CARNet

HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Sigurnost Drupal CMS sustava

NCERT-PUBDOC-2010-02-291

Nacionalni CERT

u suradnji s

LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem Nacionalni CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom Nacionalnog CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

Nacionalni CERT, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

CMS sustavi sve se češće koriste za upravljanje *web* stranicama. Među najčešće korištenima je i besplatan CMS sustav Drupal. Riječ je o sustavu koji omogućuje upravljanje *web* sjedištima koji se sastoje od jedne ili više stranica. Rad sa sustavom odvija se preko *web* preglednika. Budući da su *web* stranice i preglednici jedna od najčešćih meta napada na Internetu, sigurnost je jedno od važnijih pitanja koja se vežu uz ovaj alat. Osnovni dio Drupala je jezgra koja, između ostalog, sadrži određene sigurnosne mehanizme, a može se i nadograđivati dodatnim modulima. Mnogi od njih su oblikovani kako bi poboljšali sigurnosne značajke sustava.

Među najčešćim napadima na Drupal su XSS i CSRF napadi te podmetanje SQL koda. Sva tri napada najčešće su posljedica nedovoljne provjere podataka koje korisnik predaje stranici. Zato je filtriranje sadržaja jedna od najvažnijih sigurnosnih metoda. Drupal podržava filtriranje sadržaja ograničavanjem i provjerom ovlasti korisnika. Osim toga moguće je definirati vlastiti kod i stvoriti vlastite liste ovlasti. Osim jezgre i modula koji nude sigurnosne mehanizme, te programskih API-ja za stvaranje vlastitih mehanizama, moguće je u konačnici ispitivati Drupal pomoću posebno oblikovanih sigurnosnih alata. Oni su predstavljeni na kraju ovog dokumenta.

2. CMS Drupal

CMS (eng. *Content Management System*) sustav uključuje skup alata koji olakšavaju upravljanje podacima. Pod upravljanjem se podrazumijeva dijeljenje podataka između različitih subjekata u poslovnom procesu te upravljanje izmjenama i tokovima podataka. CMS sustavi provjeravaju prava pristupa određenom skupu podataka, olakšavaju njihov dohvat i pohranjivanje te čine jednostavnijom komunikaciju između korisnika te izradu izvještaja. Primjer CMS sustava za upravljanje sadržajem *web* sjedišta je i programski paket Drupal.

2.1. CMS sustavi

Općenito, sadržaji kojima upravlja neki CMS sustav mogu biti dokumenti, multimedijalni sadržaji, znanstveni sadržaji i slično. U poslovnim okruženjima često se upravlja sadržajem dokumenata vezanih uz organizacijske procese. Cilj poslovnih CMS-ova je upravljanje nestrukturiranim informacijama u različitim formatima. Također, moguće je upravljati elektroničkim dokumentima ili mobilnim sadržajima. Posebna vrsta CMS sustava je *web* CMS, a koristi se za upravljanje podacima na *web* stranicama, odnosno *web* sjedištima (skup tematski i sadržajno povezanih *web* stranica).

2.1.1. *Web* CMS

Web CMS je sustav koji se koristi za upravljanje HTML sadržajem *web* stranica. Olakšava stvaranje, uređivanje i održavanje velike količine dinamičkog *web* sadržaja. Ovakvi sustavi često su oblikovani s ciljem da se upravljanje sadržajem omogući korisnicima koji nemaju veliko znanje o programiranju i programskim jezicima, odnosno korisnicima s manjim tehničkim predznanjem. U tome je razlika između izgradnje i uspostavljanja *web* stranica koje zahtjeva programerske vještine te održavanja koje se izvodi preko *web* CMS sustava. Administracija se pritom najčešće obavlja putem *web* preglednika. *Web* CMS sustavi najčešće uključuju:

- automatizirane predloške koji olakšavaju izmjenu cjelokupne vizualne prezentacije sadržaja,
- odvojen prezentacijski sloj od sadržaja što omogućuje lakše upravljanje,
- programske dodatke za proširivanje funkcionalnosti,
- upravljanje radnim tokom, odnosno ciklusima događaja na stranici. Primjerice, aktivnost nekog korisnika može se zabilježiti, ali ne i objaviti izmjenom stranice prije nego drugi korisnik to odobri,
- delegiranje ovlasti i obaveza različitim korisnicima dodjelom uloga,
- upravljanje dokumentima (stvaranje, revidiranje, objavljivanje, arhiviranje i brisanje dokumenata),
- virtualizaciju sadržaja koja omogućuje provjeru učinka određene akcije prije nego se ona zaista izvede u izvornom sustavu,
- stvaranje RSS (eng. *Really Simple Syndication*) obavijesti za druge sustave i korisnike o izmjenama i novim sadržajima stranice.

2.1.2. Sigurnosna pitanja

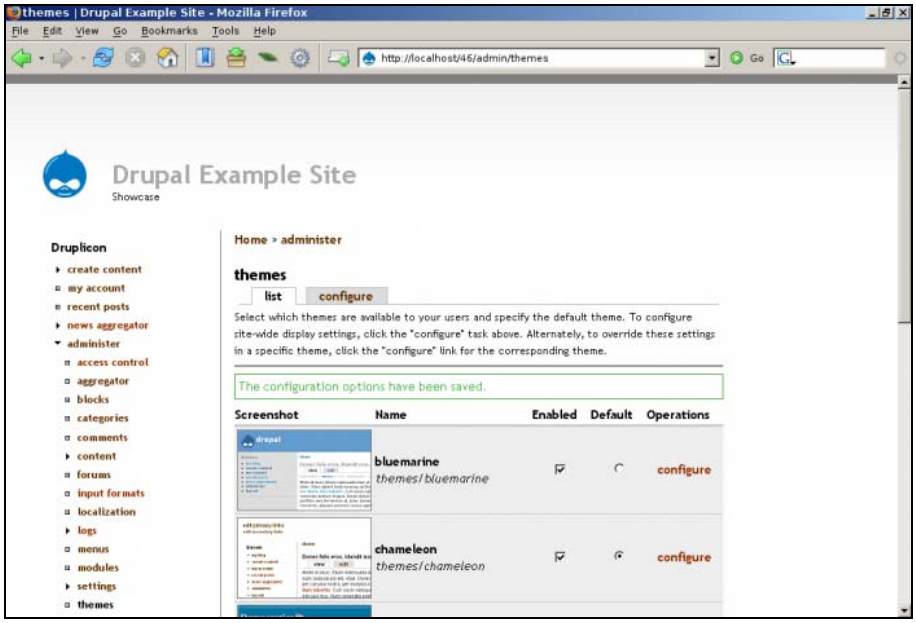
Sigurnosna pitanja *web* CMS sustava najčešće su vezana uz ranjivosti *web* stranica kojima se upravlja. To su primjerice XSS (eng. *Cross Site Scripting*) i CSRF (eng. *Cross Site Request Forgery*) ranjivosti, zatim provjera identiteta i ovlasti korisnika koji se prijavljuju na sustav te rukovanje zahtjevima prema stranici. Dio ranjivosti proizlazi iz činjenice da *web* CMS ne funkcionira sam, već ovisi o radu *web* poslužitelja i baze podataka s kojom komunicira. Osim toga, podaci o korisničkim sjednicama mogu biti sačuvani u kolačićima i zloupotrijebljeni za stjecanje neovlaštenog pristupa. Spomenuto je da se CMS sustavu pristupa preko sučelja *web* preglednika. Poznato je kako su *web* preglednici česta meta internetskih napada pa su brojne ranjivosti posljedica upravo pogrešaka u radu *web* preglednika.

