
        <configSections>
                <sectionGroup name="AppSec">
<section name="AppGenerals" type="Owasp.DefApp.DefenceMainSettingHandler,DefAppModules"/>
                </sectionGroup>
        </configSections>
        <AppSec>
                <AppGenerals>
<Activation Debug="true" Active="true" DebugFile="c:\\testdebug.log"/>
<FoundStoneModule XMLRulesDatabase="bin/" ValidatorFormMappings="ValidatorFormMappings.xml"
ValidatorRules="ValidatorRules.xml" Active="true" FSPageOutput="false"/>
                        <Handlers>
                                <HandleCookies Action="1"/>
                                <HandleFormFields Action="1"/>
                                <HandleQueryString Action="1"/>
                        </Handlers>
<RuleList>
<rule name="textchecks1" Action="deny" Type="Textrule">&lt;SCRIPT&gt;</rule>
<rule name="textchecks" Action="deny" Type="Textrule">&lt;script&gt;</rule>
</RuleList>
<Plugins>
        <plugin assembly="" />
</Plugins>
</AppGenerals>
</AppSec>
<!-- End Of DefApp Configuration Section -->
<system.web>
<httpModules>
                <add type="Owasp.DefApp.DefenceModules, DefAppModules" name="ModuleLibrary"/>
</httpModules>
```

Tablo 1.6 Web.Config Konfigürasyon Örneği

4. SONUÇ

Bu çalışmada çeşitli Filtrasyon teknikleri ve saldırı algılama teknikleri kullanılarak daha önceden yazılım geliştirme esnasında yazılım güvenliğiyle ilgili dizaynları problemli olan uygulamaların kaynak kodlarına sahip olunmadan ve uygulama üzerinde hiçbir değişiklik yapılmadan düşük maliyetli ve ileri teknik bilgiye sahip olmayan bir kişi tarafından DefApp uygulaması kullanılarak nasıl güvenli hale getirileceği gösterilmiştir. Uygulama OWASP.org En Yaygın 10 Güvenlik Açığının çoğunun mevcut kod üzerinde herhangi bir değişiklik yapılmadan engellemesini sağlamıştır. Ayrıca uygulama şu anda endüstride kullanılması planlanan üç ayrı formata destek vermesi sonucunda hali hazırda hazırlanmış olan application firewall kural takımlarının kullanılmasında mümkün kılarak bakım ve öğrenme maliyetlerini ciddi anlamda düşürmektedir. Projenin son sürümü http://www.yazilimguvenligi.com/defapp adresinden ücretsiz olarak indirilebilir.

KAYNAKLAR

[1] OWASP.org, The Ten Top Most Critical Web Application Security Vulnerabilities, January 27th 2004

[2] Ivan Ristic , Where Do Web Application Firewalls Fit in the Overall Defense Strategy? March 02, 2005

[3] Grossman Jeremiah, webappsec.org, The 80/20 Rule for Web Application Security, 1/31/2005

[4] Zimmer David, sandsprite.com, Real World XSS, 11/04/03

[5] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, Microsoft Press , Improving Web Application Security: Threats and Countermeasures, June 2003

[6] Küsmezer I.Kerem, Cruz Dinis, OWASP.ORG , DefApp Api Documentation March 2005