Prevent the DDOS Attack and XML Injection Attacks Using XSD Trace

Grup Üyeleri:

Abdul Hannan AYUBI	152120181097
Osman ÇAĞLAR	152120181033
Serdar DEMİRTAŞ	152120181011
Furkan TAŞKIN	152120181029

Öğretim Üyesinin Adı:

Doç. Dr. Ahmet YAZICI

Grup Numarası : 11

23/05/2021

İÇİNDEKİLER

1 GİRİŞ 3

2 DDOS SALDIRILARI 4

2.1 Siber Saldırı Nedir? 4
2.2 DoS Saldırısı Nedir? 4
2.3 DDoS Saldırısı Nedir? 4
2.4 DDoS Saldırı Türleri Nelerdir? 5

3 XML ENJECKSİYON SALDIRILARI 5
3.1 XML Giriş 6
3.2 XML Nedir? 7
3.3 XML Özellikleri Nelerdir? 7
3.4 XML İle Neler Yapılabilir? 7
3.5 XML Kod Örneği 8
3.6 Enjeksiyon Nedir? 8
3.7 Enjeksiyon Saldırı Türleri Nelerdir? 8
3.8 XML Enjeksiyon Nelerdir? 9
3.9 XML Enjeksiyon Türleri Nelerdir? 9

4 DDoS Ve XML Enjeksiyon Saldırılarından Korunma 9
4.1 XSD Nedir? 9
4.2 XSD Niçin Kullanılır ve Amacı Nedir? 10
4.3 XS(XML Şema) Tarihçesi 10
4.4 XML Şemasının Görevleri 11
4.5 XSD Trace Nedir ? 12
4.6 XML Enjeksiyonu Koruma yolları nelerdir? 12
4.6.1 Statik Filtreleme 12
4.6.2 Dinamik Filtreleme 13
4.6.3 Statik ve Dinamik Filtreleme Arasındaki Farklar 14

5 PROBLEM ÖZELLİKLERİ VE ÇÖZÜM YAKLAŞIMLARI 15

5.1.1 Problemin Net Tarifi ve Çözüm Önerisi 15
5.1.2 Ele Alınan Problemin Ders ile İlişkisi 16

6 UYGULAMA YAZILIMI 15

6.1.2 Uygulamamızın Kısaca Anlatımı 15
6.2.1 Bazı Kod Örneklerimiz ve Açıklamaları 16

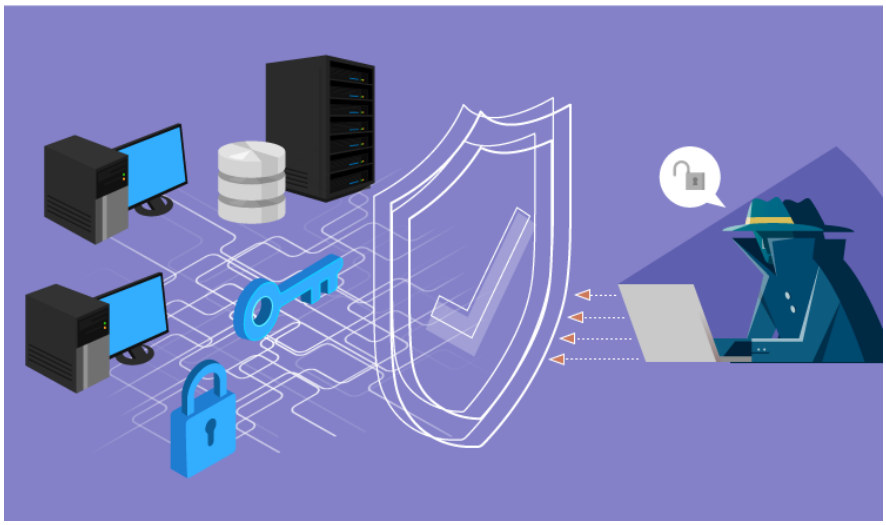
7 SONUÇ 16

8 PROJE EKİBİ DEĞERLENDİRMESİ 17
7.1 Grup Koordinatörü 17
7.2 Kim Hangi İşlerde Çalıştı? 17
7.3 Kim Ne Kadar Zaman Harcadı? (Adam-Gün) 17

9 KAYNAKLAR 19

1. GİRİŞ

Yaşadığımız çağın bir parçası haline gelen internet ve teknoloji dünyası her gün gelişmektedir. Buna paralel olarak insanların bu sektörlere ihtiyaçları da artmaktadır. İnternet dünyasının uçsuz bucaksız olması aslında bu sektörü çekici kılsa da insanların güvenliği ve internetin doğru kullanılması oldukça önemli bir konudur. Eski çağlardan beri önemini sürdüren güvenlik konusu gelişen teknoloji ile güvenlik konusu daha da bir önem kazanmıştır. İnsanlar tarafından her gün kullanılan sosyal medya ve sitelerin güvenliği şüphesiz en hassas ve mahrem konuların başında yer almaktadır. Peki bu güvenlik nasıl sağlanıyor? İnternete bağlı olan birçok sistem aslında her ne kadar güvenli olduklarını iddia etseler de insanlar tarafında her zaman o sistemler kırılmıştır. İnternetin güvenli olmasından sorumlu olan insanlara her gün ihtiyaçlar artmaktadır. Bunun sebebi ise insanların güvenliğe verdiği önemden kaynaklanmaktadır. Veri güvenliği, ağ güvenliği ve uygulama güvenliği olarak alt parçalara ayrılan güvenlik sektörü temelde yapılan bir sistemin veya uygulamanın ayakta durmasını, sorunsuz çalışmasını vb. çok önemli konuları sağlamak için oluşturulmuştur. Projemizin konusu da bu konulardan birini ele almaktadır. Geleceğin altını olarak da ifade edilen verilerin ne kadar güvenli olması gerektiğini, sunuculara hangi tür saldırıların yapılabileceğini ve



Fotoğraf 1.1

onlara karşı alınması gerekken önlemleri, veri alışverişinde güvenliğin sağlanmasını vb. konuları içermektedir. Günümüz dünyasında veri çok kıymetli soyut bir kavramdır. İnternet üzerinde saniyede milyonlarca veri aktarımı yapılmaktadır. Bu verileri doğru kaynaklara ulaşması ve doğru insanlar tarafında kullanılması oldukça önemlidir. Veri kaynağını yani sunucuların güvenliği bu anlamda çok büyük öneme sahiptir. Sunucuların ya da verilerin saklandığı bulut sistemlerinin güvenliğini tehdit edecek her gün binlerce saldırılar düzenlenmektedir. Bunların bazıları başarılı olurken bazıları da önlenmektedir. Başarılı olan saldırıların çoğunda insanların verileri yanlış kişilere ulaşmakta ve veriler farklı kullanıma sunulmaktadır. Alt başlıklarımızdan biri olan veri alışverişinde ve SEO (Search Engine Optimization) sistemlerinde kullanılan XML teknolojisini güvenliğinin sağlanması konusu yer almaktadır.

2.1 Siber Saldırı Nedir?

Virüs, trojan veya benzeri zararlı yazılımlarla gerçekleştirilen, genelde planlı ve koordineli olan bu zararlı davranışlara siber saldırı denilir. Siber saldırıların bir amacı olabildiği gibi aynı zamanda herhangi bir amaç olmadan, keyfi veya ego tatmini gibi çeşitli sebeplerden saldırılar düzenlenebilmektedir. Çeşitli siber saldırı yöntemleri bulunmakta ve bunlardan biri olan DDoS saldırısı asıl konumuzu oluşturmaktadır. [1]

2.2 DoS Saldırısı Nedir?

DoS (Denial of Service Attack), yani Türkçe adıyla Servis Dışı Bırakma Saldırısı saldırıları internete bağlı bir bilgisayarın işlevlerini aksatma veya kaynaklarını tüketerek çalışmasını durdurmak için yapılan popüler ve tehlikeli bir saldırı türüdür. [2]

2.3 DDoS Saldırısı Nedir?

DDoS (Distrubuted Denial of Service Attack), yani Dağıtık Hizmet Engelleme şeklinde Türkçeye çevrilen bu saldırı tipinde, sunuculara yönelik anlık ve aşırı bir yoğunluk oluşturularak kaynak tüketimini doruk noktalara çıkarmak hedeflenir. Saldırıyı yapan kişinin birden fazla ve hatta binlerce benzersiz IP adresi kullanarak saniyede terabit boyutlarında veriyi göndermesine imkân veren bir siber saldırı türüdür. Genel itibarıyla ele geçirilmiş olan milyonlarca IP adresi botnet* ağları yardımıyla, yapay trafikleri oluşturmak için bir nevi zombilere dönüşerek hedefe saldırır. Sunucularda da ciddi açıklar verilmesini sağlayan saldırı tiplerinden biridir.

