

TEMEL WEB UYGULAMA GÜVENLİĞİ

Siber Güvenlik Yaz Kampı
06.09.2022

\$whoami



Haktan EMİK

@SDÜ 2018

Sızma Testi Uzmanı @TURKCELL

OSCE | OSCP | CEH | TSE



haktanemik.medium.com



@haktanemik

\$whoami



Anıl BAŞ

@YTÜ 2017

Sızma Testi Uzmanı @TURKCELL

OSWE | OSCP | TSE



<https://gelecegiyazanlar.turkcell.com.tr/kisi/anilbas>

<https://www.exploit-db.com/papers?author=10694>



@anilbas1

WEB TEMELLERİ 01

OWASP TOP 10 02

BURP SUITE 03

GİRİDİ & ÇIKTI DENETİMİ 04

CROSS SITE SCRIPTING (XSS) 05

INJECTION 06

KİMLİK DOĞRULAMA 07

İÇİNDEKİLER

08

OTURUM YÖNETİMİ

09

YETKİLENDİRME

10

İŞ MANTIĞI ZAFİYETLERİ

11

DOSYA YÜKLEME

12

SERVER-SIDE REQUEST FORGERY (SSRF)

13

OPEN REDIRECTION

14

DIRECTORY TRAVERSAL

15

YAPILANDIRMA HATALARI

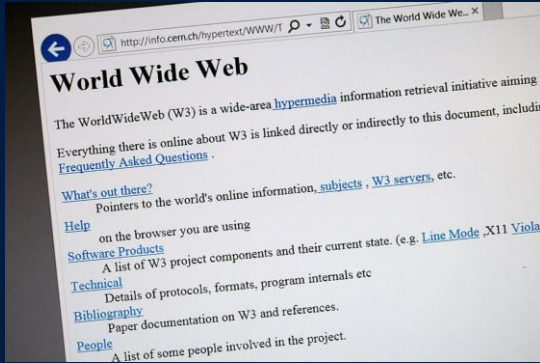
01

WEB Temelleri

Web dünyasına giriş
Temel kavramlar
Web nasıl çalışır

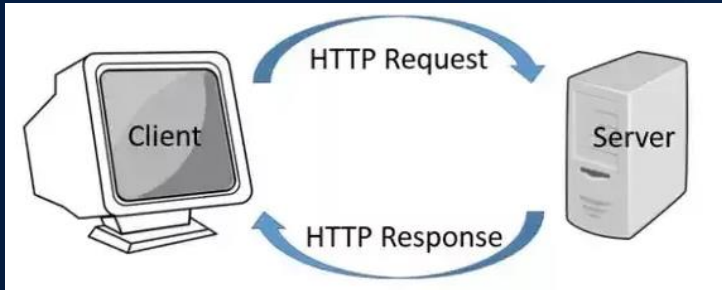


WEB Güvenliğinin Önemi



Hypertext Transfer Protocol (HTTP)

Uygulama seviyesinde bir iletişim protokolüdür
Stateless
İstemci (client) ve sunucu (server)
İstek (request) ve cevap (response)
İki bölümden oluşur; Header ve Body



HTTP İsteği

GET /index.html HTTP/1.1

User-Agent: Mozilla/4.0 (MSIE 6.0, ...

Host: example.com

HTTP Cevabı

HTTP/1.1 200 OK

Date: Fri, 31 Mar 2012 10:08:00 GMT

Server: Apache/2.4.7

Content-Length: 92

<html>

<head></head><body>...</html>

HTTP Metotları

Metot	Açıklama
GET	Veri okuma
POST	Veri gönderme/oluşturma
PUT/PATCH	Veri güncelleme/değiştirme
DELETE	Veri silme
HEAD	GET ile aynı ancak sadece başlık döner
OPTIONS	İzin verilen HTTP metotlarını döndürür

