UNIVERSITATEA "TITU MAIORESCU" DIN BUCUREȘTI

FACULTATEA DE INFORMATICĂ

LUCRARE DE DISERTAȚIE

COORDONATOR ȘTIINȚIFIC:

Conf.univ.dr. Iustin Priescu

ABSOLVENT:

Olivian-Claudiu Breda

SESIUNEA IUNIE

2020

UNIVERSITATEA "TITU MAIORESCU" DIN BUCUREȘTI

FACULTATEA DE INFORMATICĂ

LUCRARE DE DISERTAȚIE

Securitatea în WordPress

COORDONATOR ȘTIINȚIFIC:

Conf.univ.dr. Iustin Priescu

ABSOLVENT:

Olivian-Claudiu Breda

SESIUNEA IUNIE

2020

Rezumat / Abstract

Rezumat în limba română: Lucrarea analizează securitatea unor site-uri standard construite pe platforma WordPress și compară evoluția platformei între două versiuni majore (5.0 și 5.4). O primă parte principală se referă la testarea la nivel de server (serverul https://olivian.ro). Urmează o parte în care se face o testare la nivel de instanță WordPress, se testează cu diferite soluții de securitate două instalări de WordPress, WordPress 5.0, https://olivian.ro/wp50/, și WordPress 5.4, https://olivian.ro/wp54/. Lucrarea se încheie cu soluții pentru o parte din problemele observate.

Domeniu principal: Securitate aplicații web.

Domenii adiacente: Programare, găzduire, managementul aplicațiilor web.

Tipul lucrării: cercetare.

English language abstract: The paper analyzes the security of standard-built web sites on the WordPress platform and compares the evolution of the platform between two major versions (5.0 and 5.4). The first main part refers to testing at the server level (https://olivian.ro server). Next, a WordPress instance-level test is performed: two WordPress installations are tested with different security solutions - WordPress 5.0, https://olivian.ro/wp50/, and WordPress 5.4, https://olivian.ro/wp54/. The paper finishes with solutions to some of the noticed problems.

Main field: Web application security.

Other fields: Programming, hosting, web application management

Type of work: research.

Cuprins

Lista figurilor

Lista tabelelor

Introducere

În lucrarea de față am avut ca obiectiv o comparație între securitatea platformei de gestionare a conținutului WordPress pentru două versiuni de WordPress distincte - 5.0 și 5.4. Am urmărit în paralel să văd cum a evoluat securitatea versiunii de WordPress între două versiuni destul de importante, și, la data începerii cercetării (aprilie 2020), recente - WordPress 5.0 a fost lansat în 6 decembrie 2018, iar în 31 martie 2020 a apărut WordPress 5.4. Am testat inițial serverul de pe hosting-ul site-ului propriu, olivian.ro (testare la nivel de server), ulterior am testat două instalări de WordPress distincte - WordPress 5.0, aici: https://olivian.ro/wp50/, și WordPress 5.4, aici: https://olivian.ro/wp54/ (testare la nivel de instanță). La final, am căutat soluții pentru problemele găsite.

De ce am ales tema - securitate pe WordPress? Experiența personală cu WordPress a început cu un site pe WordPress.com (e diferit față de WordPress.org) al cărui prim articol a fost în 19 noiembrie 2007 și, ulterior, a ajuns la vreo 40 de articole. Am continuat cu un blog pe domeniul propriu, bazat pe WordPress-ul pe care îl vom analiza în lucrarea de față, WordPress.org, prin martie 2008. Am început, pe măsură ce foloseam WordPress.org, să folosim platforma nu atât pentru a crea bloguri, cât site-uri de prezentare pentru diferite entități (ONG-uri, școli, clienți în colaborare cu un PFA - Persoană Fizică Autorizată). Am înregistrat ulterior mai multe domenii web, împărțite între tipuri de activități și diferite de limbi în care erau scrise articolele, urmând ca în 9 aprilie 2012 să înregistrăm olivian.ro și să unim toate vechile platforme pe una singură, pe care lucrăm și astăzi.

În toamna anului 2018 am început o colaborare ca dezvoltator pe platforma WordPress pentru o agenție de web design și development din București, activitate pe care o desfășurăm și la data scrierii acestei lucrări (luna mai 2020). Am lucrat mult mai intens, astfel, pe platforma WordPress.

