WEB saugumas

Marius Gžegoževskis

Turinys

- Pagrindiniai WEB saugumo mechanizmai aplikacijose.
- Dokumentų tipai: HTML, XML, JavaScript, AJAX, PHP.
- HTTP GET ir POST
- WEB pažeidžiamumo sritys.

Web saugumo mechanizmai

- Didžioji dauguma Web aplikacijų palaiko funkciškai panašius saugumo mechanizmus:
  - » Vartotojo teisių į programos duomenis ir funkcionalumą valdymas, siekiant apsaugoti nuo neautorizuotos prieigos.
  - » Vartotojo įvesties valdymas, siekiant užkirsti kelią specialiai žalingai suformuluotiems įvesties šablonams, kurie galėtų nepageidautinai paveikti sistemą.
  - » Tiesioginis reagavimas į atakas ar dažnus jų šablonus.
  - » Sistemos būsenos sekimas.

Vartotojo teisių valdymas

- Vartotojo teisių valdymą paprastai sudaro autentifikacija, sesijų palaikymas ir prieigos kontrolės. Šie trys komponentai sudaro fundamentalų bendro Web aplikacijos saugumo pamatą.

Įvesties valdymas

- Didžioji dalis Web aplikacijų pažeidžiamumų išnaudojama žalingai formuluojant įvestį. Tokiu atveju siekiama pakreipti aplikacijos veikimą nelaukta kryptimi. Vienas pagrindinių Web aplikacijos saugumo uždavinių yra saugus įvesties apdorojimas (input validation). Įvesties pažeidžiamumai Web aplikacijoje gali slėptis visur, todėl šiai užduočiai derėtų skirti ypač daug dėmesio.
- Pavyzdžiui: „<*scr*<*script*>*ipt*>“ nerekursyvaus filtro atveju siekiant išnaudoti XSS pažeidžiamumus.

Reagavimas į atakas

- Kuriant bei administruojant Web sistemą labai svarbu atitinkamai reaguoti į galimas jos atakas. Aplikacijos veikime iškilusios nenumatytos klaidos gali būti pirmas perspėjimas apie galimą ataką ar su ja susijusius ketinimus.
- Suinteresuoti asmenys dažnai specialiai siekia iššaukti šias klaidas sistemose, norėdami gauti daugiau informacijos apie galimus pažeidžiamumus.

Sistemos būsenos sekimas

- Beveik kiekviena Web sistema turi būti valdoma bei administruojama.
- Dažniausiai šį funkcionalumą savo ruožtu įgyvendina pati Web aplikacija, leisdama administratoriams valdyti vidinius sistemos duomenis, atlikti diagnostiką bei keisti svarbius nustatymus.
- Atsakingas administratorius turėtų reguliariai stebėti sistemą ir išaiškinti visas galimai su neautorizuota veikla susijusias anomalijas bei jų priežastis.