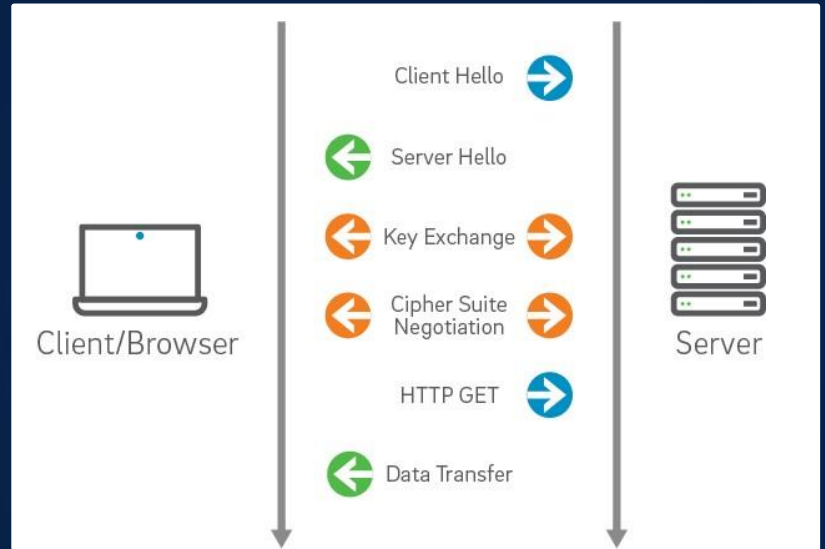
HTTP Durum Kodları

HTTP Status Codes



SSL/TLS

HTTP vs HTTPS



WEB Nasıl Çalışır?

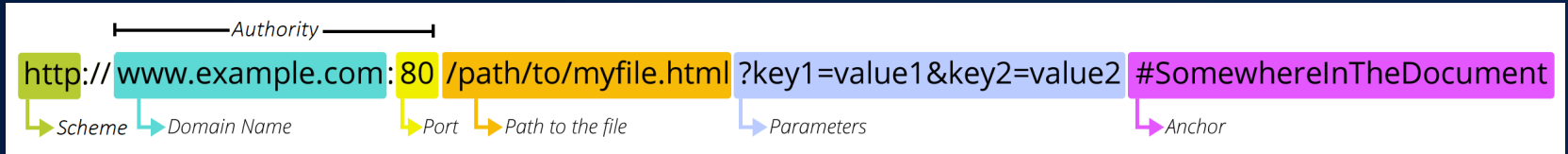
Tarayıcı açıp turkcell.com.tr adresine gitmek istediğimizde:



URL

(Uniform Resource Locator)

Kaynağı belirten tanımlayıcıdır.
Tarayıcılarda adres çubuğunda gördüğümüz ifadeler URL'dir.



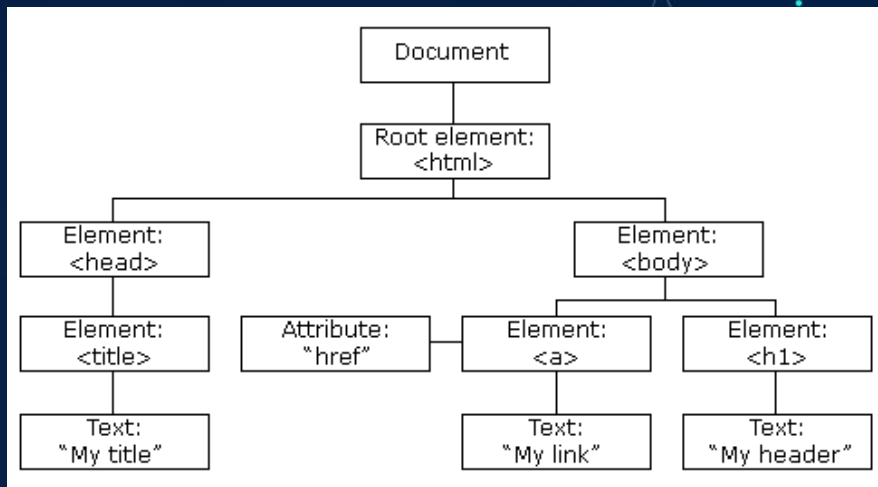
DOCUMENT OBJECT MODEL (DOM)

Tarayıcılar üzerinden görüntülediğimiz internet sayfaları birer **belge** ve bu belgedeki her bir eleman da bir **nesnedir**.