2.2. Drupal

Drupal je besplatan CMS sustav otvorenog programskog koda namijenjen upravljanju *web* stranicama. Koristi se za stranice različite složenosti i veličine, od osobnih internetskih dnevnika (eng. *blog*) do poslovnih *web* sjedišta. Kao besplatan alat pušten je u distribuciju 2001. godine, a danas ga razvija velik

tim programera i njegova popularnost iz dana u dan raste. Prema informaciji s Wikipedije, na *web* sjedištu Drupal.org registrirano je preko 650 tisuća korisnika, od toga 2 tisuće programera.

Drupal omogućuje rad preko jednostavnog osnovnog sučelja koje ne zahtjeva programerske vještine, ali pruža i mogućnost rada preko sofisticiranog sučelja za napredne korisnike. Sustav je razvijen u programskom jeziku PHP, a može se pokretati na različitim platformama koje imaju *web* poslužitelje s podrškom za programski jezik PHP (Apache, IIS, Lighttpd, nginx) i baze podataka (MySQL, PostgreSQL) za upravljanje sadržajem.



Slika 1. Drupal sustav
Izvor: Drupal.org

Drupal se sastoji od jezgrenog modula koji sadrže osnovne CMS funkcionalnosti, a može se nadograđivati različitim paketima koji uvode naprednije mogućnosti korištenja. Jezgra Drupal sustava uključuje sljedeće mogućnosti:

- prijavu na sustav,
- napredno pretraživanje,
- blogove, forume, komentare i ankete,
- mogućnost pohranjivanja sadržaja u priručnu memoriju radi poboljšanja performansi,
- korištenje opisnih URL adresa,
- ugnježđivanje izbornika,
- upravljanje sadržajem za veći broj korisnika,
- RSS,
- automatsko obavještavanje administratora o dostupnim (sigurnosnim) nadogradnjama,
- stvaranje korisničkih profila s različitim ulogama, adresama i sl.,
- upravljanje radnim tokom pomoću mehanizama okidača i akcija,
- određeni broj tematskih predložaka vizualnog izgleda stranice koje administrator može odabrati.

2.2.1. Dodatni moduli

Jezgra Drupal sustava oblikovana je tako da se lako može nadograđivati paketima koji mogu uključivati nove funkcionalnosti, ali i mijenjati izvorno ponašanje bez izmjene izvornog koda aplikacije. Dostupno je preko 4000 Drupal modula koji sadrže specifične uređivače sadržaja, alate za razmjenu privatnih poruka i slične dodatne mogućnosti, zatim sigurnosne alate i različita druga programska ostvarenja. Primjeri poznatih često korištenih modula su:

- CCK (eng. *Content Construction Kit*) - omogućuje administratorima dinamičko stvaranje tipova sadržaja pomoću kojih se bilo koja vrsta podataka može pohraniti u bazu.
- *Views* – olakšava dohvat i prezentaciju sadržaja,
- *Panels* – omogućuje administratorima vizualni dizajn proizvoljnog prikaza sadržaja.

Osim toga, velik broj paketa nadograđuje postojeće vizualne predloške (eng. *themes*) Drupal jezgre te tako proširuje mogućnosti vizualne prezentacije stranice.

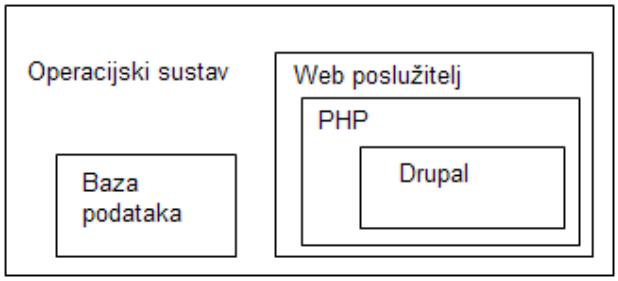
3. Općenito o sigurnosti Drupal sustava

Drupal je, kao i svaki sustav, prvenstveno osjetljiv na stjecanje neovlaštenog pristupa ukoliko se mehanizmi provjere identiteta korisnika ne koriste ispravno. Druge ranjivosti vezane su uz specifičnost uporabe, a to je u slučaju CMS sustava upravljanje *web* stranicama. *Web* stranice meta su XSS, CSRF napada, krađe korisničkih sjednica i osjetljivih podataka pohranjenih u pregledniku.

U ovom poglavlju dan je pregled osnovnih sigurnosnih mjera dostupnih u Drupal sustavu. One uključuju sigurnu i pažljivu administraciju sustava, nadogradnju sigurnosnim paketima i redovito održavanje te praćenje sigurnosnih upozorenja i noviteta.

3.1. Administracija sustava

Odgovorna administracija prva je i osnovna razina na kojoj počinje briga za sigurnost. Pritom se kod Drupal sustava ona proširuje na bazu podataka i *web* poslužitelj uz pomoć kojih se odvija rad.



Slika 2. Jednostavna Drupal instalacija

Osnovna sigurnosna pravila u administraciji *web* sustava Drupal uključuju:

- zaštitu pristupa datotekama,
- omogućavanje HTTPS protokola,
- korištenje dodatnih sigurnosnih modula,
- otklanjanje mogućnosti pokretanja proizvoljnog PHP koda,
- pažljivu provjeru korisničkog unosa i
- sakrivanje podataka od korisnika.

Web poslužitelj na kojem se pokreće sustav ne smije imati prava pisanja u Drupal datoteke koje upravljaju sadržajem stranice. U suprotnom moglo bi se dogoditi da se napadom na *web* stranicu podmetne zlućudan kod u neku od datoteka preko kojih se izvodi stranica.