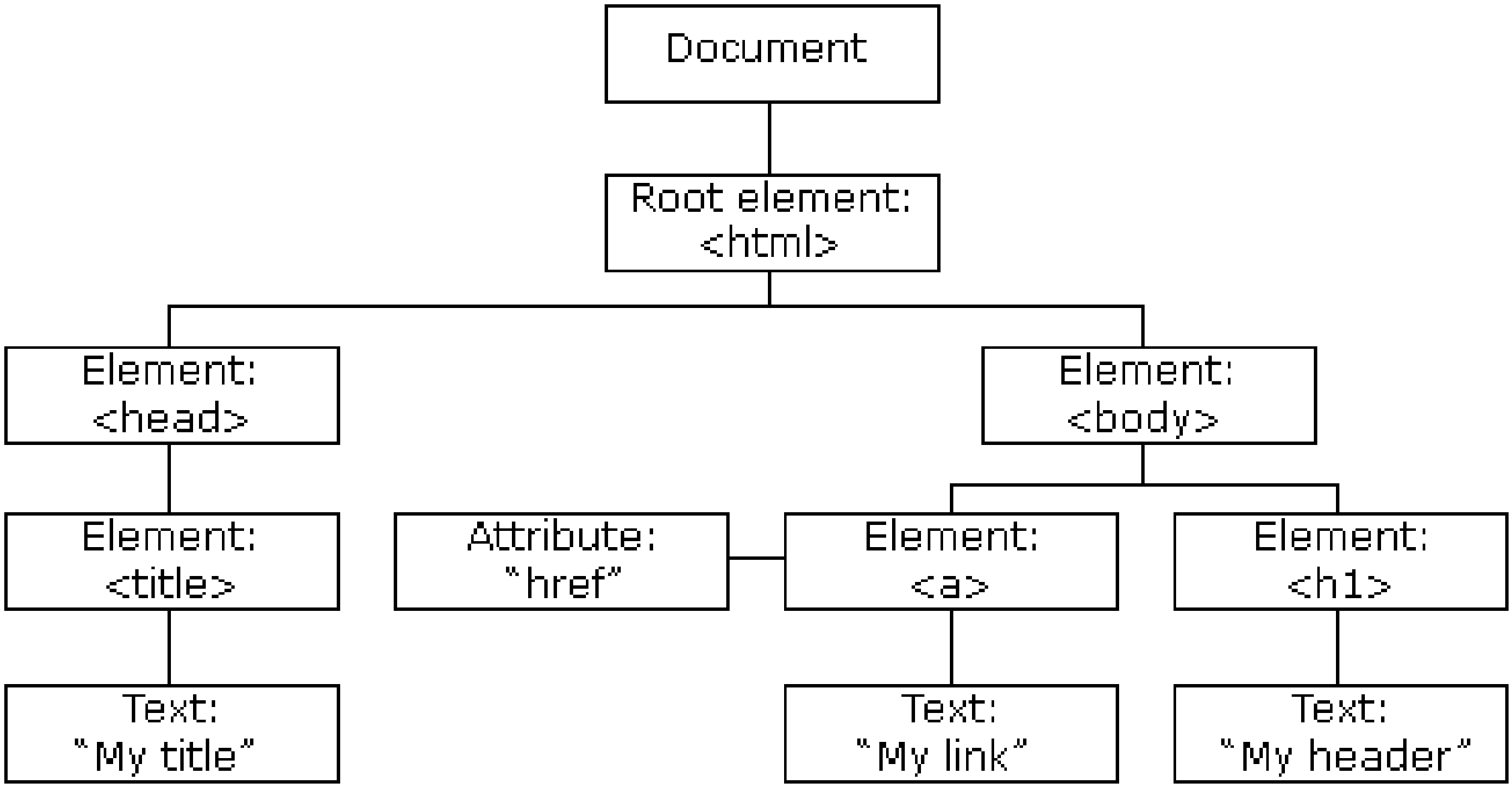
Dokumentų tipai

- HTML
- XML
- JavaScript
- PHP
- AJAX

HTML (Hyper Text Markup Language)

- Ne programavimo kalba, o žymių kalba.
- Suprojektuota interneto puslapių atvaizdavimui.
- HTML naršyklei „pasako“ ką reikia atvaizduoti (naršyklė pati nusprendžia kaip tai atvaizduoti).
- Naršyklė sukuria duomenų objektų modelį (DOM) iš HTML konteksto. Ignoruodoma neatpažįstamus „tagus“ ir atributus.

DOM (Document Object Model)

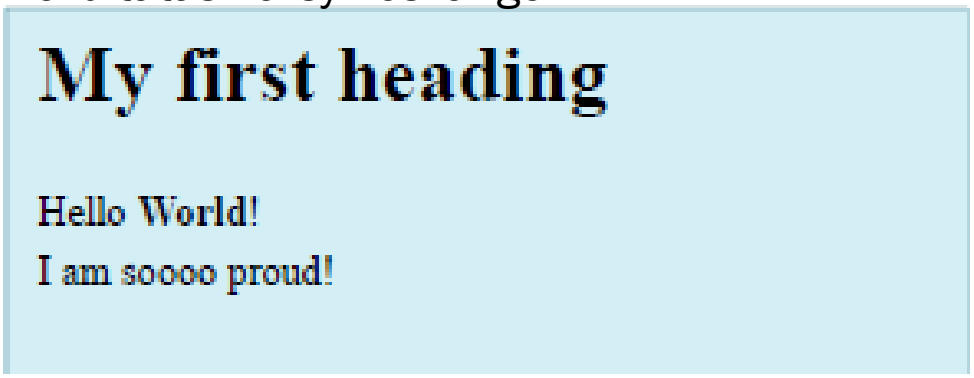


HTML pavyzdys

HTML dokumento tekstas

```
<!DOCTYPE html>
<html>
  <head>
    <title>My very first HTML page</title>
  </head>
  <body>
    <h1>My first heading</h1>
    Hello <b>World</b>!<br/>
    I am soooo proud!
  </body>
</html>
```