Uygulama → Belge
Belgedeki her bir eleman → Nesne

Tarayıcı, sunucudan dönen HTML'i parse eder, javascript ve CSS'leri de yorumlayarak DOM yapısını oluşturur..

DOM, Javascript gibi diller ile etkileşim sağlayan bir API'dir.





SAME ORIGIN POLICY (SOP)

Tarayıcılar tarafından, geliştirilen bir güvenlik mekanizmasıdır.

SOP, tarayıcılar (browser) tarafından yüklenen kaynaklarının, birbirleriyle olan paylaşımlarını/ilişkilerini belirli kurallar çerçevesinde kısıtlayan bir politikadır.

Farklı origin'e sahip kaynaklar birbirlerinin DOM'larına erişemez yani dönen cevabı okuyamaz!

SOP politikasına göre, yüklenen her bir kaynak; 3 bilginin birleşimi ile origin olarak tanımlanmaktadır.

ORIGIN

- Şema/Protokol
- Domain
- Port (Bağlantı noktası)

Örnek olarak,
<https://test.com/ornek/index.html>

Şema/Protokol: https
Domain: test.com
Port: 443

SAME ORIGIN POLICY (SOP)

<https://test.com/ornek/index.html> adresi üzerinden aşağıdaki URL'lere erişim durumları ve SOP politikasına göre alınacak yanıtlar;

URL	SONUÇ	NEDEN
https://test.com/ornek2/index.html	Erişebilir	Origin aynı, sadece dosya yolu farklı
http://test.com/ornek/index.html	Erişemez	Protokol ve port farklı
https://www.test.com/ornek/index.html	Erişemez	Domain farklı
https://test.com:8080/ornek/index.html	Erişemez	Port farklı



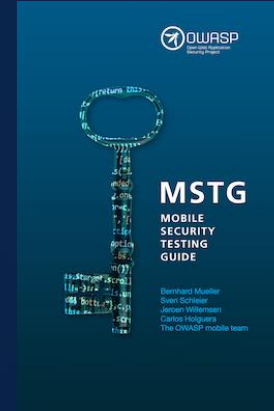
02

OWASP Top 10

OWASP
OWASP Top 10 Web Zafiyetleri
OWASP Top 10 API Zafiyetleri

OWASP

Kar amacı gütmeyen, uygulama güvenliğini arttırmak için çeşitli çalışmalar yayınlayan (checklist, best practice, testing guide ve TOP10 gibi) global bir topluluktur.





OWASP TOP 10 Web Zafiyetleri (2021)

A1 – Broken Access Control
A2 – Cryptographic Failures
A3 – Injection
A4 – Insecure Design
A5 – Security Misconfiguration
A6 – Vulnerable and Outdated Components
A7 – Identification and Authentication Failures
A8 – Software and Data Integrity Failures
A9 – Security Logging and Monitoring Failures
A10 - Server-Side Request Forgery



OWASP TOP 10 API Zafiyetleri (2019)

API1 – Broken Object Level Authorization
API2 – Broken User Authentication
API3 – Excessive Data Exposure
API4 – Lack of Resources & Rate Limiting
API5 – Broken Function Level Authorization
API6 – Mass Assignment
API7 – Security Misconfiguration
API8 – Injection
API9 – Improper Assets Management
API10 – Insufficient Logging & Monitoring