(Fotoğraf 1.2)

DoS ve DDoS saldırıları, internet tarihinin en eski ve günümüzde dahi en sık karşılaşılan saldırı türlerindedir. Bu saldırı tipinde ağ protokollerinin ve bilgisayar kaynaklarının zafiyetleri kullanıldığı için büyük oranda saldırganlar başarılı olurlar.

Genel olarak bu saldırı tipi internette bulunan "hosting" adını verdiğimiz sunuculara yapılmakta. Bu saldırı başladığında ve yeterli düzeye ulaştığında "hosting" adı verilen sunucu tarafında hizmet kesintisi yaşanır ve kullanıcılar bu sunuculara bağlantı yapamazlar. Sonuç olarak bir e-ticaret sitesinin işlevinin durdurulması ya da bir bankanın hizmet verememesi gibi riskli durumlarla karşılaşılabilir.

Bu saldırı türünde spoof** adı verilen yöntemler kullanıldığı takdirde maalesef failler bulunamaz veya analiz sırasında çok uzun bir süre geçeceği için faillerin yakalanması imkânsız hale gelebilir. [3]

2.4 DDoS Saldırı Türleri Nelerdir?

2.4.1 Hacim Odaklı DDoS Saldırıları: Volume Based DoS olarak adlandırılan bu saldırı türünde sunucularda kullanılan bant genişliklerine saldırı yapılmaktadır. Sunucunun kapasitesinin üzerinde bir veri paketi gönderildiği zaman bu bant genişliği dolarak sunucunun cevap vermesinin önüne geçilir.

2.4.2 Protokol Bazlı DDoS Saldırıları: Bu saldırı türü Open Systems Inter Connection olarak adlandırılır ve OSI bünyesindeki katmanlar hedef alınır. Bu model içerisinde yer alan 3. ve 4. katmanlardaki protokollerin bünyesinde bulunan zafiyetler kullanılır. Tehlikeli ve bir o kadar da etkili bir saldırı türüdür. Protokollerin uzun yıllardır kullanılması ve güncellenmemiş olması da bu saldırı türünü popüler yapmaktadır.

2.4.3 Uygulama Katmanlı DDoS Saldırıları: Application Layer DDoS olarak adlandırılan bu saldırı türünde veri paketlerindeki GET ve POST özellikleri kullanılır. Hedef sisteme aşırı yüklenen GET ve POST istekleri ile sistem kaynaklarını tüketerek sunucunun cevap vermesi engellenir.

2.4.4 SYN Flood DDoS Saldırıları: Bu saldırı türü sunucu odaklıdır ve TCP paketleri kullanılarak yapılır. Ciddi sonuçlara sebep olabilen bu saldırı türünde kaynaklar tamamen tüketilerek sunucunun kilitlenmesi sağlanır.

2.4.5 UDP Flood DDoS Saldırıları: UDP protokolü hedef alınarak yapılan bir DDOS türüdür. Sunucuya aşırı şekilde UDP paketleri gönderilerek UDP Port'larının kullanılamaz hale getirilmesi sonucunda sunucunun cevap verememesi sağlanır.

2.4.5 PING Flood DDoS Saldırıları: Bu saldırı türünde PING paketleri kullanılır ve aşırı şekilde tekrarlanarak sunucunun kaynak tüketimi hedef alınır. Yoğun bir şekilde gelen PING paketlerine sunucu cevap vermekle uğraşırken CPU ve RAM kullanımının kilitlenmesi ve sunucunun cevap vermemesi sağlanır. [4]

Eğer ki bir DDoS saldırısına maruz kalırsanız hat limitleriniz ve sunucu kapasitelerinize aşırı yüklenme olacağından dolayı sisteminiz devre dışı kalacaktır. Bu da iş sürekliliğinizi etkilediği gibi beraberinde maddi ve manevi kayıplar yaşamanıza, hatta markanızın itibarının zedelenmesine kadar birçok noktada sıkıntılı süreçler yaşamanıza sebep olabilir. Büyük şirketlere yapılan saldırıların yanında aynı zamanda devletlere de DDoS saldırıları yapılmakta ve çeşitli devlet kurumlarının işlevsiz hale gelmesine neden olunmaktadır. Bu tür olaylar halkı isyana sürükleyebileceği gibi sağlık hizmetlerini aksatabileceğinden ölümlere dahi yol açabilmektedir. [5]

*__Botnet__: Botnet'i tam olarak açıklayacak olursak, korsanlar tarafından, zararlı yazılım trojan 'ın çeşitli bilgisayar sistemlerine bulaştırılarak bu bilgisayardaki yetkilere sahip olmak demektir. Bir botnet trojan'i sisteme bulaştıktan sonra artık o bilgisayar bir zombiye dönüşerek botnet yöneticisinin istediğini yapmaya hazır hale gelmektedir.

Spoof: Yanıltmak, bir siber suçlunun güvenilir bir kullanıcı veya cihaz kılığına girip size korsana fayda sağlayacak ve size zarar verecek bir şey yaptırması anlamına gelen, geniş kapsamlı bir terimdir. [6]

3.1 XML Enjeksiyon

Günden güne gelişen internet teknolojisi, birçok kullanıcı için alışveriş, oyun, sosyal medya olarak bilinse de internet teknolojisi bunun çok ötesindedir. Günümüzün birçok saatini ayırdığımız sitelerin temelinde aslında basit görünse de bunun alt yapısında karmaşık teknoloji yapmaktadır. Ve bu kullandığımız sitelerin alt yapısı her gün değişiyor ve gelişiyor. İnternetin temel kullanımlarında biri olan veri alışverişi ise günümüz dünyasında çok önemli bir yere sahip. Geleceğin altını olarak bilinen verilerin korunması ve doğru kullanılması hem kullanıcılar açısında hem internet sahiplerinin çok önemlidir. XML teknoloji temelde veri alışverişinde devrim yarattığı söylenebilir. İnternetin başlangıçlarında veri alışverişin manuel yapılması hem hız açısında hem güvenlik açısında sorunlara yol açıyordur bunun önünü engelleyen teknolojilerden biri de XML teknolojisidir.

Günümüz dünyasında interneti veri alışverişi olarak da tanımlamak mümkündür. Yazılımcılar için veri alışverişi bu anlamda çok önemlidir. Kullandığımız birçok web sitelerinde veri alışverişinin kolay, hızlı ve güvenli olması için bir çok yazılım dili kullanılıyor ve onlardan biri de XML'dir. Peki XML nedir? XML (Extensible Markup Language) ya da Türk Dil korumu (Genişletilebilir İşaretleme Dili) olarak öneriyor. Peki Markup Language nedir? Markup Dili Nedir? Verileri etiketlerle işaretlemeye yarayan yapay bir dildir. Örnek markup dilleri arasında en popüler ve yaygın kullanılanlardan bazıları HTML, TeX, XML, LaTeX, WML'dir. Hem insanlar hem bilgi işlem sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan bir işaretleme dilidir. W3C (World Wide Web Consortium) tarafından tanımlanmış bir standarttır. Bu özelliği ile veri saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara format görevi de görür. (. XML dili açık kaynaklı kod ve kullanım ve geliştirme hakları, herhangi bir ülke veya kuruluşun tekelinde bulunmadığında dolayı yazılımcılar daha etkili programlama yapma ve verileri daha basite ve standardize etme imkanına sahip oluyorlar. XML sayesinde farklı türlere sahip verileri türleriyle beraber muhafıza edebiliyoruz. Ve bu sayede verilere daha kolay ve hızlı bir şekilde ulaşmamıza olanak sağlanıyor.