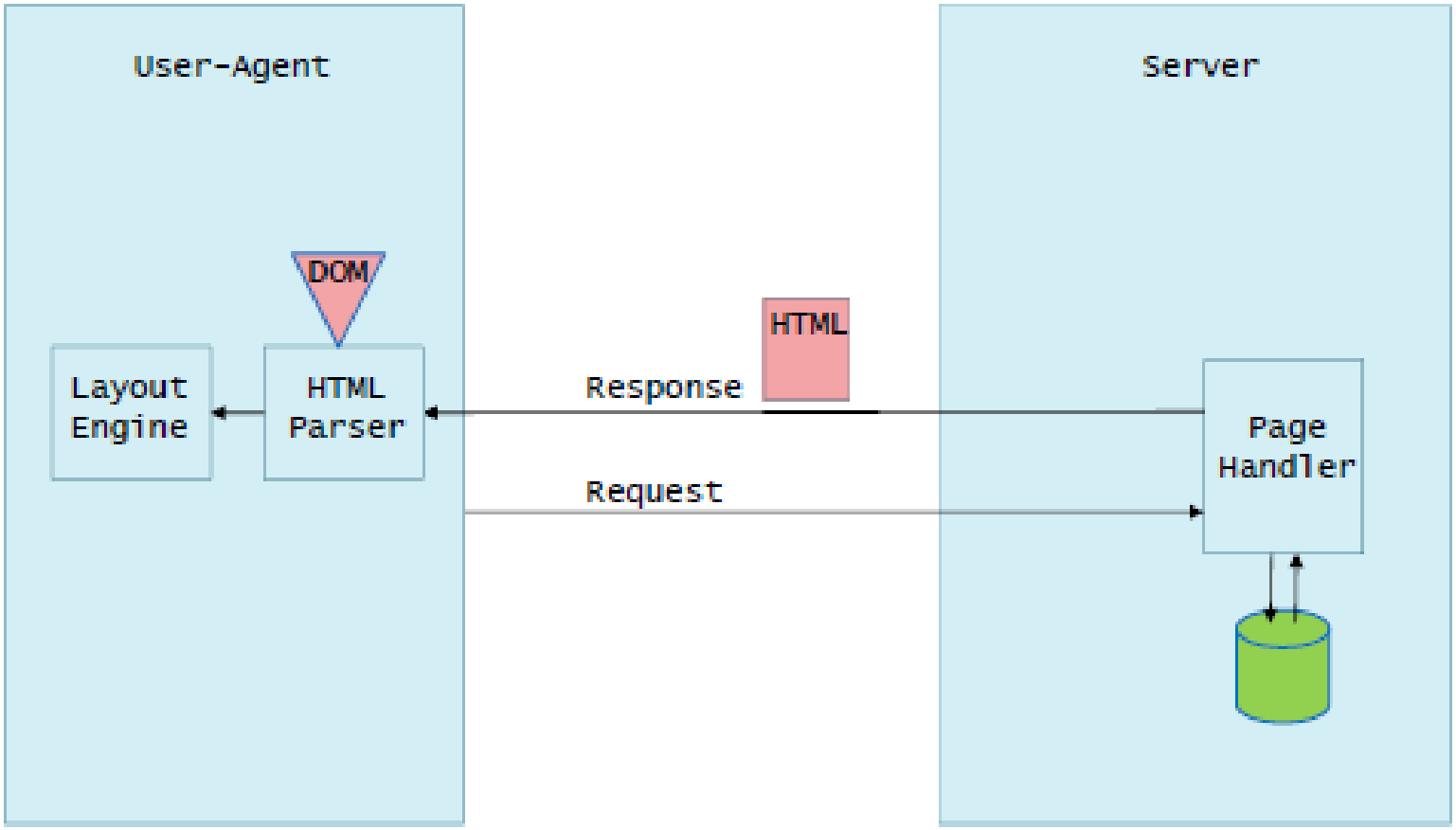
Rezultatas naršyklės lange



Komentarai:
HTML dokumentai taip pat vadinami internetiniais puslapiais.
Fiksuotas skaičius „tagų“.
Fiksuotas skaičius atributų.

Naršyklė:
Ignoruoja nežinomus tagus.

HTML veikimo principas



XML (eXtensible Markup Language)

- XML ne programavimo kalba -> ženklinimo kalba.
- Užtikrinti lengvesnį duomenų keitimąsi tarp skirtingo tipo sistemų.
- Bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.
- Pagrindinis XML kalbos vienetas yra *elementas, kuris gali turėti:*
  - norimą skaičių atributų.
  - kitus (dukterinius) šio elemento viduje esančius elementus.
  - su elementu susijusį tekstą.

XML pavyzdys

```
<asmenys>
    <asmuo pasonumeris="MP123456">
            <vardas>Vardenis</vardas>
            <pavarde>Pavardenis</pavarde>
    </asmuo>
    <asmuo pasonumeris="PK123456">
            <vardas>Jonas</vardas>
            <pavarde>Jonaitis</pavarde>
    </asmuo>
    <papildoma_informacija x="123"/>
</asmenys>
```

XML Elementų radimas dokumente

- XML elementai gali būti automatiškai randami:
  - pagal vardą;
  - pagal kelią: *asmenys/asmuo/pavarde/text()*. Rezultatas bus (Pavardenis, Jonaitis).
- XML kelių variantai aprašyti XPath standarte.
- Šiuo metu yra įvairioms kalboms skirtų bibliotekų dirbti su XML, todėl pačiam programuoti teksto analizės paprastai neprireikia.

XML Plėtimo galimybės (1 iš 2)

- Pagrindinis XML privalumas – galimybė pridėti naujo tipo elementus nepažeidžiant dokumento struktūros ir nesutrikdant tik seną formatą suprantančių programų darbo.

```
<asmuo pasonumeris="XX123456">
        <vardas>Petras</vardas>
        <pavarde>Petraitis</pavarde>
        <pastaba>Direktorius</pastaba>
</asmuo>
```

XML Plėtimo galimybės (2 iš 2)

- Turintį naują elementą *pastaba*, užklausa *asmenys/asmuo/pavarde/text()* dirbs kaip dirbusi, o elementas *pastaba* bus praleidžiamas.
- Programai besivystant, iškyla daug mažiau versijų nesuderinamumo problemų.

XML Vardų erdvės

- Jungiant kelis XML dokumentus į vieną, pasitaiko, jog sutampa skirtingą prasmę turinčių elementų vardai. Tuomet naudojamos vardų erdvės, pav <*autoinspekcija:asmuo*>, <*migracijos_tarnyba:asmuo*> ir pan. Prireikus XML dokumento antraštėje vardų erdvė apibrėžiama nurodant daug ilgesnį vardą, neretai interneto adresą. Elementą *asmenys* papildžius šiais duomenimis, dokumento pradžia atrodys taip:

```
<asmenys xmlns:autoinspekcija=http://autoinspekcija.com
 xmlns:migracijostarnyba="http://migracija.com">
```

JavaScript

- Suprojektuota pridėti interaktyvumą HTML puslapiams:
  - » Patobulinta vartotojo sąsaja.
  - » Dinaminis turinys.
- Skriptų kalba
  - » Programavimo kalba palaikantį skriptus.
  - » Skriptas = kodo eilutės, kurios yra interpretuojamos be kompiliatoriaus.
  - » Klientinė pusė: Kodas yra interpretuojamas naršyklėje.
  - » Javascript galimybės: Skaityti ir modifikuoti HTML, CSS, Duomenų validavimas iš įvesties (angl. input) formų.
  - » Saugoti bei gauti lokalią informaciją.
  - » Reaguoti į įvykius. (onClick(), mouseOver() ir t.t.).