03

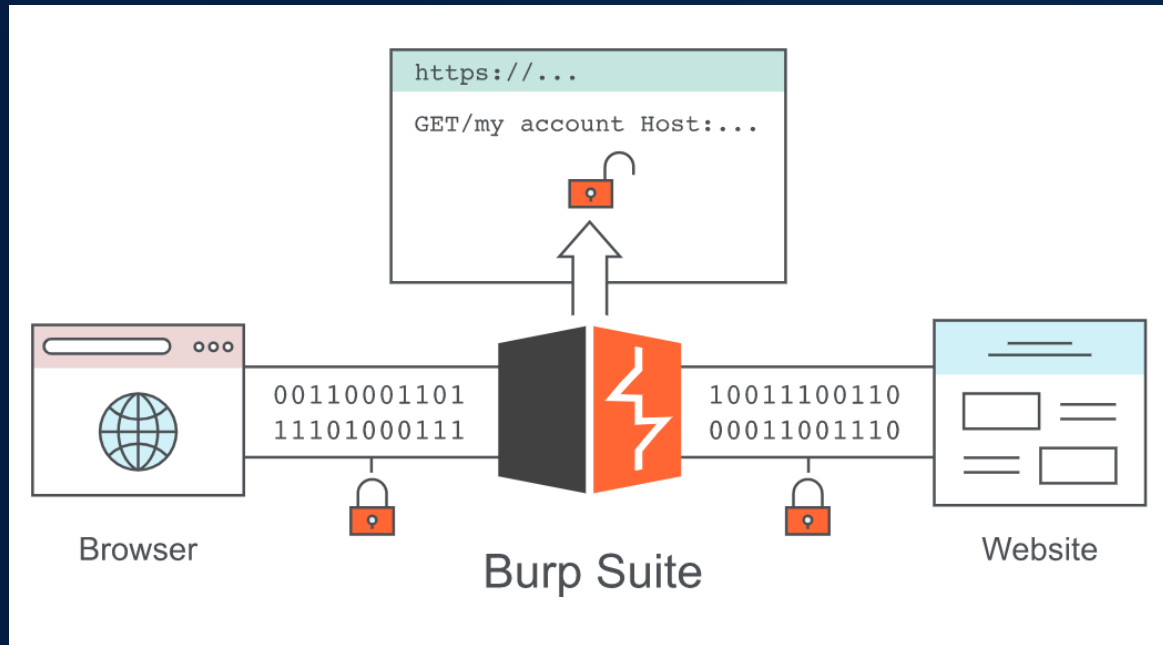
Burp Proxy

Burp Proxy
Burp Proxy Modülleri



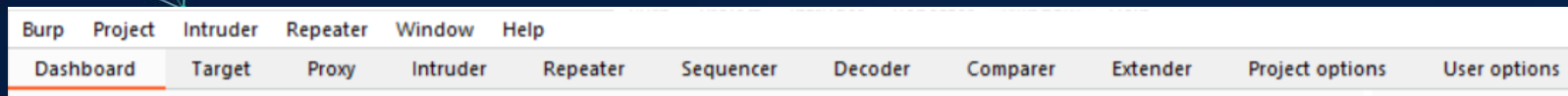
Burp Suite

Web uygulamaları üzerinde güvenlik testleri gerçekleştirmek için bir çok modül içeren proxy araçtır.



Burp Suite

Modüller



- **Target**

Hedef uygulamanın içerik ve fonksiyon gibi detaylı bilgilerine genel bir bakış sağlayan modül

- **Proxy**

Burp suite içerisinde en çok kullanılan modüllerden birisidir. Client ile server arasında istekleri inceleyip değiştirmek için kullanılır.

- **Scanner**

Hedef web uygulama üzerinde otomatize zafiyet taraması gerçekleştirmeye yardımcı olan modüldür.

- **Intruder**

Hedef web uygulamaya otomatize saldırı senaryoları oluşturmak ve spesifik saldırı tekniklerini test etmek için kullanılır.

- **Repeater**

HTTP isteklerini inceleyebilmeyi, istekler üzerinde değiştirme yapmayı ve bu değişikliğin uygulamada nasıl yorumlandığını görmemizi sağlayan basit ama işlevsel bir modül

- **Sequencer**

Veri örnekleminde rastgelelik kalitesini analiz etmek için bir araçtır. Oturum tokenları, anti-CSRF tokenlar gibi rastgele olması gereken parametrelerin kalitesini ölçmede yararlanılır.