Internet de kullandığımız birçok format veya farklı türlere sahip verileri başka türlere veya formata çevirdiğimiz zaman birçok özelliğini koruyamamaktadır. Bu durum verilere ulaşmamızı hem zorlaştırıyor hem güvenlik açısında birçok açığa sebep olabiliyor. Örneğin hemen hemen her gün kullandığımız bir çok format (.docx , .pdf, .xlsx , wbea, mp4, .png vb.) başka formatlara çevirdiğimiz zaman bir çok özelliğini kaybettiğini görüyoruz. XML bu sayılan formatlara ortak bir meta alan oluşturarak veri

alışverişini kendi formatlarıyla saklayarak yapabiliyor. Böyle hem verimizin formatı veya türü korunmuş oluyor hem veri transferi basite indirgenmiş oluyor. XML kaynak veri tabanındaki içeriği hedef platforma uygun hale getirmek ve platform transferinde programcılar için çok büyük kolaylık sağlamaktadır. [7]

3.2 XML Özellikleri nelerdir?

- XML, Case Sensivite (Büyük Küçük harfe duyarlı) bir dildir.
- XML, markup dillerinin açıklamasında kullanılan meta dilidir.
- XML formatında yazı yazarken, açılan etiketler kesinlikle kapatılmalıdır.
- XML formatında yazı yazarken, etiketleme standart değildir yani etiketlere nitelik tanımlanabilir.
- XML formatında yazı yazarken, etiketlemeler hiyerarşik yapıda olmak zorundadır. [8]

3.3 XML ile Neler Yapılabilir?

Site haritası oluşturabilir. XML site botları arama motoru botları için yazılmıştır. Herhangi bir arama motoru botu, XML dosyasına bakarak siteyle ilgili tüm gerekli bilgileri çok seri biçimde çıkarır.

SEO (Search Engine Optimization) önemlidir. XML sayesinde sitemizde olan bir içeriği SEO tarafından algılanmasını kolaylaştıracak algoritmalar oluşturulabilir.

Veritabanlarının aktarımı yapılabilir.

Geliştirilmiş E-Ticaret Takibi için DataLayer (Veri Katmanları) oluşturulabilir. [9]

3.4 XML Kod Örneği

```
<?xml version="1.0" encoding="utf-8"?>
<Databases>
  <Database>
    <Username>userName 123456789</Username>
    <IP>124.222.15.7</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>User1234</Username>
```

```
    <IP>124.222.333.23</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>User?1234</Username>
    <IP>222.134.33.222</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>User_0n3</Username>
    <IP>111.232.335.42</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>Us3r_21m</Username>
    <IP>111.123.211.90</IP>
    <Password>123</Password>
  </Database>
</Databases>
```

3.5 Enjeksiyon Nedir?

Injection (Enjeksiyonlar), web uygulamalarını hedefleyen en eski ve en tehlikeli saldırılar arasındadır ve veri hırsızlığına, veri kaybına, veri bütünlüğünün kaybına, hizmet reddine ve tam sistem tehlikesine yol açabilir. Enjeksiyon güvenlik açıklarının birincil nedeni kullancı girişlerinde genellikle yetersiz doğrulamadan kaynaklanmaktadır. Enjeksiyon saldırısı, ağa enjekte edilen ve veri tabanındaki tüm bilgileri saldırgana getiren kötü amaçlı bir kod veya yazılımdır. Bu saldırı türü, web güvenliğinde önemli bir sorun olarak kabul edilmekte ve OWASP (The Open Web Application Security Project) İlk 10'da bir numaralı web uygulaması güvenlik riski olarak listede yer almaktadır. [11]

3.6 İnjection (Enjeksiyon) Saldırılarının Sınıflandırılması

- SQL Injection (SQL Enjeksiyon)
- Cross-site scripting (XSS) Siteler arası komut dosyası çalıştırma (XSS)
- XPath injection XPath enjeksiyonu
- Template injection Şablon enjeksiyonu
- Code Injection: Kod Enjeksiyonu
- CRLF Injection : CRLF Enjeksiyonu
- Email header injection : E-posta üstbilgisi Enjeksiyonu
- Host Header Injection Ana Bilgisayar Üstbilgisi Enjeksiyonu
- LDAP Injection LDAP Enjeksiyonu
- OS command injection OS komutu Enjeksiyonu [11]

3.7 XML Enjeksiyon Nedir?

XML enjeksiyon saldırısı bir XML uygulamasını veya XML hizmetini mantığını kavrayıp onu tehlikeye atan bir saldırıdır. İstenmeyen XML içeriğini veya XML yapısına uygun yazılımı XML mesajına enjekte edip bir uygulama veya yazılımı amacı dışında kullanılmaya müsait hale getirilebilir. Eğer herhangi bir XML enjeksiyon saldırısı başarılı bir şekilde tamamlanırsa saldırgan tüm veri tabanını çalabilir veya hatta web sitesinin **Admin** yöneticisi olarak oturum açabilir. XSS, Dos ve Ddos saldırıları gibi diğer saldırı türleri de kötü amaçlı XML enjeksiyonu saldırıları için kullanılabilir. [12]

3.8 XML Saldırı Türleri

XML saldırı türleri genellikle ikiye ayrılır. Bunlar biri XML Bomb Attacks (XML bombalı Saldırı) bir diğer ise XXE Injection (XML XXE Enjeksiyon) saldırısıdır.

Bir XML Bomba saldırısı hem iyi biçimlendirilmiş hem de geçerli XML olabilir, ancak XML ayrıştırıcısının veya çıktısını işleyen uygulamanın yürütülürken kilitlenmesine veya çökmesine neden olacak şekilde tasarlanmıştır.

XML XXE Enjeksiyon saldırısı: OWASP tanımına göre, XML XXE saldırısı, XML girişini ayrıştıran bir uygulamaya yönelik bir saldırı türüdür. Bu saldırı, harici bir varlığa başvuru içeren XML girdisi, zayıf yapılandırılmış bir XML ayrıştırıcı tarafından işlendiğinde meydana gelir. [12]

4.1 XSD Nedir?

XSD en kısa tabirle bir dosya uzantısıdır. Metin tabanlı bir dosya biçimidir ve XML formunu açıklayan ve o XML belgesinin yapısının nasıl olacağını belirten, XML belgesi ile ilgili kuralları belirlemek için kullanılan XML tabanlı bir dildir. XSD dosya uzantısına sahip dosyalar "XML Şema" dosyalarıdır. Bu uzantıya sahip dosyaları açmak için genellikle "SchemaViewer", "Visual XSD" ya da buna benzer programlar kullanılır. Bu tarz programlar XSD uzantılı dosyaları bir ağaç formunda gösterir ve kullanıcının şemayı anlamasını kolaylaştıran yazılımlardır. XSD dosyaları aynı zamanda metin tabanlı dosyalar oldukları için sıradan metin editörleri tarafından da açılabilir.

XSD, XML Schema Definition (XML Şema Tanımı)' ın kısaltmasıdır. Bu tarz dosyalar XML dosyalarına dahil edilerek kullanılır. XSD dosyaları ".xsd" dosya uzantısına sahiptir. Bu dosya türünün kendisi de aynı zamanda XML belgesidir. XSD dosyaları gibi hemen hemen aynı amaçlara hizmet eden farklı isimlere sahip dosya türleride bulunmaktadır (DTD, Relax-NG, W3C gibi.). [13]

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="kisiler">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="100" name="kisi">

          <xs:complexType>
            <xs:sequence>
              <xs:element name="sira" type="xs:unsignedByte" />
              <xs:element name="adi" type="xs:string" />
              <xs:element name="soyadi" type="xs:string" />
            </xs:sequence>
          </xs:complexType>

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

4.2 XSD Niçin Kullanılır ve Amacı Nedir?

XSD, XML belgesinin yapısını, içeriğinde barındırdığı nesnelerin veri türlerini ve bu nesnelerin birbirleriyle olan ilişkilerini ve özelliklerini tutar. Nesneler arasındaki sınıf dağılımını barındırır. XSD, XML tabanlı olduğu için diğer şema tanımlarına göre çok daha yaygın kullanılır. Kısacası bir XML belgesine ait objelerin tanıtmak ve açıklamak amacıyla kullanılır. DTD' ye göre okumak ve anlamak daha kolay olduğu için daha çok tercih edilir. [14]

4.3 XS (XML Schema) Tarihçesi?

XSD 1.0 ilk olarak 2001 tarihinde yayınlanmış, 2004 yılında ise önceki sürümdeki birçok hatanın giderilmesi ile XSD 1.1 yayınlanmıştır. 5 Nisan 2012 tarihinde W3C Recommendation ekibine katılmıştır. [15]

W3C (World Wide Web Consortium): Web'in geliştirilmesini amaçlayan ve standartlarını belirleyen bir topluluktur. Bu kurumun asıl amacı HTML dilinin yaygınlaşmasını ve tüm cihaz ve tarayıcılar aynı görselliğe sahip olmasını sağlamaktır. Tüm web sitelerinin aynı standartlarda yazılmasını sağlamayı amaçlarlar.