Am lucrat la lansarea a mai mult de 100 de site-uri pe platforma WordPress, cu intensitate a implicării diferită (instalarea unei teme/template pentru un site deja lansat, reinstalarea unei alte teme pentru un site deja lansat, site realizat de la zero - scris cod, contribuție semnificativă la realizare site). De asemenea, am gestionat de-a lungul timpului (mentenanță WordPress - actualizări site-uri) câteva zeci de site-uri. De asemenea, am folosit din perspectiva de blogger, autor de articole, platforma WordPress pentru a scrie câteva mii de articole pe diferite platforme, avem așadar și experiență pe partea de folosire a platformei.

Pe partea de programare / cunoștințe Internet, în afara programului de masterat curent (Universitatea "Titu Maiorescu" din București - Masterat Securitatea sistemelor informatice și a rețelelor informaționale), am absolvit un liceu profil informatic, am urmat cursurile unui masterat în IT (Modelare și Tehnologii Informatice - Universitatea „Ovidius” din Constanța), și am lucrat aproape continuu din martie 2006 la diferite locuri de muncă care presupuneau folosirea intensă a Internetului (în principal - SEO, Search Engine Optimization - optimizare pentru motoarele de căutare, dar și uzabilitate, UX - User Experience).

Ca notă de redactare a lucrării, pentru un paragraf în care am citat exact, sau aproximativ, o resursă, am pus la final de paragraf sursa. Dar dacă după un paragraf urma o enumerare, am pus sursa doar la paragraful de sus, chiar la finalul lui, lăsând să se înțeleagă că enumerarea face parte din paragraful citat și conține referința. Am evitat astfel să punem sursa la fiecare item al unei enumerări, nu ar fi fost plăcut vizual ca dacă e o enumerare cu 5 elemente cu aceeași sursă, să menționăm sursa la fiecare element din cele 5.

Capitolul 1. Aspecte teoretice

1.1. Motivația demersului

Conform "Ghidului Pentru Securizarea Aplicațiilor Și Serviciilor Web. Versiunea 1.0 – 24 Februarie 2012." (Centrul Național de Răspuns la Incidente de Securitate Cibernetică), "o etapă esențială pentru orice dezvoltator de aplicații web este identificarea acestor vulnerabilități și tratarea acestora corespunzător". Am pornit de la această observație în încercarea de a găsi cele mai importante aspecte ale securității unei instanțe de WordPress. Astfel, am dorit să testăm cât de sigură este o implementare de WordPress, în varianta standard. (Ghid pentru securizarea aplicațiilor și serviciilor Web. Versiunea 1.0 – 24 februarie 2012, 2012)

Tema lucrării de față este foarte generoasă. "Securitatea în WordPress" este un subiect dezbătut intens în numeroase lucrări. Lista bibliografică a lucrării curente are o parte din resursele identificate, însă cu siguranță lista este mult mai bogată. Ce am încercat să facem în această lucrare a fost să căutăm să venim cu analize recente, unele din ele pentru versiuni de WordPress recente (5.0 și 5.4), și să analizăm o instalare fără nicio setare suplimentară făcută. Am analizat WordPress atât la nivel de server (hosting), cât și la nivel de instanță WordPress cu setări standard. Am căutat, de asemenea, să găsim soluții pentru problemele întâmpinate. În acest sens, prin complexitate, dată recentă a analizei și combinație între partea practică a problemelor și cea a soluțiilor, demersul are caracteristici unice, la momentul redactării.