Omogućivanje HTTPS protokola važno je u slučaju slanja osjetljivih podataka kao što su lozinke. Pritom je potrebno postaviti digitalni certifikat i konfigurirati *web* poslužitelj tako da se omogući enkripcija. U Drupalu se HTTPS omogućuje dodavanjem linije

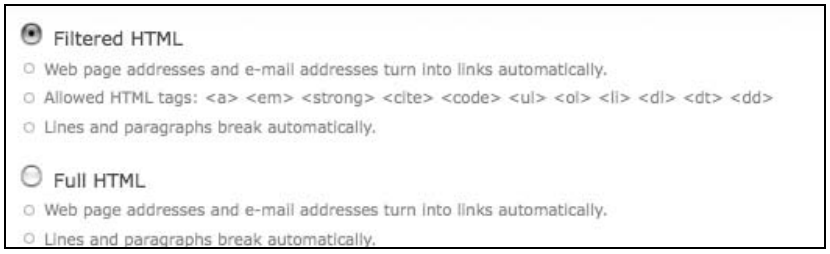
```
$conf['https'] = TRUE;
```

u datoteku *„sites/default/settings.php“*.

Korištenjem dodatnih sigurnosnih modula (npr. OpenPGP, *No Anonymous Session*, Swekey *hardware authentication*, *Salted Passwords* i drugi) Drupal omogućuje pouzdaniju provjeru identiteta korisnika, zaštitu od neželjenih sadržaja, otkrivanje te otklanjanje ranjivosti i slične mogućnosti.

Otklanjanje mogućnosti pokretanja nesigurnog i neprovjerenog PHP koda preko stranice može se izvesti dodjelom odgovarajućih ovlasti korisničkim računima ili potpunim isključivanjem ove mogućnosti za sve korisnike.

Pažljiva provjera korisničkog unosa obavlja se pomoću postavki posebnih filtara. Primjerice postavka „*Full HTML*" omogućuje korisniku postavljanje bilo kakvih slika ili *JavaScript* koda na stranicu. Vrste sadržaja mogu se i ograničiti definiranjem dopuštenih HTML oznaka i blokova (*SCRIPT, LINK, META*…). Mijenjanje postavki filtara ulaznog sadržaja i poretka filtara zbog svoje osjetljivosti treba se izvoditi s posebnom pažnjom.



Slika 3. Filtririranje HTML-a u Drupalu
Izvor: Cracking Drupal

Skrivanje podataka od korisnika uključuje zaštitu direktorija i oblikovanje poruka o pogreškama koje neće otkrivati informacije o sustavu.

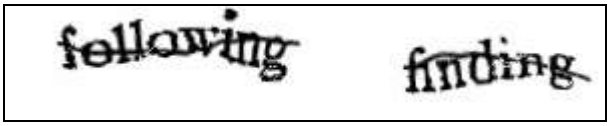
3.2. Sigurnosni moduli

Drupal sigurnosni moduli dijele se u osnovne kategorije:

- sigurnost,
- provjera identiteta korisnika (autentikacija) i
- zaštita od neželjenih poruka (eng. *spam*).

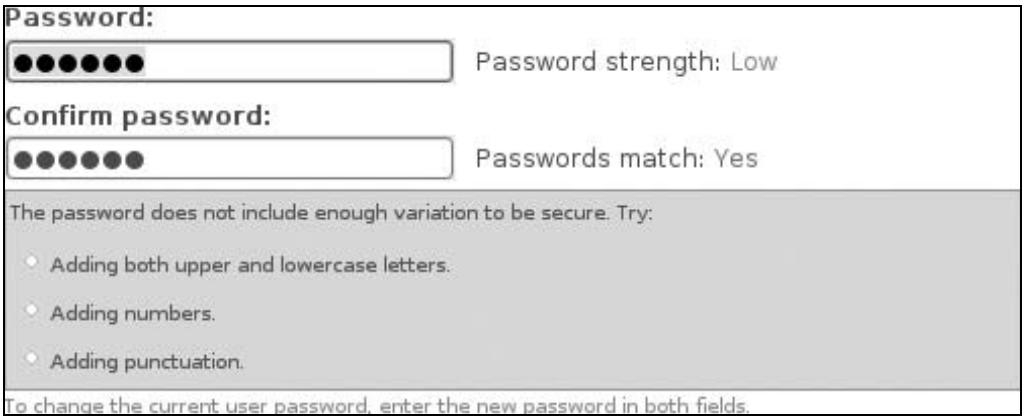
Raspon usluga sigurnosnih modula vrlo je širok, a u sljedećim natuknicama dani su neki primjeri:

- zaštićena komunikacija – modul *Client Side Encryption* uključuje metode kriptiranja i dekriptiranja poruka,
- upravljanje korisničkim sjednicama (njihovim trajanjem, brojem i dr.) – modul *No Anonymous Session* onemogućuje uspostavljanje sjednica za anonimne korisnike,
- upravljanje lozinkama uključuje zahtijevanje snažnih lozinki te određuje broj dozvoljenih krivih unosa – modul *Password Strength* provjerava i zahtjeva lozinke određene složenosti,
- unaprjeđenje autentifikacije korisnika dodatnim sigurnosnim mehanizmima – modul *Certificate Login* provjerava korisnike na temelju certifikata,
- upravljanje neželjenim porukama – modul *CAPTCHA* otklanja automatske *spam* poruke tako što zahtjeva prijepis teksta s ekrana,



Slika 4. Primjer Captcha teksta za provjeru
Izvor: Wikipedia

- provjera IP adresa – modul *GoAway* zabranjuje pristup zadanim IP adresama,
- zaštita podataka o korisnicima – modul *IP Anonymize* briše informacije o IP adresama korisnika,
- određivanje zakonskih prava i uvjeta korištenja – modul *Legal* prikazuje uvjete korištenja korisnicima koji se žele registrirati,
- otkrivanje i otklanjanje ranjivosti – modul *MD5 Check* stvara i provjerava MD5 sažetke



Slika 5. Alat za provjeru jačine i složenosti lozinke
Izvor: Cracking Drupal

3.3. Održavanje sigurnosti sustava

Tim programera koji se brine za razvoj i održavanje Drupal paketa redovito izdaje programske ispravke za uočene programske propuste, uključujući i sigurnosne probleme. Osim toga, stalno se razvijaju nove inačice Drupala s ciljem poboljšanja učinkovitosti i funkcionalnosti. Kod nadogradnje paketa može se raditi o nadogradnji nekog od modula ili jezgre Drupala programskim dodatkom koji sadrži određene ispravke, ili se cijeli sustav može nadograditi na noviju inačicu, kao npr. prijelaz s inačice 5.x na 6.x. Prije svake nadogradnje savjetuje se stvaranje sigurnosne kopije (eng. *backup*).