JavaScript pavyzdys

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testing JavaScript</title>
    <script type="text/javascript">
      function writeText(txt) {
        document.getElementById("demo").innerHTML=txt;
      }
    </script>
    <noscript>
      JavaScript disabled or unsupported!
    </noscript>
  </head>
  <body>
    <h1>Event demo</h1>
    <button onclick="writeText('You did it!')">Press me</button>
    <p onmouseover="writeText('Don\'t touch the text!')" id="demo"></p>
</body>
</html>
```

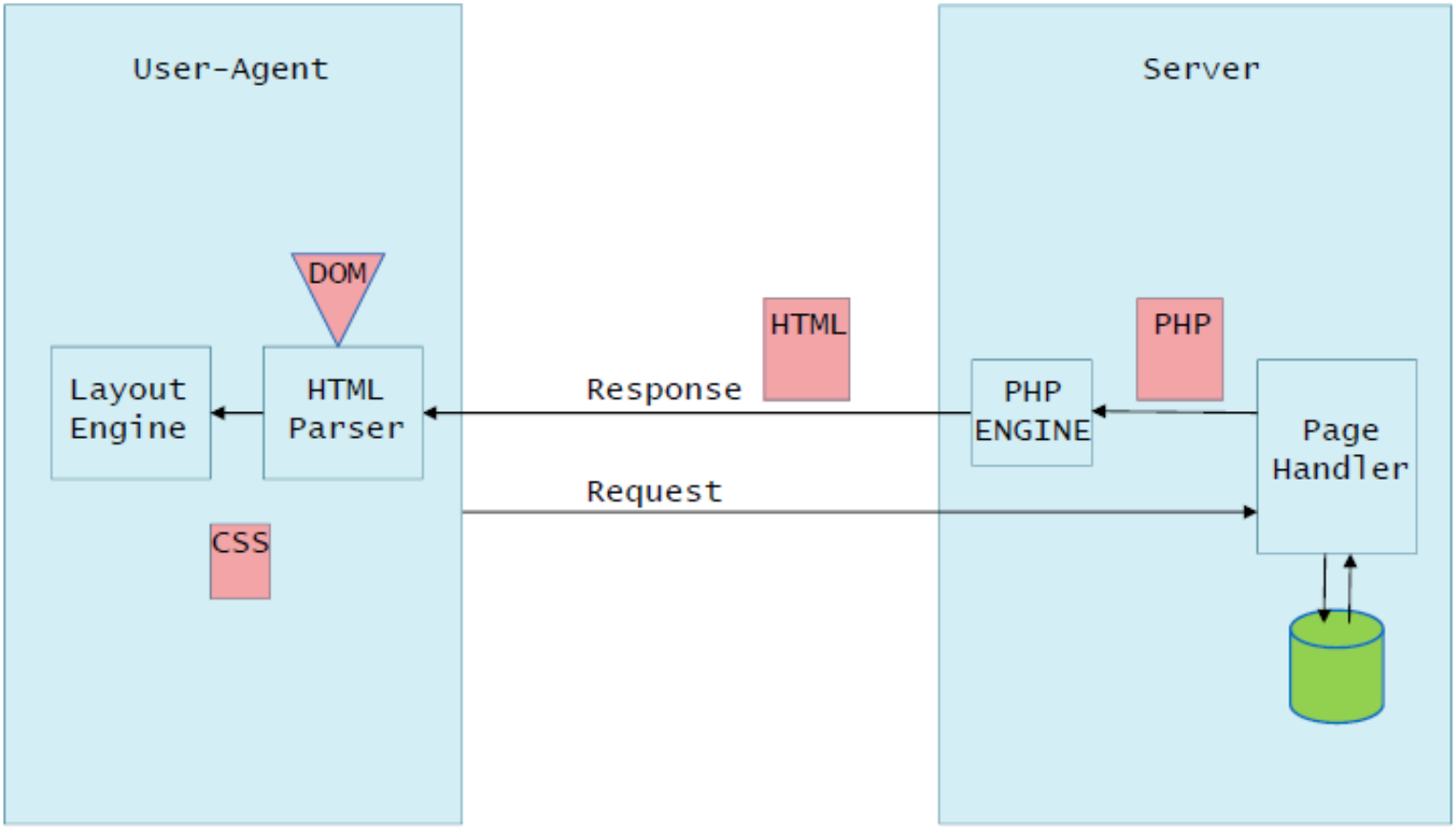
PHP

- PHP = PHP Hypertext Preprocessor
- ASP = Active Server Pages (Microsoft IIS)
- Serverio pusės skriptinės kalbos
  - » Kodas yra interpretuojamas serveryje – pateikiamas internetinis puslapis kaip rezultatas.
- ASP ir PHP gali:
  - » Dinamiškai modifikuoti ar pridėti turinį interneto puslapiams.
  - » Reaguoja į HTML formų užklausas.
  - » Prieiga prie duomenų bazių.
  - » Slepia kodą nuo kliento.
  - » Minimizuoja tinklo pralaidumą.

PHP pavyzdys

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php  //start PHP code
      echo “Hello World”;  #output text
    ?>
  </body>
</html>
```

PHP veikimo principas



AJAX

- AJAX = Asynchronous JavaScript and XML
  - » Ne programavimo kalba.
- Technika skirta duomenų apsikeitimui su serveriu atnaujinti puslapio tam tikras dalis, neperkraunant jo iš naujo.
- AJAX tai :
  - » Naudojama sparčių dinaminių internetinių puslapių kūrimui.
  - » Pagrindas interneto standartų.
  - » Nepriklausomas nuo platformos ar naršyklės.
- „Google suggest“ sukurtas AJAX pagrindu:
  - » Google Maps
  - » Gmail
  - » Youtube
  - » Facebook tabs