4.4 XML Şemasının Görevleri

- Bir belgede görünebilecek öğeleri tanımlar.
- Bir belgede görünebilecek listeleri tanımlar.
- Hangi öğelerin alt öğe olduğunu tanımlar.
- Alt öğelerin sırasını tanımlar.
- Alt öğelerin sayısını tanımlar.
- Bir öğenin boş olup olmadığını tanımlar.
- Öğeler ve özellikleri için veri türlerini tanımlar.
- Öğeler ve özellikleri için varsayılan ve sabit değerleri tanımlar. [15]

4.5 XSD Trace Nedir?

Bir sistemin bütünlüğüne zarar verecek saldırıları veya verilerimizin sakladığımız veri tabanı sistemlerine yapılacak olan saldırıları gözlemlemek ve sistemde önlem almak için oluşturulan bir yazılımdır. Bu algoritma belli günler ve saatlerde yapılan kullanıcı isteklerini gözlemliyor. Ayrıca yapılan DDos saldırılarının kayıtlarını bir sonraki saldırıları önlemek amacıyla tutabilir. Eğer ki belli bir kullanıcıdan yapılan istek sayısı bir gün de ortalama istek sayısından fazla ise o kullanıcı IP sını blokluyor/banlıyor. Ve Ayrıca bu algoritma Captcha oluşturuyor bu sayede kullanıcının bir bot mu ya da gerçek kullanıcı mı olduğunu tespit edebiliyor. Eğer ki kullanıcı IP adresi gerçek değilse o kullanıcı IP sını banlıyor. [16]

4.6 XML Enjeksiyonu Koruma yolları nelerdir?

XML Enjeksiyonundan korunmak için iki türlü filtreleme mevcuttur. Birincisi Statik filtreleme İkincisi ise Dinamik Filtreleme olarak ikiye ayrılır.

4.6.1 Statik Filtreleme

Rule number Description

Rule 1: Field-> Type checks-> Dynamic XSD decision -> Valid -> Grant

Rule 2: Field-> length checks-> Length lesser than expected-> Dynamic XSD decision-> Invalid-> Deny

Rule 3 Field-> length checks -> Length greater than expected -> Dynamic XSD decision -> Invalid -> Deny

Rule 4 Field> length Checks -> meets length criteria -> Dynamic XSD decision -> Valid -> Grant

Statik model de kurallar biraz daha katı olduğu için geliştirilen Dinamik modelde bunları daha sistematik ve hafifletilebilir yapılabilir. [17]

Static pattern with Length Restriction

```
<xsd:schemaxmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:elementname="UserName">
<xsd:simpleType>
<xsd:restrictionbase="xsd:string">
<xsd:patternvalue="[A-Z]*[0-9]*[a-z]*[^\-/]"/>
<xsd:lengthvalue="12"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:schema>
```

(Fotoğraf 1.3)

4.6.2 Dinamik Filtreleme

Adım 1: Kullanıcı giriş istekleri alınır.

Adım 2: Fixed XSD modülü tarafında girilen verileri doğrulama

Adım 3: Girilen verilerin sınır değerlerini hesaplama

Adım 4: Dinamik XSD tarafında oluşturulan şemadan verilerin sınır değerlerinin 1 den büyük veya küçük olmasını kontrollerini sağlama

Adım 5: Saldırı şüpheli girdiler kontrollerini sağlamak

Adım 6: Eğer girdilerin sınır değerleri uygun değilse kullanıcı banla ve I adresini bir sonraki saldırı için kaydet

Adım 7: Eğer kullanıcı gerçek ve doğru bir kullanıcı ise isteğini yerine getir

Adım 8: Kullanıcı isteği boyunca bu süreci devam ettir. [18]

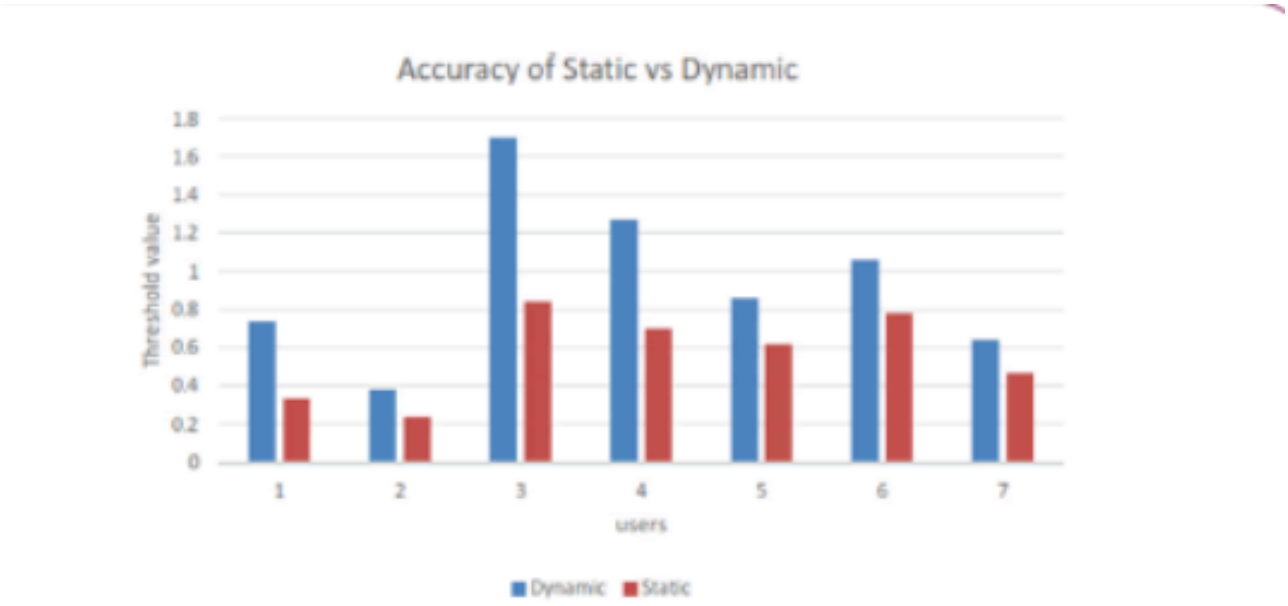
4.6.3 Statik ve Dinamik Filtreleme Arasında Farklar

Fixed grid XSD	Dynamic Grid XSD		
S. No.	Integer		Additional info
1	Type will be checked	Type will be checked	
2	Length will be checked	Length will be checked	
3	Space permitted or not	Space permitted or not	If space not permitted, input string will be space nullified
4		Negative character checks	
5		Float value checks	
6		Max permitted and min permitted values	

(Table-1)

Fixed grid XSD	Dynamic grid XSD		
S. No.		String	Additional info
1	Type will be checked	Type will be checked	
2	Length will be checked	Length will be checked	
3	Space permitted or not	Space permitted or Not	If space not permitted, Input String will be space nullified
4		Special character existence will be checked	
5		First character of the column can be caps or some special indications	
6		Numeric values permitted in the column	

(Table -2)

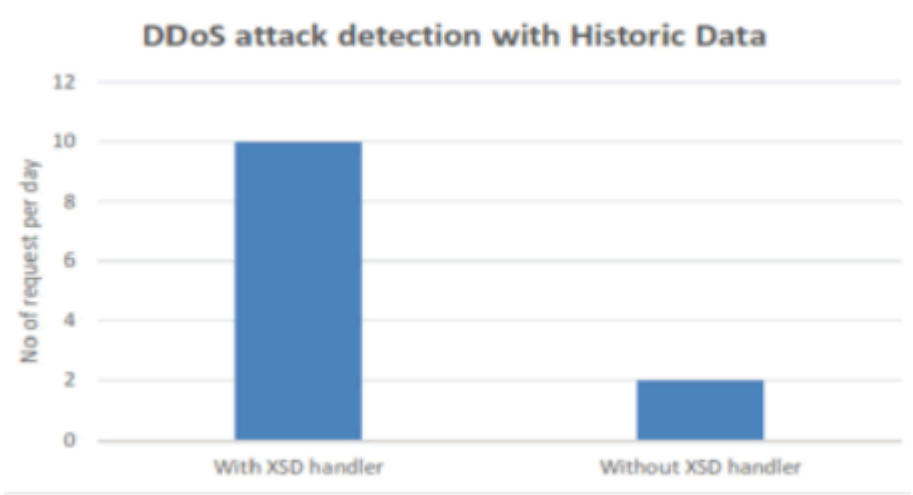


(Fotoğraf 1.4)