Redovita nadogradnja smanjuje mogućnost zlouporabe. Od inačice Drupal 6.x, jezgra uključuje i modul *„Update status"* koji redovito provjerava jesu li dostupne novije inačice bilo kojeg od modula koji se koriste i obavještava administratora.

4. Specifične ranjivosti Drupala

U prethodnom poglavlju navedeni su općeniti načini za brigu o sigurnosti Drupal sustava. U ovom poglavlju tema su specifične ranjivosti Drupala kao CMS sustava. U drugom dijelu poglavlja dani su i neki naputci o preporučenim postupcima i funkcijama koje valja koristiti kako bi se minimizirale opasnosti. Osim toga, na kraju su predstavljena i tri alata za ispitivanje koda Drupal sustava na ranjivosti.

4.1. Pregled ranjivosti Drupal sustava

Kod Drupal sustava najčešće su specifične ranjivosti vezane uz *web* stranice, kao što su CSRF i XSS ranjivosti. Osim toga javljaju se problemi podmetanja SQL koda i stjecanja neovlaštenog pristupa. Ti su problemi vezani uz nedovoljnu provjeru podataka koje korisnik predaje stranici te uz nesiguran postupak prijave na sustav i autentikacije. U tablici koja slijedi prikazana je statistika pojave ranjivosti iz koje je vidljivo kako su navedena četiri problema dominantna kod Drupala.

Ranjivost	Broj pojava	Udio u ukupnim ranjivostima
XSS	55	44
Neovlašten pristup	17	14
CSRF	12	10
Podmetanje SQL koda	12	10
Pokretanje koda	10	8
Objašnjena i objave	4	3
Postavljanje sjednice	3	2
Podizanje ovlasti	2	5
Slanje proizvoljnih datoteka	2	5
Podmetanje zaglavlja elektroničke pošte	2	5
Zaobilaženje provjere CAPTCHA modula	2	5
Rastavljanje HTTP odgovora	2	4
Prepisivanje datoteka	1	2
Otkrivanje osjetljivih podataka	1	2
Oponašanje sjednice	1	2

Tablica 1. Ranjivosti Drupala
Izvor: Cracking Drupal

4.1.1. XSS

XSS je vrsta ranjivosti koja uključuje mogućnost umetanja koda u stranicu preko polja predviđenih za unos podataka. Pritom taj kod može mijenjati podatke s ovlastima korisnika koji ga je pokrenuo što uključuje primjerice i promjenu lozinke. XSS se najčešće izvodi preko *JavaScript* koda, ali može i preko bilo kojeg drugog programskog jezika. XSS napadi mogu:

- prikazati podatke iz zahtjeva za stranicom koje je korisnik upravo unio,
- pohraniti podatke koje je korisnik unio u bazu podataka,
- izmijeniti programski kod.

Primjer ovakvog napada je podmetanje HTML koda u podatkovno polje ranjive stranice. Taj se kod nakon predaje podataka pokreće na stranici, a može primjerice uključivati prikaz lažnog obrasca za prijavu korisnika. Takav podmetnuti obrazac upisano ime i lozinku korisnika može predati napadaču umjesto ranjivoj stranici.

Za zaštitu od ove vrste napada važno je provjeravati podatke koje korisnik unosi i filtrirati HTML sadržaj kako bi se otklonio eventualno zlonamjerno oblikovani programski kod. Drupal sadrži takve filtre, ali ovisi o korisniku sustava hoće li se oni koristiti. Na donjoj slici dan je primjer napada prilikom kojeg je na kraj URL zahtjeva za ranjivom stranicom „*vulnerable/show-me-the-data*“ dodan kod:

```
<IMG SRC=javascript:alert('XSS')>
```

koji vraća poruku upozorenja kako je prikazano na sljedećoj slici.

Slika 6. Upozorenje preglednika o XSS ranjivosti
Izvor: Cracking Drupal

4.1.2. CSRF

CSRF je vrsta napada u kojoj napadač izvodi akcije u ime korisnika. Dva osnovna tipa ovog napada se odnose na GET i POST akcije. GET akcija se odvija svaki put kad korisnik klikne na neku poveznicu ili pokrene URL zahtjev za stranicom. POST se odvija svaki put kada korisnik predaje neki obrazac stranici.

Drupal štiti od POST CSRF napada tako što se svakoj POST akciji dodjeljuje oznaka (eng. *token*) na temelju identifikatora sjednice i privatnog ključa stranice čime se otežavaju napadački upadi. Najčešće se ova vrsta napada izvodi prilikom dohvata podataka (HTTP GET zahtjevi).

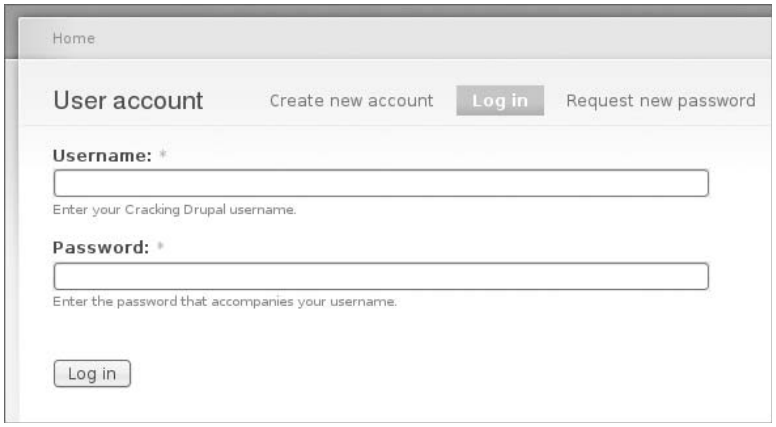
Primjer ovakvog napada je postavljanje posebno oblikovane poveznice na zloćudnoj stranici. Takva se poveznica može podmetnuti kao izvor neke slike. Ako se korisnik druge, ranjive stranice navede na pregled te slike, a pritom je u pregledniku zapamćena njegova prijava na ranjivoj stranici, automatski će se izvršiti podmetnuti zloćudan zahtjev. On može izgledati kao u sljedećem primjeru:

```
<img src="http://ranjiva.stranica/stetna_radnja?racun=admin">
```

4.1.3. Neovlašten pristup

Problem neovlaštenog pristupa podacima vezan je uz nesigurnu provjeru identiteta korisnika, odnosno uz ranjivosti u postupku autentikacije. One su najčešće posljedica korištenja slabih lozinki koje je moguće otkriti napadom pomoću rječnika (eng. *dictionary attack*) ili tzv. „*brute force*“ napadom. Riječ je o metodama koje iskušavaju različite lozinke, nasumice ili prema vjerojatnosti pojave (npr. neke tipične često korištene lozinke). Osim toga, lozinke se mogu otkriti ukoliko se šalju preko nezaštićenih veza, npr. autentikacija se obavlja koristeći obični HTTP protokol. U tom slučaju lozinke se šalju kroz Internet u otvorenom čitljivom obliku i bilo tko ih može otkriti prisluškivanjem prometa. Zato se preporuča korištenje sigurnih protokola koji uključuju enkripciju, kao što je HTTPS.