AJAX pavyzdys „Google suggest“ (pasiūlymai)

Google suggest: 1

Enter a name:

Suggestions:

Google suggest:2

Enter a name: e

Suggestions: Eva, Eve, Evita, Elizabeth, Ellen

Google suggest: 3

Enter a name: el

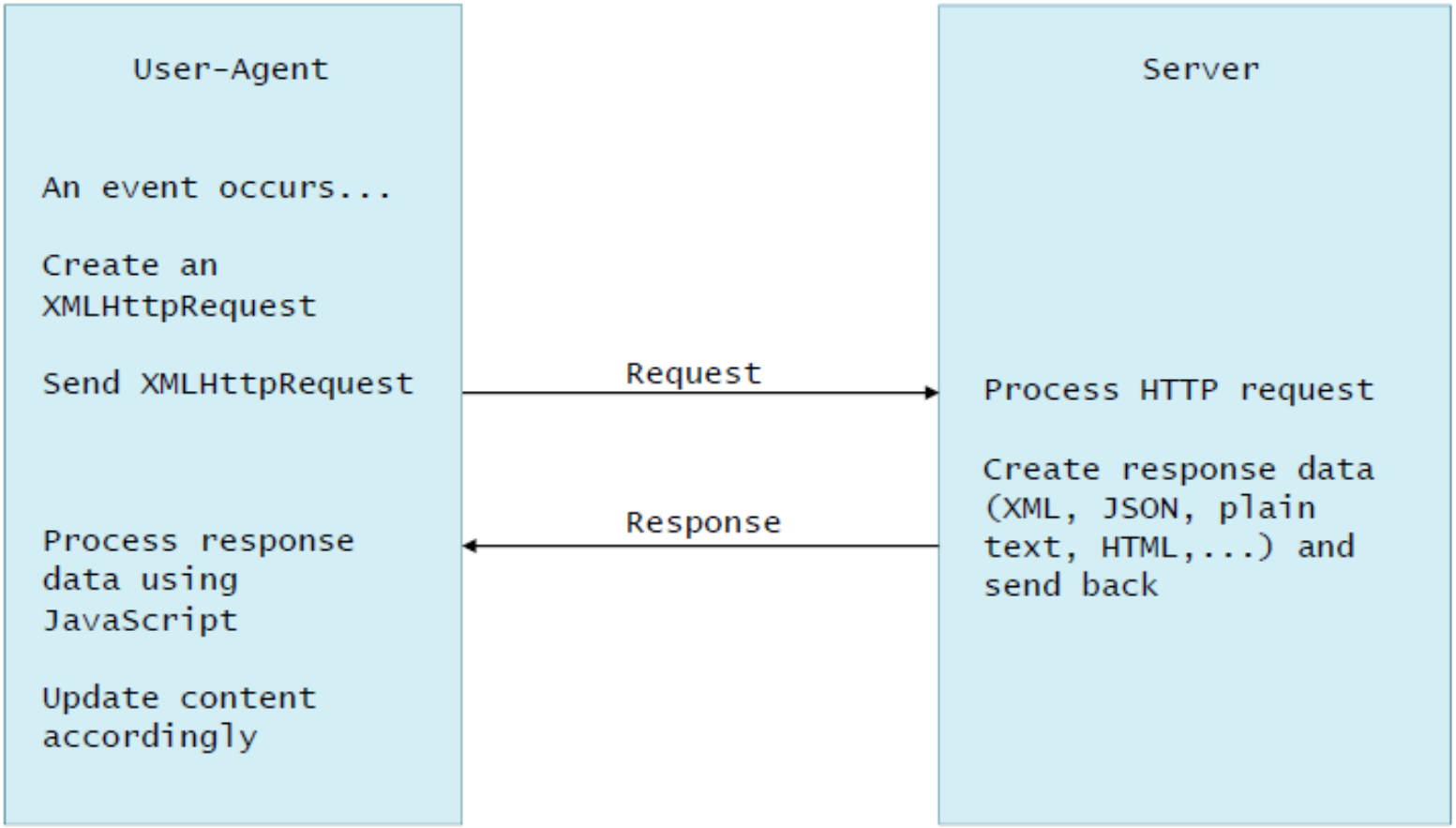
Suggestions: Elizabeth, Ellen

Google suggest: 4

Enter a name: ell

Suggestions: Ellen

AJAX pavyzdys (1 iš 2)



AJAX pavyzdys (2 iš 2)

```
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript">
    function showSuggestion(str) {
      var xmlhttp;
      if (str.length==0) {
        document.getElementById("sugg").innerHTML="";
        return;
      }
      xmlhttp=new XMLHttpRequest();
      xmlhttp.open("GET","gethint.asp?q="+str,true);
      xmlhttp.onreadystatechange=function() {
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
          document.getElementById("sugg").innerHTML=xmlhttp.responseText;
      }
      xmlhttp.send();
    }
  </script>
</head>
<body>
  <p><b>Enter a name:</b><input type="text" onkeyup="showSugg(this.value)"/><p/>
  <p>Suggestions: <span id="sugg" style="color:magenta"></span></p>
</body>
</html>
```

PHP superglobalūs kintamieji (1)

Superglobalūs kintamieji gali būti pasiekiami iš bet kur.