5.1 Problem Özellikleri ve Çözüm Yaklaşımı

5.1.1 Problemin Net Tarifi ve Çözüm Önerisi

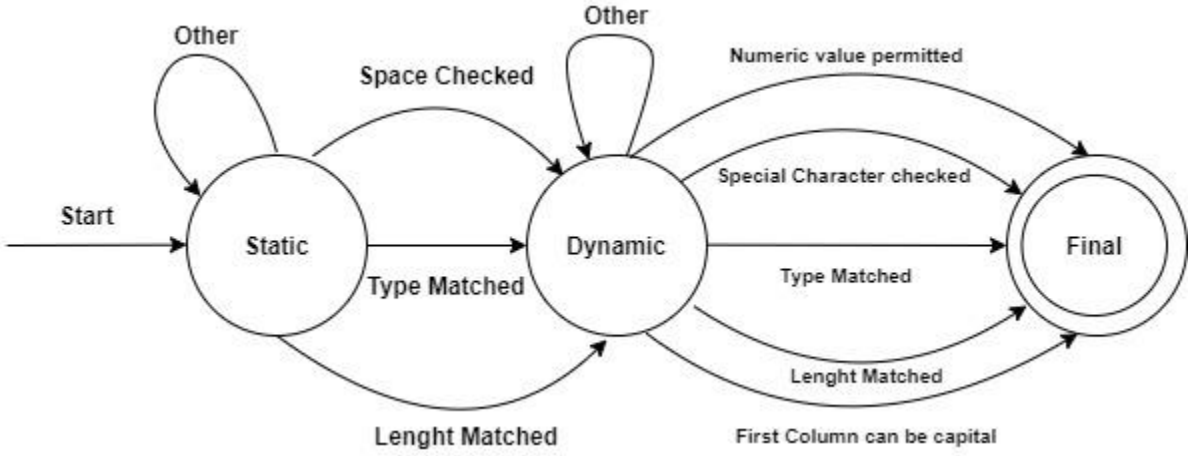
İnternetin gelişmesiyle birlikte ona olan ihtiyaçlarımız da her gün gitgide artmaktadır. Kullandığımız web sitelerin güvenliği günümüz dünyasında çok önemlidir. Ve web sitelerin temelini oluşturan unsurlardan biri de veri alışverişidir. Web sitelerin güvenliğini sağlamak amacıyla binlerce yöntem geliştirilmiştir. Temel olarak Web sitelerin de veri akışını en hızlı ve en güvenli bir şekilde sağlamak çok önemlidir. Veri akışını en güvenli bir şekilde sağlamak amacıyla web servis güvenlik alanında farklı çalışmalar mevcuttur. Projemizin konusu ise bu web güvenlik servis katmanlarında XSD Trace nasıl kullanabiliriz kısmıdır. XSD trace bir web servis güvenlik katmanı oluşturmakta ve bu katmanlar sayesinde iki tehlikeli saldırıları (DdoS saldırısı ve Enjeksiyon saldırısı) gibi saldırılardan korumaktadır. Bu Web servis katmanında kullandığımız XML ve XSD kodları ile veri akışındaki hızı ve güvenliği sağlamayı amaçlamaktayız. Kullandığımız XSD Trace, DTD, Xpath ve XSL teknolojileri sayesinde web sitelerimizdeki verileri kullanıcıların girdilerine bağlı olarak kontrollerini sağlamak ve yazılımda oluşabilecek olası risklere karşı tedbir almak gibi konularda başarılı olabiliriz. Doğal olarak web sitelerimizdeki güvenlik çok farklı şekillerde sağlanabilir ve daha fazla güvenlik için farklı güvenlik katmanları oluşturulabilir ve daha fazla kontroller sağlanabilir. Özellikle DdoS saldırılarından korunmak için DB için çok sağlam güvenlik katmanları oluşturulabilir. Güvenlik katmanlarını oluştururken veri akşını hızlı ve güvenli bir şekilde sağlanması temel hedefimiz olmalıdır.



(Fotoğraf 1.5)

5.1.2 Yapılan Projenin Ders ile İlişkisi

Projemizin temelini oluşturan konusu ise veri güvenliğidir. Veri akışını sağlarken yapılan işlemlerin güvenli ve hızlı yapılması çok önemlidir. Burada oluşturulan katmanlardan biri ise XSD Trace katmanıdır. Bu katmanda kullanıcının girdilerini kontrollerini sağlanmakta, Girdinin uzunlukların ve tipleri, içerisinde Rakam barındırıp barındırmadığı gibi konuları vb. konuları içermektedir. Şu ana kadar derste gördüğümüz ile ilişkilendirecek olursak bu katman için bir **NFA** sistemi oluşturulabilir. Start state 'i olarak kullanıcının girdiği girdilerin uzunluklarının ve içerisinde Alfabe ve Rakam olup olmadığını kontrollerini sağlayan bir NFA sistemi oluşturulabilir ve bunu sitenin alt yapısına kodlanabilir.



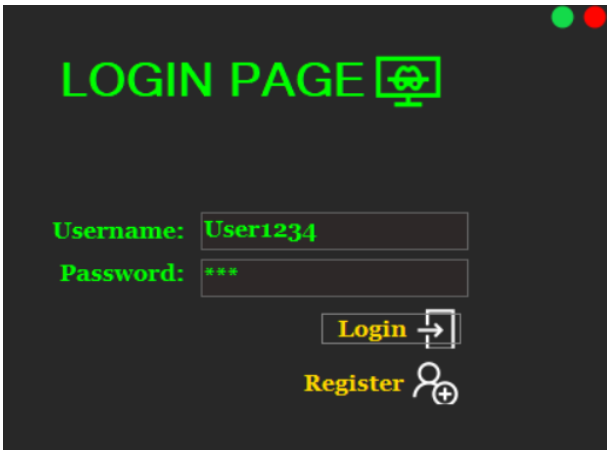
(NFA Diyagramı)

6.1 UYGULAMA YAZILIMI

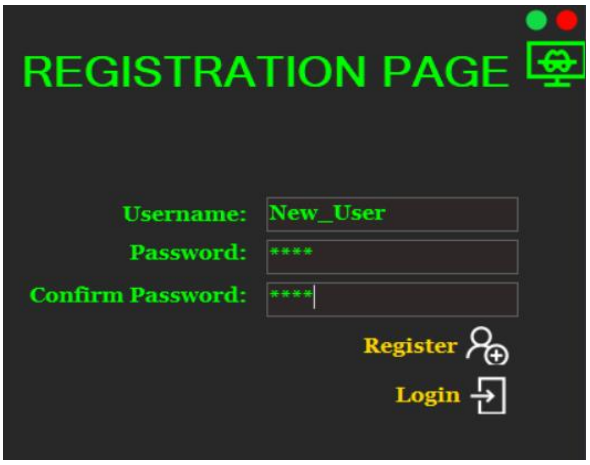
6.1.1 Uygulamamızın kısaca anlatımı

Bu uygulamamızda XSD Trace kullanarak DDoS ve XML enjeksiyon saldırılarından korunmak için bir uygulama simülasyonu yaptık. Bu uygulamamızı Windows Form kullanarak daha iyi sunabileceğimizi düşündüğümüzden dolayı C# programlama dilini tercih ettik.

Uygulamamızın ilk açılışında bizi kullanıcı ekranı karşılamakta. Bu ekranda istersek yeni kullanıcı ekleyerek veya varsayılan olarak bizim **XML** dosyamızdan çektiğimiz sabit kullanıcılar ile giriş yaparak uygulamamızı başlatıyoruz.