Izvorno Drupal koristi korisničko ime i lozinku za autentikaciju, ali takav oblik autentikacije može pružati nedovoljnu sigurnost ukoliko se radi o osjetljivijim radnjama kao što su financijske transakcije. U takve svrhe koriste se složeniji i sigurniji autentikacijski protokoli koji su dostupni u obliku dodatnih Drupal modula.



Slika 7. Obrazac za autentifikaciju
Izvor: Cracking Drupal

4.1.4. Podmetanje SQL koda

Ova vrsta napada vezana je uz nedovoljnu provjeru podataka koje unosi korisnik, a koriste se pri stvaranju i pokretanju SQL upita. Na taj način omogućuje se dohvat zaštićenih podataka, ali i podmetanje nepredviđenih naredbi. Npr., ako se u polju očekuje unos korisničkog imena koje se zatim umeće u sljedeću SQL naredbu:

```
statement = "SELECT * FROM korisnici WHERE ime = '" + korisnickoIme
+ "';".
```

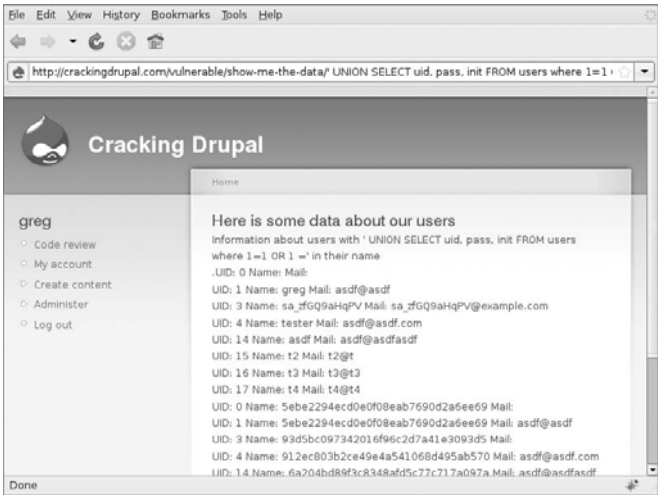
Podmetanjem sljedećeg sadržaja u podatkovno polje

```
a';DROP TABLE users; SELECT * FROM userinfo WHERE 't' = 't
```

dobiva se niz naredbi:

```
SELECT * FROM korisnici WHERE ime = 'a';
DROP TABLE korisnici;
SELECT * FROM InfoKorisnika WHERE 't' = 't ';
```

Prva naredba može ne vratiti niti jednog korisnika jer nitko nema ime „a“. Druga naredba će zatim pobrisati tablicu korisnici, a treća vratiti cijeli sadržaj tablice *InfoKorisnika* jer je izraz *'t' = 't'* uvijek istinit.



Slika 8. Primjer podmetanja SQL koda u URL zahtjev
Izvor: Cracking Drupal

4.1.5. Ostale prijetnje

Osim ranjivosti kod samog upravljanja *web* sadržajem treba uzeti u obzir cijeli kontekst u kojem Drupal sustav radi, a koji uključuje *web* poslužitelj, bazu podataka i naravno sam operacijski sustav. Već je spomenuto u poglavlju o administraciji sustava da valja ograničiti prava *web* poslužitelja isključivo na čitanje programskog koda Drupala. Napadi na *web* poslužitelje mogu se izvoditi slanjem velikog broja zahtjeva koji će dovesti do zagušenja i DoS (eng. *Denial of Service*) stanja. 2006. godine ispravljena je pogreška u Drupal sustavu koja je omogućivala izvođenje ove vrste napada dodavanjem velikog broja stranica na jedno *web* odredište te preopterećivanjem sustava. Općenito, u slučajevima ranjivosti koje omogućuju DoS napade rješenja se uvode izmjenom ranjivog koda.

Poznat pristup koji umanjuje opasnost od takvih napada je tzv. „*Least Privilege Principle*" koji kaže da se svakom korisniku (čovjeku ili procesu) trebaju dati samo one ovlasti koje su nužne za njegov rad. U slučaju Drupal sustava to se može primijeniti, primjerice, na slučaj kada jedan sustav podržava više *web* odredišta. Pritom je prilikom održavanja lakše koristiti jedan administratorski račun za sva sjedišta, ali to znači i da će ranjivost jednog sjedišta koja omogućuje stjecanje neovlaštenog administratorskog pristupa ugroziti sva ostala sjedišta. Zato se preporuča koristiti ovaj princip koji minimizira štetu od eventualnih napada.

Osim prijetnji koje ugrožavaju programski sustav napadi se uvijek mogu izvoditi i preko socijalnog inženjeringa koji korisnike s nedovoljno iskustava navodi na štetne radnje i otkrivanje osjetljivih podataka.

4.2. Zaštita sustava

Kod zaštite Drupal sustava prva i osnovna aktivnost o kojoj svaki administrator treba voditi računa je redovita nadogradnja. O dostupnosti novih programskih inačica paketa i modula administratora redovito obavještava modul *„Update status",* koji je dio jezgre u inačicama 6.x i novijim, a za inačicu 5.x dostupan je kao dodatan modul.

Kod odabira dodatnih modula za Drupal preporučljivo je dati prednost popularnijim i češće korištenim paketima. Naime, veća je vjerojatnost da će se otkriti ranjivosti šire korištenih paketa, a popularnost ukazuje i na kvalitetu usluge. Dobar pokazatelj sigurnosti modula je i činjenica da je u prošlosti bio nadograđivan sigurnosnim ispravcima jer to znači da se njegovo stanje prati. Osim toga, preporuča se koristiti i dodatne sigurnosne module od kojih su neki spomenuti već u poglavlju u dodatnim Drupal modulima.