$_SERVER – Informacija iš serverio IP, antraštės

$_SERVER['REMOTE_ADDR'] gražiną užklaustą IP adresą.

$_SERVER['REMOTE_PORT'] gražiną portą.

$_SERVER['HTTP_USER_AGENT'] gražina naudojamos naršyklės informaciją.

$_SERVER['HTTP_REFERER'] gražina URL.

Serveris atsakingas už šituos taip pat superglobalius kintamuosius

$_GET, $_POST, $_COOKIE ir $_REQUEST

PHP superglobalūs kintamieji (2)

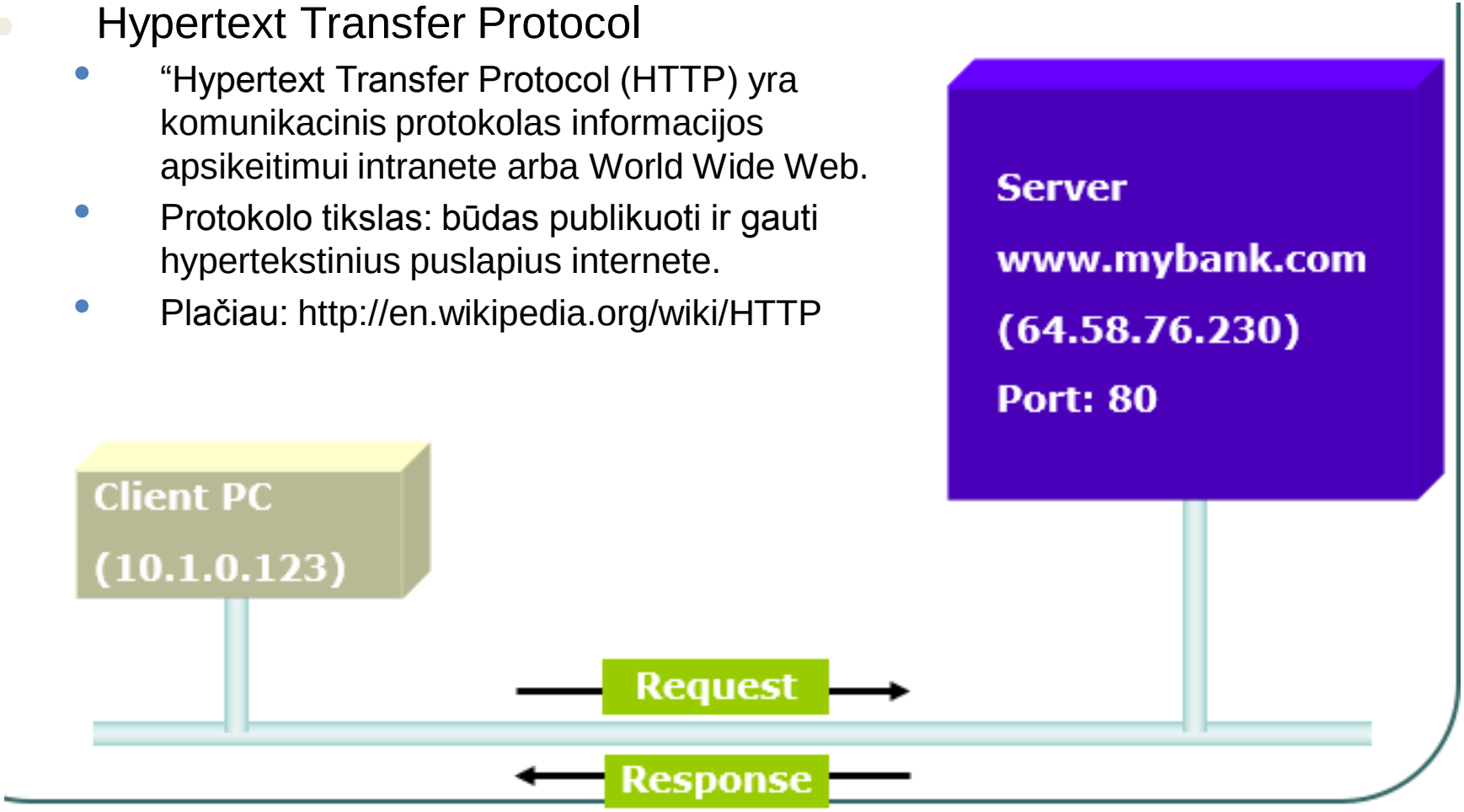
- Kintamieji yra siunčiami naudojant GET yra patalpinami superglobaliame kintamajame $_GET
- http://server.com?fname=John&lname=Doe
- $_GET[’fname’] gražins John
- $_GET[’lname’] gražins Doe

PHP superglobalūs kintamieji (3)

- Taip pat siunčiami kintamieji naudojant POST užklausą
- $_POST[’fname’] gražins John
- $_POST[’lname’] gražins Doe
- Cookie informacija yra saugoma $_COOKIE
- Jei nežinom ar nesvarbu kokia informacija bus saugoma tada naudojam $_REQUEST šis kintamasis turės visus 3: $_GET, $_POST ir $_COOKIE
- Cookies turi prioritetą pagal nutylėjimą.

HTTP (Hypertext Transfer Protocol)

- Hypertext Transfer Protocol
  - “Hypertext Transfer Protocol (HTTP) yra komunikacinis protokolas informacijos apsikeitimui intranete arba World Wide Web.
  - Protokolo tikslas: būdas publikuoti ir gauti hypertekstinius puslapius internete.
  - Plačiau: http://en.wikipedia.org/wiki/HTTP



HTTP Request - GET

- Formų duomenys yra koduojama URL.
- Vienas paprasčiausių HTTP metodų naudojamų internete.
- Turi būti naudojamas gauti informacijai, bet ne veiksmams, kurie turi atoveiksmį.