- **Decoder**

Kodlanmış verileri çözümlmek veya ham verileri çeşitli kodlanmış ve karma biçimlere dönüştürmek için basit bir araçtır.

- **Comparer**

Herhangi iki veri ögesi arasında karşılaştırma yapmak için kullanılan basit bir araçtır.

Abstract geometric patterns in the top corners of the slide, consisting of interconnected lines and dots forming triangular and polygonal shapes.

Uygulama güvenliğinde en önemli nokta...

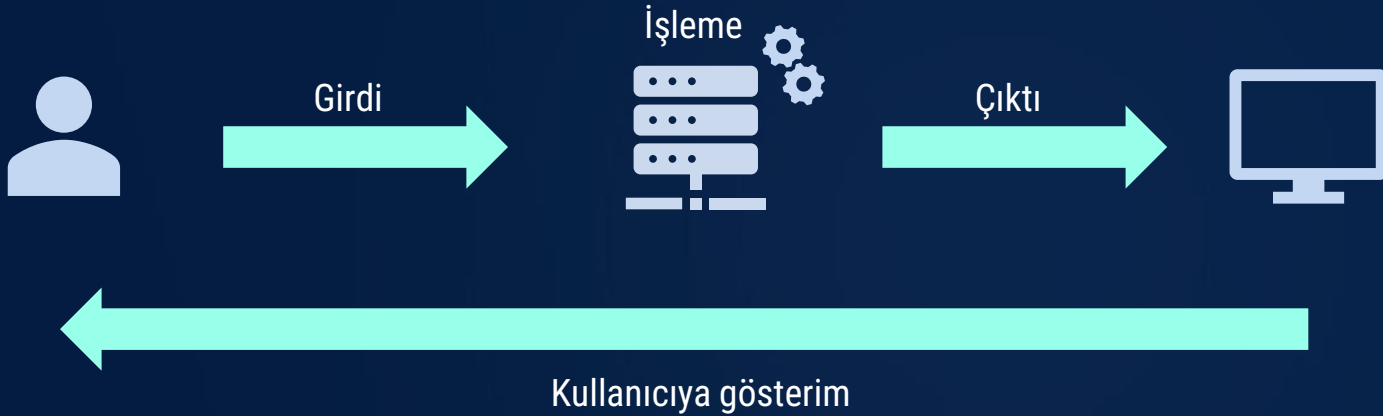
INPUT

04

Girdi & Çıktı Denetimi

Girdi & Çıktı Denetimi
Girdi & Çıktı Denetimi Yaklaşımları

Girdi & Çıktı Denetimi



Girdi Denetimi

Kara Liste (Black List) Yaklaşımı



Bilinen kötü karakter veya karakter dizileri reddedilir.
Örnek olarak "<script>" ya da or "1=1 -" geçen girdilerin reddedilmesi.



Varsayılan olarak tüm girdileri **işleme** eğilimindedir, yalnızca istenmeyen bir veri ile karşılaşıldığında reddeder.



Tehdit odaklı bir yaklaşımdır.



Deny list olarak da adlandırılır



Güvensiz bir yöntemdir ve kullanılması risklidir.

```
List<Pattern> blklistPtrns = new  
ArrayList<Pattern>();
```

```
Pattern p = Pattern.compile("<script>");  
blklistPtrns.add(p);
```

```
for (Pattern aPattern : blklistPtrns){  
    //tam matching yapmıyoruz  
    if (aPattern.matcher(taintedInput).find())  
        //throw exception  
}
```

Girdi Denetimi

Beyaz Liste (White List) Yaklaşımı



Uygulama yalnızca onay verilen karakter veya karakter dizilerini kabul eder.



Varsayılan olarak tüm girdileri **reddetme** eğilimindedir, yalnızca kabul edilen bir veri ile karşılaşıldığında işler.



Güven odaklı bir yaklaşımdır.