4.2.1. Konfiguracija Drupala

Kod konfiguracije jezgre Drupal sustava preporuča se poduzeti sljedeće korake:

- Pažljivo odrediti korisničke uloge i prava pojedinih uloga – budući da ova podjela definira ovlasti nad sustavom korisnicima kojima su uloge podijeljene, propusti će otvoriti prostor za zlouporabe.
- Posebno korisna mogućnost Drupala je da korisnicima ograniči vrste HTML sadržaja koje mogu unositi. Tako se anonimnim korisnicima može primjerice omogućiti samo unošenje bezopasnih sadržaja u oznakama <*blockquote*> ili <*em*>, a ovlaštenim korisnicima mogu se omogućiti ugnježđivanje sadržaja u <*embed*> blokovima. Posebno osjetljivi HTML blokovi su: *SCRIPT, IMG, IFRAME, EMBED, OBJECT, INPUT, LINK, STYLE, META, FRAMESET, DIV, BASE, TABLE, TR i TD*. Njihovo korištenje zato treba omogućiti samo naprednijim i pouzdanim korisnicima.
- Također, Drupal sustav omogućuje pokretanje PHP koda korisnicima preko PHP filtra. U Drupal inačicama 5.x ova se mogućnost mora otkloniti nadogradnjom posebnim modulom Paranoia, dok se u inačicama 6.x filtar jednostavno može isključiti. Moguće je i ograničiti korištenje filtra samo na korisnike s višim ovlastima, ali u tom slučaju uvijek ostaje prostor za zlouporabe. Administrator sustava treba procijeniti koje je rješenje u slučaju njegovog *web* sjedišta najbolje.

4.2.2. Korisničke ovlasti

Korisničke ovlasti mogu se stvarati i dodavati preko Drupal aplikacijskog programskog sučelja (API). Za to se mogu definirati funkcije *hook_perm()* koje stvaraju nove ovlasti. Funkcijom *user_access('ovlasti')* mogu se provjeriti ovlasti trenutnog korisnika. Kod definiranja ovlasti za

pristupanje izbornicima koriste se ključevi *„access arguments“* i *„access callback“*. Njima se definira funkcija koja se koristi za provjeru ovlasti i argumenti koji joj se prosljeđuju. Ako se ne definira *callback* funkcija, koristi se funkcija *user_access()*. Pritom je za inačice Drupala starije od 6.2 vrijedilo da se ti ključevi nasljeđuju od roditeljskog izbornika dok se ti ključevi u novijim inačicama moraju definirati za svaki izbornički element posebno. Vrijednosti ključeva se nasljeđuju jedino u slučaju tipa *MENU_DEFAULT_LOCAL_TASK*.

```
<?php
  $items['user'] = array(
    'title' => 'My account',
    'page callback' => 'user_view',
    'page arguments' => array(1),
    'access callback' => 'user_view_access',
    'access arguments' => array(1),
  );
  $items['user/%user'] = array(
    'title' => 'View',
    'access callback' => 'user_view_access',
    'access arguments' => array(1),
  );
  $items['user/%user/view'] = array(
    'title' => 'View',
    'type' => MENU_DEFAULT_LOCAL_TASK,
  );
?>
```

Iz gornjeg primjera koda vidljivo je kako je u slučaju izbornika *„user/%user“* (ugnježđeni izbornik) bilo potrebno ponovno definirati ključeve jer nisu naslijeđeni od izbornika *user*. Za ugnježđeni izbornik *„/%user/view“* to pak nije bilo potrebno zbog definiranog tipa.

Kod stvaranja novih ovlasti važno je odrediti optimalnu zrnatost – dodjeljivati puno odvojenih ovlasti za zasebne korisničke uloge ili proglasiti sve korisnike s kritičnim ulogama administratorima? Drugi način daje previše nepotrebnih odgovornosti korisniku. Prvi pak može dovesti do pretjeranog, nepotrebnog razdvajanja ovlasti i uloga.

U slučaju pokušaja zabranjenog pristupa nekom dijelu sustava potrebno je poslati određenu poruku. Kako bi se smanjila vjerojatnost otkrivanja osjetljivih informacija programiranjem vlastitog koda za slanje te poruke, dostupna je Drupal API funkcija *drupal_access_denied()* čije se korištenje preporuča kad god je to moguće.

Osim toga u Drupalu postoji mogućnost da jedan korisnik na neko vrijeme preuzme ovlasti drugog korisnika. Tu se otvara prostor za zlouporabe. Takav slučaj je dodjela povišenih ovlasti korisniku tijekom izvođenja nekog procesa u kojem se dogodi pogreška koja izazove izlaz iz programa. Budući da se korisniku nisu uspjele vratiti stare ovlasti on dalje nastavlja rad s povišenim ovlastima. Kako bi se ovaj slučaj spriječio moguće je koristiti funkciju *session_save_session()* koja će argumentom *TRUE* ili *FALSE* odrediti čuvaju li se promjene koje proces izvodi. Primjer primjene ove funkcije je u donjem kodu (*uid* 1 je u Drupalu identifikator administratora).

```
global $user;
$current_user = $user;
session_save_session(FALSE);
$user = user_load(array('uid' => 1));
action_as_another_user();
$user = $current_user;
session_save_session(TRUE);
```

4.2.3. Provjera ulaznih i izlaznih podataka

Drupal API pruža više funkcija za filtriranje ulaznih i izlaznih podataka. Primjeri takve funkcije je *db_query()*. Ona omogućuje da se korisnički unos integrira u upit bazi podataka. Pritom je važno koristiti provjeru ulaznih podataka. Npr., navedena su dva načina izvođenja upita:

```
SELECT mail FROM users WHERE uid = 1;
```

```
db_query(“SELECT mail FROM {users} WHERE uid = %d“, $uid);
db_query('SELECT mail FROM {users} WHERE uid = '.$uid);
```

Prvi način formatiranja (*%d)* govori funkciji da *$uid* varijablu provjeri i koristi kao cijeli broj. Drugi primjer samo nadovezuje tu varijablu u upit. U drugom slučaju podmetanje proizvoljnog SQL koda u upit može dovesti do njegovog izvođenja, dok će u prvom slučaju pokušaj biti neuspješan jer će se neočekivani formati odbaciti. Ovo je jednostavan primjer, ali korištenjem različitih načina formatiranja podataka prilikom prijenosa u *db_query()* funkciju može se spriječiti veći broj napada podmetanjem SQL koda.

Slična funkcija za formatiranje i provjeru izlaznih podataka je *t()*.

```
$output = 'Info korisnka '. $user_search .' in their name<br>.’;
$output = t(’Info korisnika '. $user_search .' in their name<br>.’);
$output = t(’Info korisnika @search in their name<br>.’,
array(’@search’ => $user_search));
```

U prvoj liniji koda prikazana je poruka koja se treba ispisati korisniku. Pritom se varijabla *$user_search* samo umeće u poruku i može sadržavati bilo što. U drugoj liniji je poziv funkcije *t()* kojoj se predaje isti znakovni niz kao u prethodnoj. U ovom slučaju funkciji se ne predaju nikakve informacije o formatiranju niza pa neće biti nikakve koristi od provjere. U trećem slučaju, koristi se formatiranje varijable kojim se zahtjeva da ona bude tipa *array* (polje).