(Kullancı Giriş Sayfası)

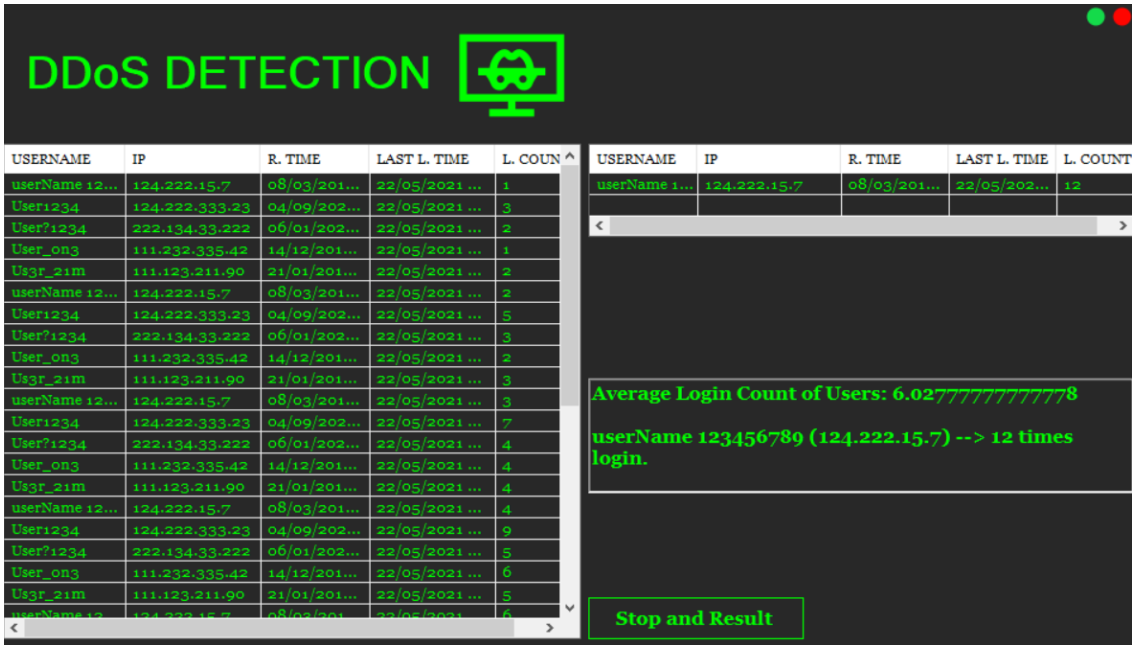


(Kullancı Yeni kayıt Sayfası Giriş Sayfası)

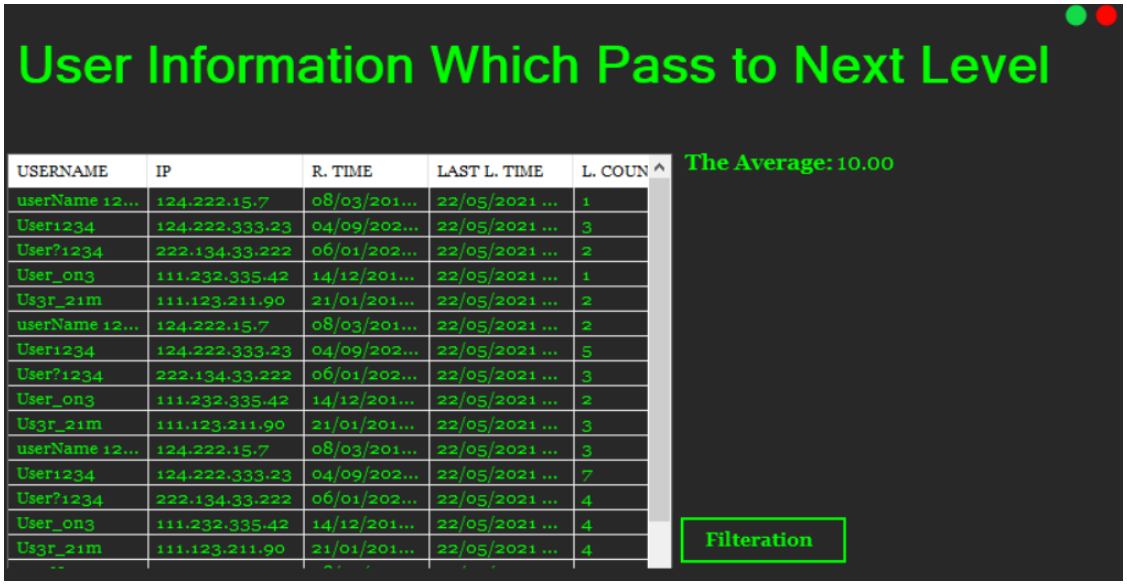
```xml
<?xml version="1.0" encoding="utf-8"?>
<Databases>
  <Database>
    <Username>userName 123456789</Username>
    <IP>124.222.15.7</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>User1234</Username>
    <IP>124.222.333.23</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>User?1234</Username>
    <IP>222.134.33.222</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>User_0n3</Username>
    <IP>111.232.335.42</IP>
    <Password>123</Password>
  </Database>
  <Database>
    <Username>Us3r_21m</Username>
    <IP>111.123.211.90</IP>
    <Password>123</Password>
  </Database>
</Databases>
```

Varsayılan kullanıcı bilgilerini içeren XML dosyamız

Sonrasında gelen DDoS Detection ekranının sol tarafında uygulamamıza giriş yapan kullanıcılarımızın kullanıcı adlarını, IP adreslerini, sisteme kayıt tarihlerini, son giriş tarihlerini ve son olarak giriş sayılarını izleyebiliyoruz. Sağ üst tarafta ise en son giriş yapan kullanıcımızın bilgilerini gösteren bir yer ve son olarak sağ alt tarafta ise toplam giriş sayısından kullanıcı sayısını bölerek elde ettiğimiz kullanıcı başına ortalama giriş sayısı bilgisini gösteren kısım bulunmakta ve her kullanıcı girişinde güncellenmektedir. En altta bulunan "stop and result" butonumuz ile kayıt almayı durduruyoruz. Elde ettiğimiz kullanıcı başına ortalama giriş sayısı bilgisi ile "filteration" butonu tıklayarak kullanıcı filtreleme işlemini yapacağımız sayfaya geçiş yapıyoruz.

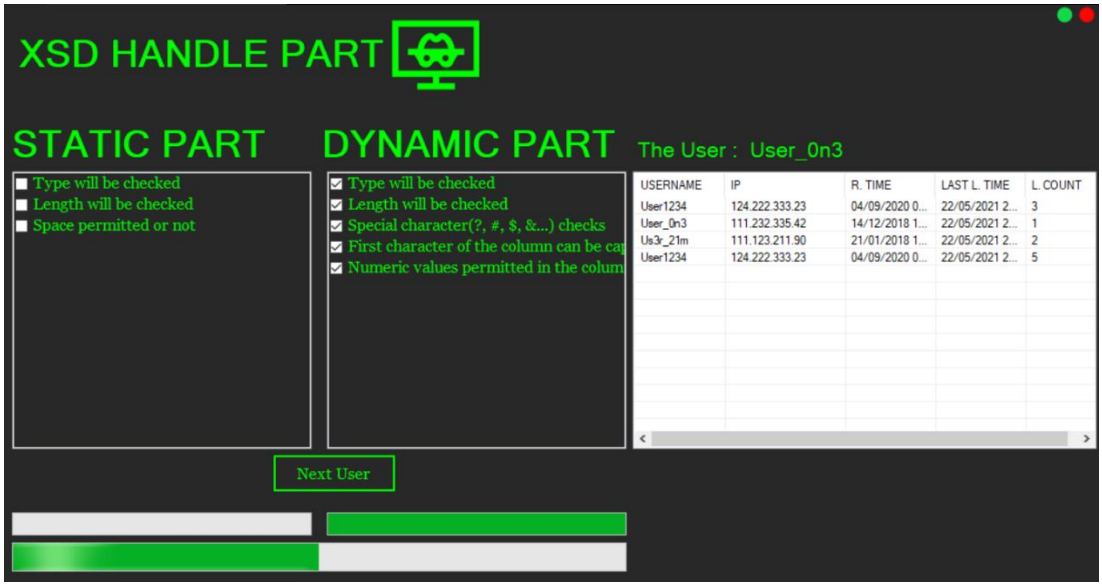


(DDoS Filtreleme Sayfası)



(DDoS Filtreleme Geçen Kullanıcılar ve Ortalama Giriş Sayısı)

Açılan yeni sayfamızda XSD Handle yöntemini kullanarak filtreleme işlemine başlıyoruz. Aşağıdaki görselde sol tarafta bulunan ve ilk olarak Static ve sonrasında Dynamic kısımlarındaki filtrelerden başarıyla geçen ve sisteme erişim izni verilen kullanıcılarımızı sağ taraftaki listede görebilmekteyiz. Tüm kullanıcıları filtreledikten programımız bize daha fazla kontrol edilecek kullanıcı kalmadığı uyarısı veriyor.