Allow list olarak da adlandırılır



Güvenli ve tavsiye edilen girdi denetimi

```
List<Pattern> = whtlistPptrns = new  
ArrayList<Pattern>();
```

```
aPattern = Pattern.compile("[a-zA-Z0-9]{2,100}");  
whtlistPptrns.add(aPattern);
```

```
for (Pattern aPattern : whtlistPptrns){  
    //tam matching yapıyoruz  
    if (!aPattern.matcher(taintedInput).matches())  
        // throw exception  
}
```

Çıktı Denetimi

Encoding



Kullanıcıya gösterilecek verilerin içindeki özel karakterlerin **teknolojiye özel formata** değiştirilmesi.

< → <
> → >



Amaç hedef yorumlayıcı için özel karakterlerin kodlama işlemi sonrası önemlerini yitirmiş olmasıdır.

```
String encJS = ESAPI.encoder().encodeForJavaScript( input );
```

```
Codec c = new OracleCodec();  
String escSQL = ESAPI.encoder().encodeForSQL( c, input );
```

05

Cross Site Scripting (XSS)

Reflected XSS
Stored XSS
DOM XSS

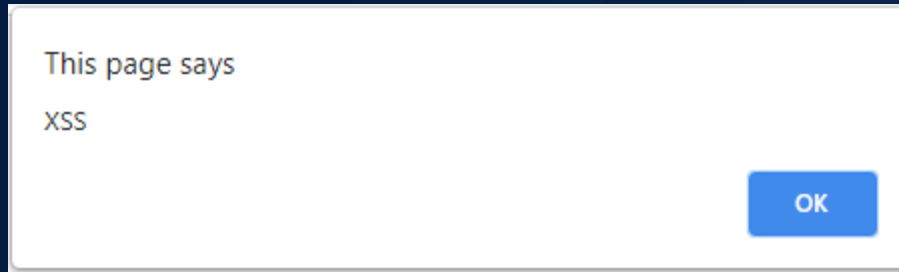
Cross Site Scripting (XSS)



Kullanıcıdan alınan verinin, tekrar kullanıcıya gösterildiği yerlerde olabilir.



Kullanıcı, `<script> alert("xss") </script>`



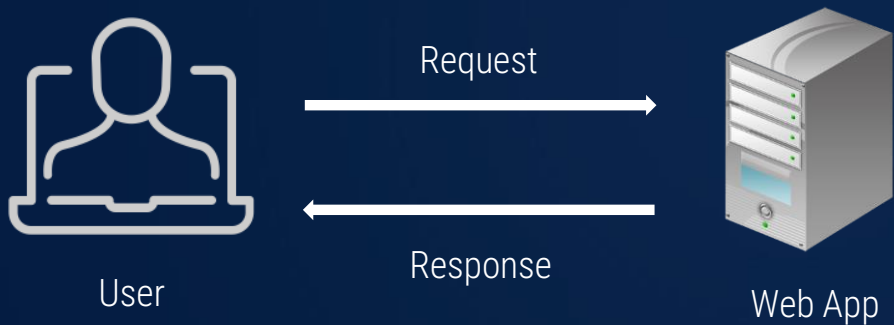
Cross Site Scripting (XSS)

Türleri



Reflected XSS

Kullanıcıdan alınan verinin tekrar kullanıcıya yansıtıldığı durumlarda oluşur. Saldırı kodu istek içerisinde yer aldığı için isteği yapan kişide zafiyet tetiklenir.