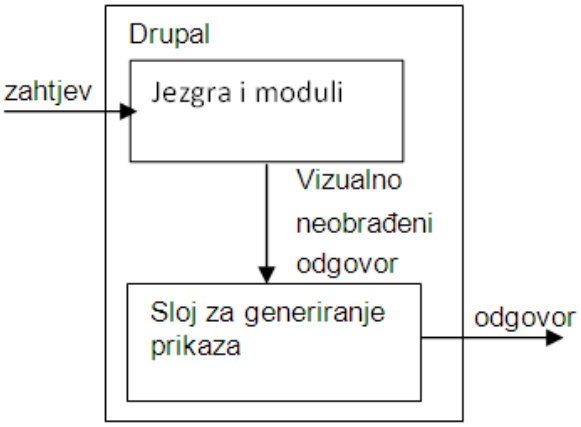
Drupal pruža i čitav niz drugih funkcija za filtriranje sadržaja čiji je kratak pregled dan u donjoj tablici.

Funkcija	Što radi	Kada se preporuča
check_plain	Provjerava je li dani sadržaj običan tekst bez HTML koda	Kako bi s izbjegli XSS napadi kod prosljeđivanja HTML korisničkog unosa
check_markup	Provjerava HTML sadržaj prema postavkama HTML filtra za trenutnog korisnika	Kad se mogu dopustiti određeni, ali ne svi HTML sadržaji
filter_xss_admin	Provjerava unos administratora na XSS napad	Kad se provjerava unos administratora koji smije sadržavati HTML
l	Pridružuje poveznice na *web* sadržaje pritom provjeravajući sigurnost zadanih URL-ova	Kod uporabe dinamičkih poveznica i premještanja stranica na druge domene ili poslužitelje

Slika 9. Drupal funkcije za filtriranje sadržaja

4.2.4. Sigurnost predložaka i tema

Sloj za stvaranje vizualnog prikaza stranice odvojen je od same Drupal jezgre i modula. Zahtjev za stranicom prvo se obrađuje u jezgri i dodatnim modulima Drupala. Pritom se stvaraju podaci za prikaz rezultata. Ti se podaci prosljeđuju sloju za prikaz stranice. Taj sloj stvara konačni kod za prikaz stranice koji se šalje *web* pregledniku, a može uključivati HTML, CSS, JavaScript, slike i druge elemente. Kako bi se izbjeglo prosljeđivanje grešaka u obradi, na ovaj sloj preporuča se u potpunosti obraditi podatke prije prosljeđivanja i tako pojednostaviti zadaću ovog sloja. Cilj je otkloniti sve probleme prilikom obrade zahtjeva tako da posao ovog sloja ostaje praktički samo dizajn stranice. To ima smisla jer vizualni identitet stranice često dizajniraju i programiraju korisnici koji nisu programeri.



Slika 10. Shema Drupal sustava

4.2.5. Osiguravanje pristupa sustavu

Jezgra Drupal sustava sadrži jednostavne mehanizme za provjeru prava pristupa *web* sadržajima. Neki od njih spomenuti su u dijelu dokumenta koji govori o korisničkim ovlastima. Osim toga spomenuti su moduli koji sadrže napredne mehanizme autentikacije korisnika. Ukoliko uz sve dostupne metode programer želi izgraditi vlastiti sustav za autentikaciju korisnika, može to učiniti kroz *Node Access* API. Ipak valja imati na umu da je u većini slučajeva bolja odluka osloniti se na gotova provjerena rješenja nego razvijati svoja. Uz to, važno je procijeniti postoji li stvarna potreba za uvođenjem novih mehanizama ili se željeni učinak može postići korištenjem već dostupnih mehanizama.

4.3. Automatsko ispitivanje sigurnosti

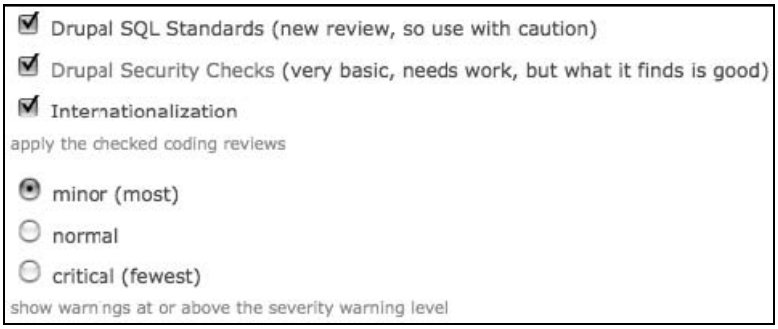
U ovom dijelu poglavlja ukratko su predstavljena tri Drupal modula koji se mogu koristiti za ispitivanje Drupal koda na ranjivosti. Riječ je o modulima:

- Coder modul,
- Security Scanner i
- Grendel Scan.

Od ovih paketa ne treba se očekivati revolucionarnost. Oni mogu otkriti tipične i česte propuste, no napadač uvijek može iskoristiti kod na specifičan i dovoljno jedinstven način da zaobiđe ovakve automatizirane provjere.

4.3.1. Coder modul

Coder modul je prilično moćan alat za analizu Drupal koda. Osim izvornog skupa testova koje sadrži, moguće ga je proširivati dodavanjem novih testova. Također, drugi moduli mogu proširivati taj skup testovima vlastite sigurnosti.

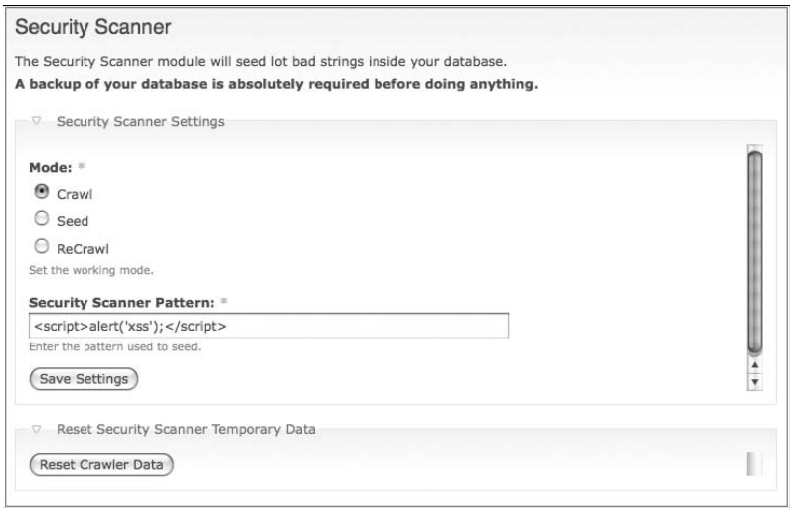


Slika 11. Coder modul postavke
Izvor: Cracking Drupal

Ispitivanja ovog modula pokazala su da uspješno uočava manje i relativno česte propuste (npr. izostanak provjere izlaznih podataka funkcijom *t()*). Ipak niti približno ne uspijeva uočiti sve propuste. Osobito je neosjetljiv na logičke, semantičke te složene XSS ili SQL propuste.