(Statik ve Dinamik Filtreleme ve Filtrelemeyi Geçen kullanıcı listesi)

Bu yaptığımız demo uygulamamızdan da anlaşılacağı üzere, gerekli filtremelerden geçemeyen kullanıcılarımız sisteme erişim sağlayamıyor ve bunun sonucunda sistem DDoS ve XML enjeksiyon saldırılarından korunarak sağlıklı bir şekilde çalışmasına devam ediyor.

6.2.1 Bazı kod örneklerimiz ve açıklamaları

Bu kodumuzda db.xml dosyamızda bulunan varsayılan (default) kullanıcılarımızın bilgilerini çekip uygulamamıza aktarıyoruz.

```
string username = string.Empty;
string IP = string.Empty;
string password = string.Empty;
users = new List<User>();

XmlDataDocument xmldoc = new XmlDataDocument();
XmlNodeList xmlnode;
int i = 0;
string str = null;
FileStream fs = new FileStream("db.xml", FileMode.Open, FileAccess.Read);
xmldoc.Load(fs);
xmlnode = xmldoc.GetElementsByTagName("Database");
for (i = 0; i <= xmlnode.Count - 1; i++)
{
    xmlnode[i].ChildNodes.Item(0).InnerText.Trim();
    username = xmlnode[i].ChildNodes.Item(0).InnerText.Trim();
    IP = xmlnode[i].ChildNodes.Item(1).InnerText.Trim();
    password = xmlnode[i].ChildNodes.Item(2).InnerText.Trim();
    User user = new User(username, password, IP, TimeGenerator.RandomDay(), DateTime.Now, 0);
    users.Add(user);
}
```

(XML programa dahil etme kodu)

Kodların bir kısmında Object oriented programming dan yararlandık. Kullanıcıların bilgilerin tutmak için bir kullanıcı sınıfı oluşturduk ve bu sınıfta kullanıcıların IP adreslerini , Kullanıcı adlarını vb. özelliklerini kaydettik. Daha sonra obje haline gelen kullanıcıları bir bu objeleri tutan list özelliklerinden yararlandık. Kullanıcı bilgilerin bu **List** özelliklerinde tutulması hem kod basitliği hem kullana birliğini artırmaktadır. Daha sonra bir statik DB sınıfı oluşturarak kullanıcı objelerini orada sakladık ve bu sayede **Singleton** design pattern ı da kullanılmış oldu.

Kullanıcın sisteme giriş yapabilmesi için geçmesi gereken çeşitli filtreler bulunmaktadır. Aşağıdaki görsellerde bu filtrelerden bazılarına ait kod örneklerini görebilirsiniz.

```
public int SpaceCheck(int i, int ifStatic)
{
    if (!(listViewNew.Items[i].SubItems[0].Text.Contains(" ")))
    {
        if (ifStatic == 0)
        {
            return 34;
        }
        else if (ifStatic == 1)
        {
            return 20;
        }
        else
        {
            return 0;
        }
    }

    else
    {
        return 0;
    }
}
```

(Boşluk barındırma kontrolü)

```
public int FirstCharCheck(int id)
{
    string firstChar = listViewNew.Items[id].SubItems[0].Text;
    char secondChar = firstChar[0];
    bool hasUpperCase = char.IsUpper(secondChar);

    if (hasUpperCase)
        return 20;

    return 0;
}
```

(İlk harf büyük olma kontrolü)

```
public int SpecialCharCheck(int id)
{
    List<char> blockedChars = new List<char>();

    char[] tempChars = { '#', '&', '?', ',', '*', '.', '$' }; // blocked charecters

    blockedChars.AddRange(tempChars);
    string userName = listViewNew.Items[id].SubItems[0].Text;

    int flag = 0;

    for (int i = 0; i < userName.Length; i++)
    {
        if (blockedChars.Contains(userName[i])) // valid username
        {
            flag = 1;
            return 0;
        }
    }
    if (flag == 0) // not valid username
        return 20;

    return 0;
}
```

(Özel karakter barındırma kontrolü)

7.1 SONUÇ

Sonuç olarak kullandığımız web sitelerinin omurgasını oluşturan sistemlerin ve veri alışverişinin sağlanması son derece önemli ve hassas bir konudur. Burada oluşturulan farklı web servis katmanları sayesinde bunları yapılabilmektedir. Güvenlik katmanlarını her ne kadar çok olması sistemin güvenliğine olan etkisi de paralel bir şekilde artmaktadır. Bizim projemizin konusu da güvenlik katmanlarını veri akışındaki önemini açıklamak ve temel veri saklanmasındaki teknoloji olan XML teknolojisini anlamak ve XSD Trace güvenlik katmanlarını kullanarak web sitemizdeki güvenliğini sağlamaktır. XSD Trace teknolojisi ile oluşturulan yazılım sayesinde sitemizdeki XML verilerini güvenliğini sağlamakta ve aynı zamanda iki önemli saldırılardan korunmak için tedbirler almaktayız. Bunun için oluşturulan DdoS saldırılarından korunma kısmında günlük belirli IP den yapılan istekleri gözlemlemek yapılan isteklerin gerçek IP adresinden olup olmadığını kontrollerini sağlamak ve buna karşı önlem almak için vardır. Bir diğer güvenlik katmanı olan XSD Filtreleme katmanında ise bunları bir üst düzey olan dinamik filtrelemeler ile kontrollerini sağlamak amacıyla oluşturulmuştur. Dinamik doğrulama sırasında, sistem zaman içindeki saldırıları tespit etmek için eğitilme eğiliminde yapılabilir ve bu tür bilgiler genelleştirilebilir ve öğrenilen veri seti doğrudan gelecekteki diğer tasarım doğrulama makinelerine dahil edilebilir, yani sistem karmaşık bir setle yeterince eğitildiğinde değerlerin ve hatta gerçek zamanlı olarak, saldırıların tespiti için sistem için gereken süre en aza indirilebilir. Sistemi saldırılara karşı belirli bir süre kararlı ve güvenilir hale getirilebilir. Mevcut web hizmetlerini ve kodu kesintiye uğratmadan böyle bir saldırı tespiti / doğrulaması, gelecekte iki adımda doğrulama mekanizması ve bu tür saldırıları önlemek için derin öğrenme mekanizması içerebilir.

8.1 PROJE EKİBİ DEĞERLENDİRMESİ

8.1.1 Grup Koordinatörü

Grup koordinatörü arkadaşımız 152120181097 NO'lu Abdul Hannan Ayubi'dir. Grup toplantı koordinasyonlarını ve grup içi görev dağılımını kendisi üstlenmiştir.

8.2.1 Kim Hangi İşlerde Çalıştı?

152120181011 Serdar Demirtaş → DDoS Attacks Ön Raporu Hazırlama, Rapor Son Hali Düzenleme, Sunum Hazırlama (Enjeksiyon Saldırıları), Ara Rapor Hazırlama, Demo Uygulama Yapımı, Final Raporu Hazırlama

152120181029 Furkan Taşkın → DDoS Attacks Ön Raporu Hazırlama, Görev Takip Sistemi Oluşturma, Sunum Hazırlama (Ddos Saldırıları), Ara Rapor Hazırlama, Demo Uygulama Yapımı, Final Raporu Hazırlama

152120181033 Osman Çağlar → XSD Trace Ön Rapor, Rapor İçi Görsellendirme, Sunum Hazırlama (XML, XSD nedir), Ara Rapor Hazırlama, Demo Uygulama Yapımı, Final Raporu Hazırlama