Uygulama

Reflected XSS

Örnek-1



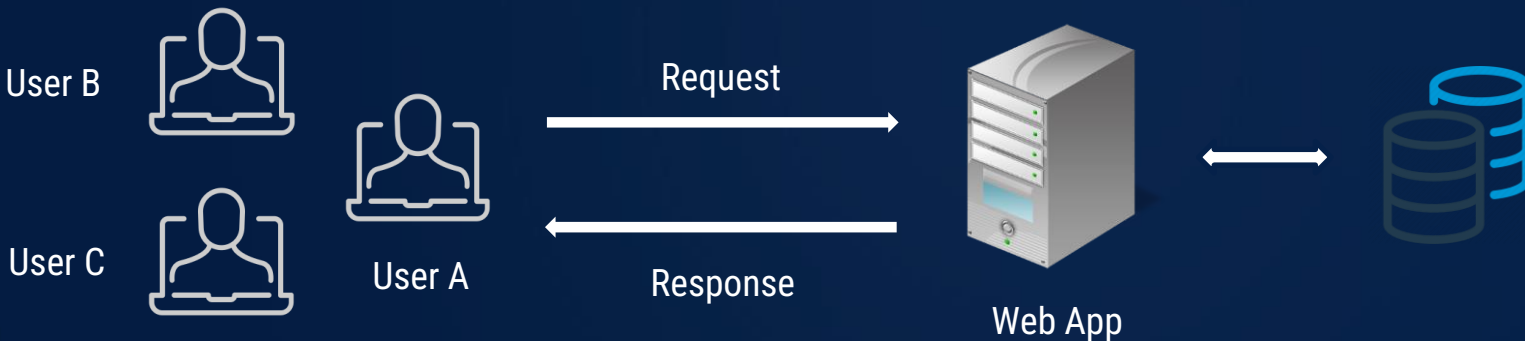
Cross Site Scripting (XSS)

Türleri



Stored XSS

Gönderilen saldırı kodu DB'ye kayıt edilir. Ardından, verinin gösterildiği yerde zafiyet tetiklenir.





Uygulama

Stored XSS

Örnek-1



Cross Site Scripting (XSS)

Türleri



DOM XSS

Javascript ile DOM manipölasyonu yapılırken, kullanıcıdan alınan verinin kullanılması durumunda ortaya çıkar.



User



DOM XSS

Sources & Sinks

Sources

`document.documentURI`
`document.baseURI`
`location.search`
`location.href`
`location.hash`
...



Sinks

`document.write`
`document.writeln`
`element.innerHTML`
`eval`
`setTimeout / setInterval`
`execScript`
...

<https://github.com/wisec/domxsswiki/wiki/>



Uygulama

DOM XSS

Örnek-1



Cross Site Scripting (XSS)

Nelere Sebep Olabilir?



Oturum bilgisi çalınabilir



Kullanıcılar, zararlı sayfalara yönlendirilebilir



Tarayıcıda JS çalıştırılabildiği için saldırganın hayal gücüne kalmış durumdadır ☹

Cross Site Scripting (XSS)

Önlemler



Verinin, nereye basılacağına uygun olarak encode işlemi uygulanmalıdır. (Context bazlı encoding)



DOM XSS için doğru çıktı metodu kullanılmalıdır.

Örneğin, bir div etiketine yazmak için innerHTML yerine textContext kullanılması gibi

```
<b>Current URL:</b> <span id="contentholder"></span>
...
<script>
document.getElementById("contentholder").textContent = document.baseURI;
</script>
```



Cookie'ye HTTPOnly bayrağının eklenmesi, Content Security Policy, X-XSS-Protection

06

Injection 101

OS Command Injection
XML External Entity
SQL Injection



Injection



Türkçe olarak "dahil etme" diyebiliriz.



Herhangi bir alt yapıda kullanıcı girdisi "**herhangi bir girdi denetimi olmadan**" dinamik olarak sorgulamaya dahil edilirse o altyapıda injection meydana gelebilir.



SQL

LDAP

NoSQL Query

OS Komutları

XML Parser

SMTP Header

ORM Query

XPATH





Genellikle sunucunun doğrudan ele geçirilmesine neden olabilirler.

OS Command Injection



shell (kabuk) meta karakterleri

Ayrıcılar (separators):

	<p>& && </p>	<p>; Newline (0x0a or \n) #</p>	
---	--	--	---

Inline execution:

` injected command `

\$(injected command)

Terminal zamanı!



OS Command Injection



Kullanıcı girdilerinin kontrolsüz olarak OS komutlarına dahil edilmesi sonucu oluşan zafiyetlerdir

```
ProcessBuilder b = new ProcessBuilder("C:\\DoStuff.exe -arg1 -arg2");
```

```
ProcessBuilder b = new ProcessBuilder("ping " + ip);
```

```
http://sensitive/something.php?dir=%3Bcat%20/etc/passwd
```



Uygulama

OS Command Injection

Örnek-1

Örnek-2



OS Command Injection

Önlemler



OS komutlarını doğrudan çağırmak yerine built-in fonksiyonları kullanmayı tercih edebilirsiniz.