4.3.2. Security Scanner

Security Scanner je modul razvijen 2008. godine koji zasad traži samo XSS ranjivosti. Ovaj alat prilikom ispitivanja stvara mnogo beskorisnih podataka u bazi pa se njegovo izvođenje preporuča na ispitnoj kopiji stranice.



Slika 12. Konfiguracija Security Scanner alata
Izvor: Cracking Drupal

Po završetku skeniranja potrebno je osvježiti stranicu. Tada će se na njoj prikazati poruka upozorenja koja može izgledati slično kao primjer na donjoj slici.

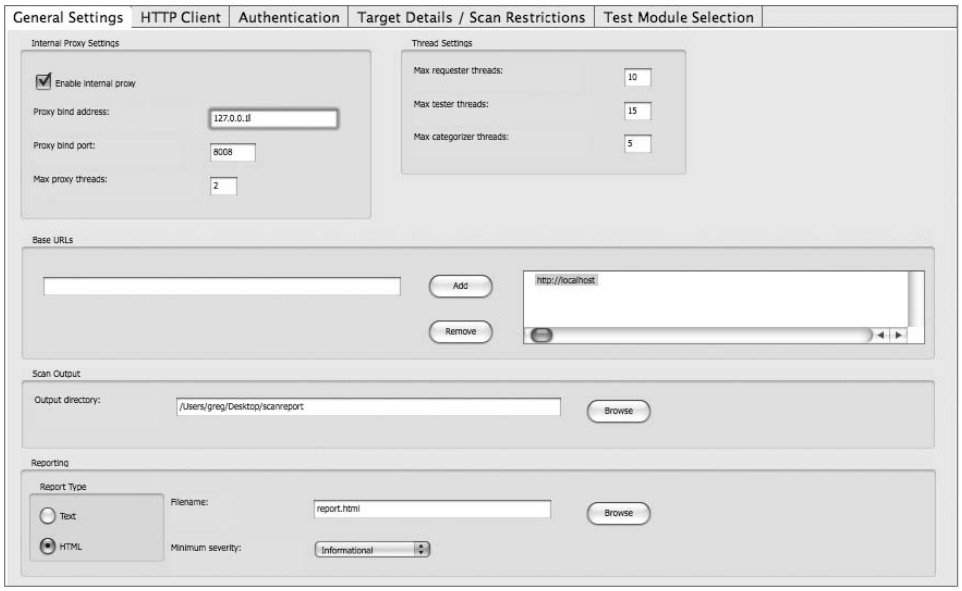
Slika 13. Primjer poruke upozorenja nakon skeniranja
Izvor: Cracking Drupal

4.3.3. Grendel-Scan

Grendel-Scan je noviji alat za ispitivanje Drupal koda koji u usporedbi s dva prethodno spomenuta alata ima znatno veće mogućnosti provjere ranjivosti. To ujedno znači da testovi mogu potrajati duži vremenski period. Grendel koristi dvije metode za pronalaženje kritičnih dijelova stranice:

- Pronalazi URL-ove u HTML-u i *JavaScriptu*. Potom izvodi veći niz zahtjeva kako bi pronašao ranjivosti,
- Grendel se konfigurira kao posredni poslužitelj *web* preglednika, a potom se pamte i skeniraju stranice koje se posjećuju preko preglednika.

Grendel pronalazi SQL, XSS ranjivosti, HTML oznake, ali kao i preostala dva modula ne uspijeva pronaći specifične ranjivosti.



Slika 14. Korisničko sučelje alata Grendel-Scan
Izvor: Cracking Drupal

Teško je očekivati od automatiziranog alata da će uspjeti prepoznati sva kritična područja koja napadač može iskoristiti ili koja programer može previdjeti. Jednako kako ih previdi prilikom stvaranja koda, tako ih može previdjeti prilikom stvaranja automatiziranog ispitivača koda. Ipak, velik broj zlouporaba događa se na poznatim i jednostavnim ranjivostima koje se nekim od ovih alata lako mogu ukloniti.

5. Zaključak

U ovom dokumentu predstavljen je CMS sustav Drupal. Navedene su njegove ranjivosti te mogućnosti njihovog otklanjanja i sprečavanja. Detaljnijom analizom sigurnosnih aspekata može se zaključiti kako Drupal posjeduje dovoljno mehanizama zaštite, što u obliku programskih sučelja, što u obliku gotovih paketa, za primjerenu zaštitu CMS sustava i korisničkih podataka. Ipak, valja imati na umu kako potpuna zaštita programskih sustava nije moguća zbog raznovrsnosti napada i ranjivosti koje se mogu potkrasti u bilo kojem kodu. Ono što ovaj sustav čini relativno pouzdanim jest činjenica da se on redovito prati i nadograđuje.

Svakako najvažnija zadaća korisnika ovakvog sustava je informiranje o njegovim mogućnostima. Potom valja odabrati rješenja i metode koje su najprikladnije za konkretnu svrhu i konkretnog korisnika. Primjerice, ukoliko se radi o korisniku koji nema programerskih vještina, svakako je preporučljivo koristiti gotove alate i module umjesto da se razvijaju vlastiti. S druge strane kod specifičnih potreba, uz dovoljno znanja bolje rješenje možda će biti upravo rad s programskim sučeljima. Konačno, važno je znati u što se korisnik upušta, kakvi su rizici, ali i kakve su mogućnosti nošenja s tim rizicima. Ovaj dokument je tek okviran pregled tih rizika i mogućnosti. Ozbiljnije upoznavanje s Drupalom zahtjeva korištenje službene dokumentacije i dugotrajniji rad sa samim programom.

6. Reference

[1] Drupal Security, http://drupal.org/project/security, veljača 2010.
[2] Greg James Knaddison: Cracking Drupal A Drop in the Bucket
[3] Drupal Security team, http://drupal.org/security-team, veljača 2010.
[4] Wikipedia Drupal, http://en.wikipedia.org/wiki/Drupal, veljača 2010.