HTTP Request - GET

GET http://www.mysite.com/kgsearch/search.php?**catid=1** HTTP/1.1

Host: www.mysite.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.mysite.com/

HTTP Request - GET

- http://www.google.com/search?hl=en&lr=&c2coff=1&rls=GGLG%2CGGLG%3A2005-26%2CGGLG%3Aen&q=http%3A%2F%2Fwww.google.com%2Fsearch%3Fhl%3Den%26lr%3D%26c2coff%3D1%26rls%3DGGLG%252CGGLG%253A2005-26%252CGGLG%253Aen%26q%3Dhttp%253A%252F%252Fwww.google.com%252Fsearch%253Fhl%253Den%2526lr%253D%2526c2coff%253D1%2526rls%253DGGLG%25252CGGLG%25253A2005-26%25252CGGLG%25253Aen%2526q%253Dhttp%25253A%25252F%25252Fwww.google.com%25252Fsearch%25253Fsourceid%25253Dnavclient%252526ie%25253DUTF-8%252526rls%25253DGGLG%25252CGGLG%25253A2005-26%25252CGGLG%25253Aen%252526q%25253Dhttp%2525253A%2525252F%2525252Fwww%2525252Egoogle%2525252Ecom%2525252Fsearch%2525253Fsourceid%2525253Dnavclient%25252526ie%2525253DUTF%2525252D8%25252526rls%2525253DGGLG%2525252CGGLG%2525253A2005%2525252D26%2525252CGGLG%2525253Aen%25252526q%2525253Dhttp%252525253A%252525252F%252525252Fuk2%252525252Emultimap%252525252Ecom%252525252Fmap%252525252Fbrowse%252525252Ecgi%252525253Fclient%252525253Dpublic%2525252526GridE%252525253D%252525252D0%252525252E12640%2525252526GridN%252525253D51%252525252E50860%2525252526lon%252525253D%252525252D0%252525252E12640%2525252526lat%252525253D51%252525252E50860%2525252526search%252525255Fresult%252525253DLondon%25252525252CGreater%252525252520London%2525252526db%252525253Dfreegaz%2525252526cidr%252525255Fclient%252525253Dnone%2525252526lang%252525253D%2525252526place%252525253DLondon%252525252CGreater%252525252BLondon%2525252526pc%252525253D%2525252526advanced%252525253D%2525252526client%252525253Dpublic%2525252526addr2%252525253D%2525252526quicksearch%252525253DLondon%2525252526addr3%252525253D%2525252526scale%252525253D100000%2525252526addr1%252525253D%2526btnG%253DSearch%26btnG%3DSearch&btnG=Search

HTTP Requests - POST

- Duomenys yra iterpti užklausos viduje.

- Yra naudojamas veiksmams kurie turi atoveiksmį:
  - » Talpinti informaciją.
  - » Atnaujinti duomenis.
  - » Užsakyti produktus
  - » Ir kiti.

HTTP Requests - POST

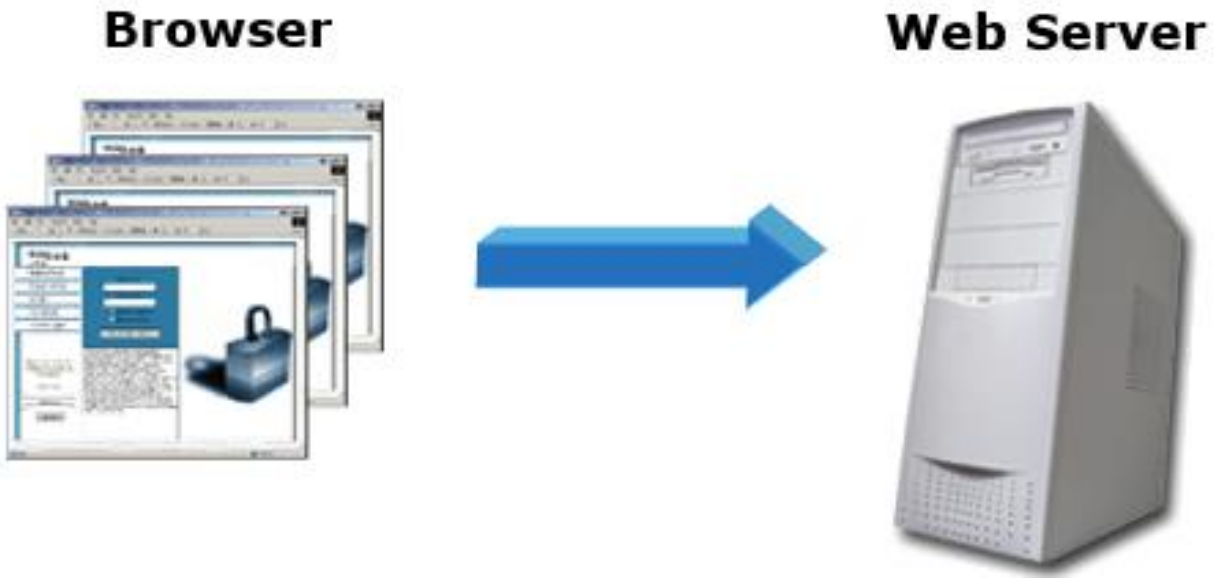

```
POST http://www.mysite.com/kgsearch/search.php HTTP/1.1
Host: www.mysite.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.mysite.com/

catid=1
```