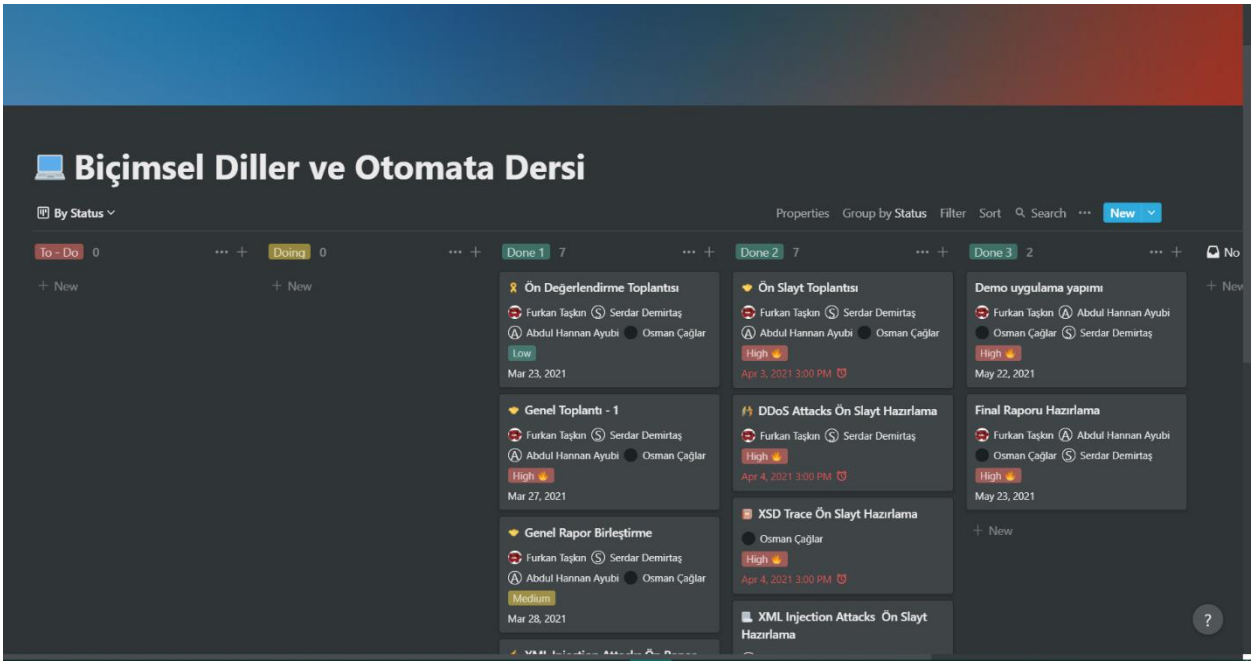
152120181097 Abdul Hannan Ayubi → XML Injection Attacks Ön Rapor, Grup Koordinatörlüğü, Sunum Hazırlama (XML Enjeksiyonlarından XSD Trace kullanarak koruma), Ara Rapor, Demo Uygulama Yapımı, Final Raporu Hazırlama

8.3.1 Kim Ne Kadar Zaman Harcadı? (Adam-Gün)

Görev takip sistemi olarak "Notion" uygulamasını kullandık. Kullanım kolaylığı açısından bu uygulamayı tercih ettik. Yapmış olduğumuz "To-Do List" aracılığıyla görev takibini yaptık.

Toplantılarımızı ve çalışmalarımızı "Discord" uygulaması üzerinden aynı anda gerçekleştirdik. Bu nedenle her grup üyesinin projeye harcadığı zaman hemen hemen aynıdır.

Öğrenci No	**Adı**	**Soyadı**	**Gün (Ön Rapor)**	**Gün (Sunum)**	**Gün (Ara Rapor)**	**Gün (Demo Uygulama ve Final Raporu)**
152120181097	Abdul Hannan	Ayubi	10 Saat (1 Gün)	24 Saat (3 Gün)	8 Saat (1 Gün)	32 saat (4 gün)
152120181033	Osman	Çağlar	8 Saat (1 Gün)	24 Saat (3 Gün)	8 Saat (1 Gün)	32 saat (4 gün)
152120181029	Furkan	Taşkın	8 Saat (1 Gün)	24 Saat (3 Gün)	8 Saat (1 Gün)	32 saat (4 gün)
152120181011	Serdar	Demirtaş	8 Saat (1 Gün)	24 Saat (3 Gün)	7 Saat (1 Gün)	32 saat (4 gün)


Biçimsel Diller ve Otomata Dersi
By Status
Properties
Group by Status
Filter
Sort
Search
New
To - Do 0
Doing 0
Done 1 7
Done 2 7
Done 3 2
Ön Değerlendirme Toplantısı
Furkan Taşkın Serdar Demirtaş
Abdul Hannan Ayubi Osman Çağlar
Low
Mar 23, 2021
Ön Slayt Toplantısı
High
Apr 3, 2021 3:00 PM
Demo uygulama yapımı
Osman Çağlar Serdar Demirtaş
High
May 22, 2021
Genel Toplantı - 1
Mar 27, 2021
DDoS Attacks Ön Slayt Hazırlama
Apr 4, 2021 3:00 PM
Final Raporu Hazırlama
May 23, 2021
Genel Rapor Birleştirme
Medium
Mar 28, 2021
XSD Trace Ön Slayt Hazırlama
XML Injection Attacks Ön Slayt Hazırlama


Open as page
Share
Updates
Favorite
Final Raporu Hazırlama
Assign
Furkan Taşkın Abdul Hannan Ayubi Osman Çağlar
Serdar Demirtaş
Status
Done 3
Priority
High
Date Created
May 22, 2021
Due Date
May 23, 2021
Property
Empty
Add a property
Add a comment...
Press Enter to continue with an empty page, or pick a template (↑↓ to select)
Task
Empty page

9.1 Kaynaklar:

1. https://berqnet.com/blog/siber-saldiri
2. https://berqnet.com/blog/dos-ddos-saldirisi-nedir
3. https://www.karel.com.tr/blog/ddos-nedir-ddos-saldirisi-nasil-yapilir
4. https://it.bilgi.edu.tr/tr/guvenlik/ddos/#:~:text=Distributed%20Denial%20of%20Service%20(Da%C4%9F%C4%B1t%C4%B1k,sisteme%20veya%20siteye%20giri%C5%9Finin%20engellenmesidir.
5. https://tr.wikipedia.org/wiki/Denial-of-service_attack#Sald%C4%B1r%C4%B1_ara%C3%A7lar%C4%B1
6. https://medium.com/@siberguvenlik/botnet-a%C4%9F%C4%B1-ddos-nedir-a-%C5%9F-2feec0c9fe35
7. https://www.kaspersky.com.tr/resource-center/definitions/ip-and-email-spoofing
8. https://stackoverflow.com/questions/3403644/what-is-the-purpose-of-xsd-files
9. https://www.w3schools.com/
10. https://stackoverflow.com/questions/6487171/when-should-xsd-files-be-used
11. (https://wmaraci.com/nedir/w3c)
12. (https://www.vidobu.com/egitim/xml-nedir-xml-nasil-kullanilir/20836/xml-teknolojisi-nedir)
13. XML - Vikipedi (wikipedia.org)
14. XML Nedir ve XML ne amaçla kullanılır? (mediaclick.com.tr)
15. XML Nedir? | Hosting.com.tr
16. A New Approach to Prevent the DDOS Attack and XML Injection Attacks Using XSD Trace Handler in Web Service | Insight Medical Publishing (imedpub.com)
17. A New Approach to Prevent the DDOS Attack and XML Injection Attacks Using XSD Trace Handler in Web Service | Insight Medical Publishing (imedpub.com)
18. 3_8_4-XML-Injections.pdf (wisc.edu)
19. XML external entity attack - Wikipedia
20. XXE (XML External Entity) Güvenlik Zafiyeti | BGA Security
21. (https://www.yusufsezer.com.tr/xml-xsd-nedir/)
22. Fotoğraf (1.1) Kaynak: https://www.pona.com.tr/ag-guvenligi-nedir/
23. Fotoğraf (1.2) Kaynak: https://www.varonis.com/blog/what-is-a-ddos-attack/
24. Fotoğraf(1.3)Kaynak: A New Approach to Prevent the DDOS Attack and XML Injection Attacks Using XSD Trace Handler in Web Service | Insight Medical Publishing (imedpub.com)
25. Fotoğraf (1.4) Kaynak: A New Approach to Prevent the DDOS Attack and XML Injection Attacks Using XSD Trace Handler in Web Service | Insight Medical Publishing (imedpub.com)
26. Fotoğraf (1.5) Kaynak: A New Approach to Prevent the DDOS Attack and XML Injection Attacks Using XSD Trace Handler in Web Service | Insight Medical Publishing (imedpub.com)