Uygulamayı çalıştıran kullanıcıların mümkün olduğunca minimum OS komutu çalıştıracak şekilde yetkilendirilmesi önerilir.



Kullanıcı girdileri validasyon işleminden geçirilmeli.



Validasyonda **White list**'ler kullanılabilir.

XML External Entity Injection (XXE)



XML (eXtensible Markup Language), verileri depolamak ve taşımak için tasarlanmış bir dildir



XML Parsing

```
<?xml version="1.0"?>
<catalog>
  <book id="1">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
    with XML.</description>
  </book>
</catalog>
```

XML External Entity Injection (XXE)

XML Özellikleri - DTD



DTD (Document Type Definition), XML belgesinin yapısını ve niteliklerini tanımlar



Belgenin içerisinde tanımlanabilir veya başka bir kaynaktan dahil edilebilir.



SYSTEM operatörü
(http://, file://)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

XML External Entity Injection (XXE)

XML Özellikleri - Entity



Entity, XML'e dinamiklik katmak için tanımlanabilecek varlıklardır.



Internal, External Entity



SYSTEM operantına erişimi vardır!

DTD Example:

```
<!ENTITY writer "Donald Duck.">
```

```
<!ENTITY copyright "Copyright W3Schools.">
```

XML example:

```
<author>&writer;&copyright;</author>
```

XML External Entity Injection (XXE)



XML Injection'ın, XML'in Parsing yapıldığı anda meydana gelir.



XML verisine external entity tanımları dahil edilerek, Parsing sırasında XML verilerin işlenmesine müdahale edilmesine olanak tanıyan bir güvenlik zafiyetidir.

XML External Entity Injection (XXE) Etkileri



Dosya okuma

Örnek XML istek verisi:

```
<?xml version="1.0" encoding="UTF-8"?>  
<stockCheck><productId>381</productId></stockCheck>
```

External Entity dahil edilmesi:

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE demo [ <!ENTITY test SYSTEM "file:///etc/passwd" ]>  
<stockCheck><productId>&test;</productId></stockCheck>
```

Dönen cevap:

```
Invalid product ID: root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
...
```

XML External Entity Injection (XXE) Etkileri



SSRF

HTTP GET isteğinin yapılması

```
<!DOCTYPE demo [ <!ENTITY test SYSTEM  
"http://internal.vulnerable-website.com/"> ]>
```



DoS (Denial of Service)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
    <!ELEMENT foo ANY >
    <!ENTITY xxe SYSTEM "file:///dev/random" >]>
<foo>&xxe;</foo>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE bomb [
  <!ELEMENT bomb ANY>
  <!ENTITY fun "haha">
  <!ENTITY fun1 "&fun;&fun;&fun;&fun;&fun;&fun;&fun;&fun;">
  <!ENTITY fun2 "&fun1;&fun1;&fun1;&fun1;&fun1;&fun1;&fun1;&fun1;">
  <!ENTITY fun3 "&fun2;&fun2;&fun2;&fun2;&fun2;&fun2;&fun2;&fun2;">
  <!-- repeat many more times -->
]>
<bomb>&fun3;</bomb>
```

XML External Entity Injection



Dosya yükleme üzerinden XXE



SVG, ofis dosyaları (docx, ppt, xlsx)

facebook careers



Uygulama

XXE Injection

Örnek-1

Örnek-2

Örnek-3



XML External Entity Injection

Önlemler



XML Parserların, XML External Entity özelliği ve DTD işlemcileri devre dışı bırakılmalıdır

```
SAXParserFactory spf = SAXParserFactory.newInstance();  
spf.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true);  
spf.setFeature("http://xml.org/sax/features/external-general-entities", false);  
spf.setFeature("http://xml.org/sax/features/external-parameter-entities", false);
```