GET vs. POST Security

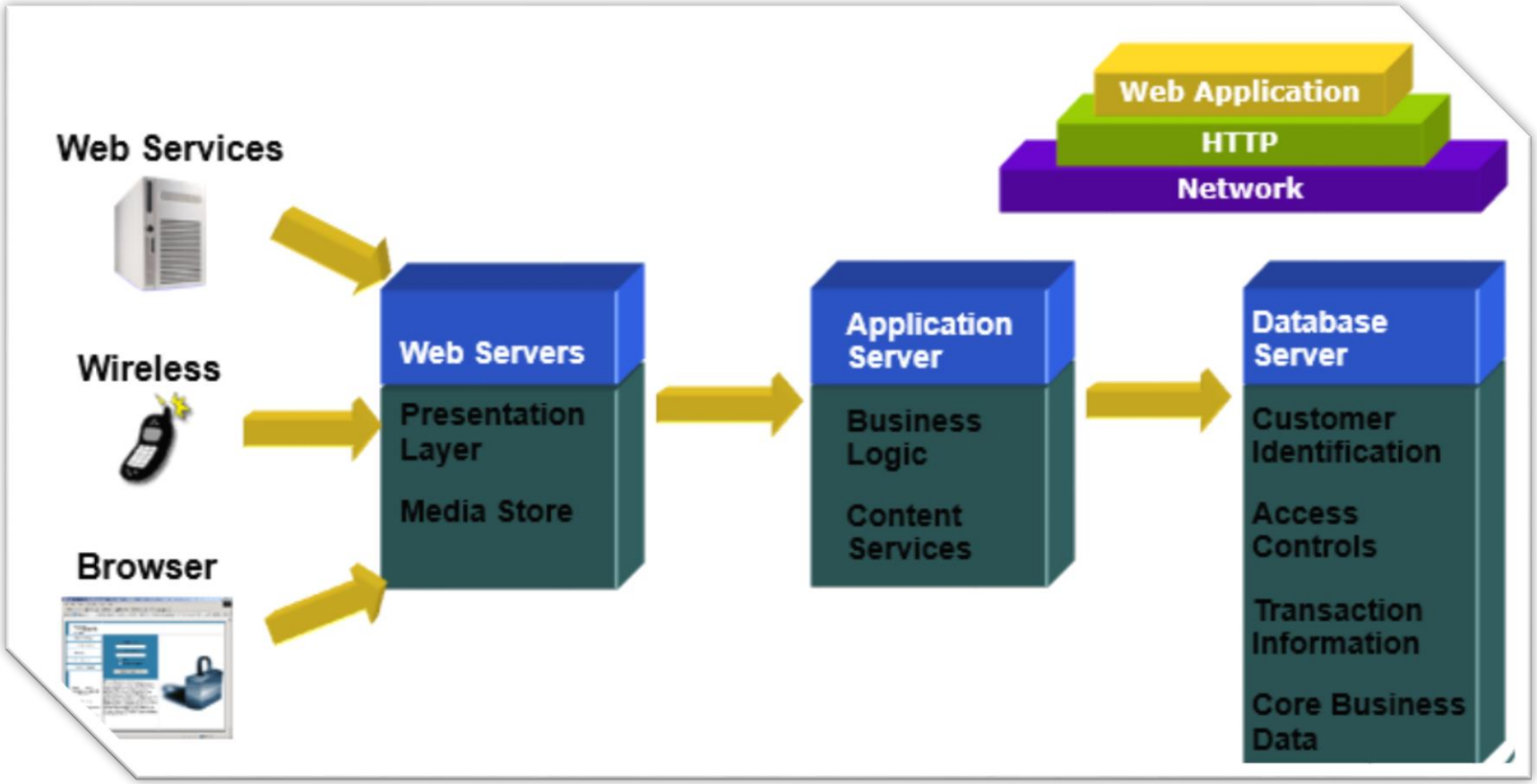
- Informacija saugoma parametruose gali nusakyti aplikacijos veikimo principą.
- GET parametrai yra kuo puikiausiai matomi naršyklės adresų juostoje.
- POST parametrai yra paslėpti nuo vidutinio lygio vartotojų:
  - » Vartotojai gali peržiūrėti šaltinio kodą.
  - » Vartotojai gali peržiūrėti paketus.
  - » Vartotojai gali isiterpti ir modifikuoti web užklausas.

Web svetainės

- **Be aplikacijų**
- **Statinis turinys**
- **Rankiniu būdu surašytos nuorodos.** (Angl. Hard coded links)



WEB aplikacijos



Kodėl WEB aplikacijos yra pažeidžiamos

Security Professionals Don't Know The Applications

"As a Network Security Professional, I don't know how my companies web applications are supposed to work so I deploy a protective solution…but don't know if it's protecting what it's supposed to."

The Web Application Security Gap

Application Developers and QA Professionals Don't Know Security

"As an Application Developer, I can build great features and functions while meeting deadlines, but I don't know how to develop my web application with security as a feature."

WEB aplikacijų pažeidžiamumai

Keletas WEB aplikacijų pažeidžiamumų atsiradimo priežąsčių:

- Neatidžiai suprogramuota aplikacija.
- Maži aplikacijos pakeitimai reikalauja pakeisti tam tikrą sritį.
- Programos logika.
- Kas dar galėtų įtakoti aplikacijų pažeidžiamumą?