1.2. Limitări ale demersului

Există numeroase aspecte în afara scopului lucrării de față. Spre exemplu, în ceea ce privește domeniul de aplicare a Regulamentului (UE) 2016/679, acesta "se aplică prelucrării datelor cu caracter personal, efectuată total sau parțial prin mijloace automatizate, precum și prelucrării prin alte mijloace decât cele automatizate a datelor cu caracter personal care fac parte dintr-un sistem de evidență a datelor sau care sunt destinate să facă parte dintr-un sistem de evidență a datelor." (Ghid întrebări și răspunsuri cu privire la aplicarea regulamentului (UE) 2016/679, fără dată) În acest sens, există numeroase aspecte care țin de prelucrarea datelor cu caracter personal, așadar de securitatea unui site, care nu au fost analizate în lucrarea de față. Analiza s-a restrâns exclusiv asupra securității WordPress unei instalări standard, fără particularizări ale acesteia (template-uri / teme / grafică), fără materiale adiționale (pluginuri / module), fără nicio pagină / articol / comentariu adăugate în site. Desigur, acest lucru lasă mult loc de testare

în alte posibile lucrări, iar ca aspect pozitiv al acestui tip de demers a fost faptul că, prin axarea pe niște aspecte specifice, am putut testa în amănunt, am putut aprofunda bine tema aleasă.

În altă ordine de idei, probabil cea mai mare limitare a testelor efectuate pe server și pe cele două instanțe de WordPress a fost aceea că nu pot înlocui, cel puțin în momentul de față, experiența unei persoane care analizează un site. Testele făcute au avut această limitare. Am încercat să găsim o rezolvare acestei probleme prin furnizarea, în capitolul dedicat soluțiilor concrete, a unor pași manuali ce pot fi făcuți pentru a îmbunătăți securitatea WordPress.

1.3. De ce am ales WordPress ca platformă pentru a fi testată?

Inițial, WordPress a fost inițial o ramură a unui software mai vechi numit b2/cafelog. WordPress a fost dezvoltat de Matt Mullenweg și Mike Little. În prezent este întreținut de o echipă de dezvoltatori (inclusiv Matt Mullenweg). (Król și Silver, 2013)

WordPress a apărut cu succes ca o platformă de blog și s-a extins la un sistem complet de gestionare a conținutului (CMS), care include instrumente și funcțiile pentru a publica un întreg site web, fără a necesita o mulțime de expertiză tehnică sau înțelegere. (Sabin-Wilson, fără dată)

Open source (sursa deschisă) este piatra de temelie a WordPress și a comunității WordPress. Mai exact, licența publică generală (General Public License - GPL) ghidează principiile care se referă la dezvoltarea WordPress. Faptul că WordPress are la bază cod deschis a facilitat dezvoltarea lui. (Brazell, 2010)

Notă - există două tipuri de WordPress (Smith și McCallister, 2010):

- Cea mai simplă versiune de utilizat este versiunea gratuită a software-ului WordPress pentru bloguri, găzduită gratuit de Automattic, disponibilă pe WordPress.com.
- Versiunea inițială, și încă mai populară, a WordPress este software-ul gratuit care se descărcă de pe site-ul WordPress.org. Acesta presupune găzduire pe un server web și domeniu propriu (pe WordPress.com există posibilitatea de a avea site.wordpress.com).

WordPress este folosit de 63,3% din toate site-urile web al căror sistem de administrare a conținutului (în engleză: CMS, *Content Management System*) poate fi determinat prin metode automate. Acest procent este 36,1% din totalul site-urilor web. (Usage Statistics and Market Share of WordPress, May 2020, 2020) Cu alte cuvinte, mai mult de unul din 3 site-uri de pe Internet folosesc WordPress drept CMS.

Motive pentru care este ales WordPress: ușor de folosit, flexibil, personalizabil, gratuit și open source (licență liber de folosit, în anumite termene). (Valk et al., fără dată)

În ceea ce privește varianta de WordPress folosită, conform unor statistici din 12 mai 2020, 76,5% din site-uri folosesc WordPress versiunea 5 (cu toate subramurile ei, la data de 12 mai 2020 cea mai recentă versiune este 5.4.1), 21,9% din site-uri folosesc ramuri ale versiunii 4, conform tabelului de mai jos:

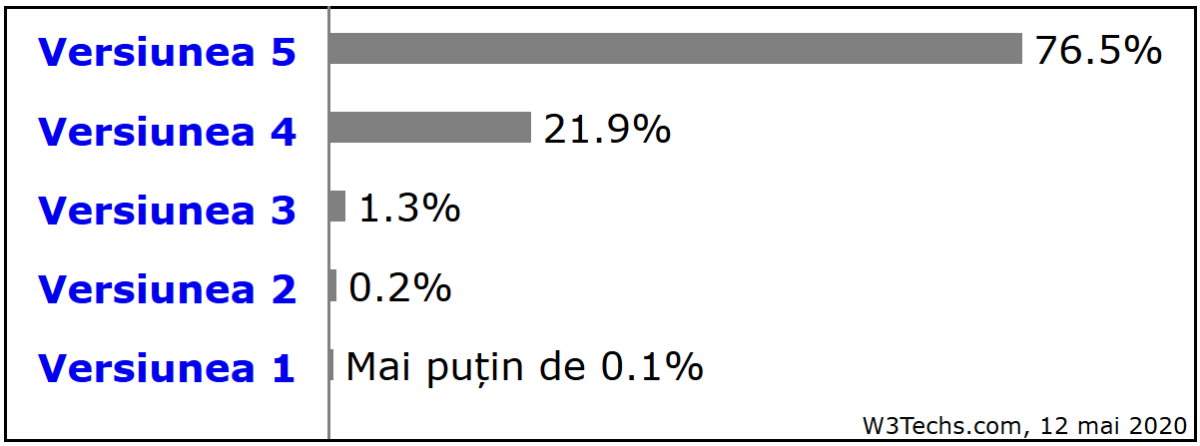


Figura 1.3.1. Procente de site-uri web care utilizează diferite versiuni de WordPress

Așadar, aproximativ 3 din 4 site-uri folosesc WordPress varianta 5 și sub-ramuri ale sale.

Am ales să testăm versiunea 5.0 (prima versiune oficială, non-beta / non-testare, din ramura 5 a WordPress), în comparație cu versiunea 5.4 (una din ultimele versiuni, chiar ultima la momentul începerii acestei cercetări), pe baza unei cercetări anterioare pe site-ul CVE®. (Wordpress: CVE security vulnerabilities, versions and detailed reports, fără dată)

Notă - sunt două tipuri de versiuni WordPress - minore și majore. Pentru înțelege diferența dintre cele două, trebuie analizat numărul versiunii WordPress. Lansările majore schimbă una dintre primele două cifre din numărul versiunii (de exemplu, 4.0.8 la 4.1.0 este o actualizare majoră). Comunicările minore modifică numărul versiunii minore, care este cifra după al doilea punct zecimal (de exemplu, 4.0.8 la 4.0.9). (MacDonald, 2014)

Conform site-ului The MITRE Corporation, pe site-ul CVE® apare o listă de înregistrări - fiecare din ele cu număr de identificare, descriere și referință publică - pentru vulnerabilități de securitate cibernetică cunoscute public. Datele din CVE sunt utilizate în numeroase produse și servicii de securitate cibernetică din întreaga lume, inclusiv în baza de date națională a vulnerabilităților din S.U.A., NVD, Baza de date națională a vulnerabilităților din S.U.A. (în

limba engleză: National Vulnerability Database). (CVE - Common Vulnerabilities and Exposures (CVE), fără dată)

Versiune WordPress	Număr rezultate (potențiale breșe, documentate public)	URL folosit pentru date (sursa datelor)
(orice versiune; probabil cele mai vechi versiuni sunt cele mai afectate)	294	https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/
2	39	https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-31373/opxss-1/Wordpress-Wordpress-2.0.html
3.1.3	48	https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-121203/Wordpress-Wordpress-3.1.3.html
4	33	https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-176072/Wordpress-Wordpress-4.0.html
5.0 RC2	10	https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-276684/Wordpress-Wordpress-5.0.html
5.2.2	7	https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-335030/Wordpress-Wordpress-5.2.2.html
5.4	0	Nelistată nicio problemă

Tabel 1.3.1 Număr vulnerabilități afișate public pe site-ul CVE® (Wordpress: CVE security vulnerabilities, versions and detailed reports, fără dată)

Am prezentat, mai jos, rezultatele și în format grafic, am inclus și o linie a tendințelor. Ca interpretare personală a datelor, este firesc ca în momentul de față să fie relativ puține breșe cunoscute pentru versiunea 5.4, dar probabil situația se va schimba în câteva luni, odată cu descoperirea a noi vulnerabilități. (Wordpress: CVE security vulnerabilities, versions and detailed reports, fără dată)

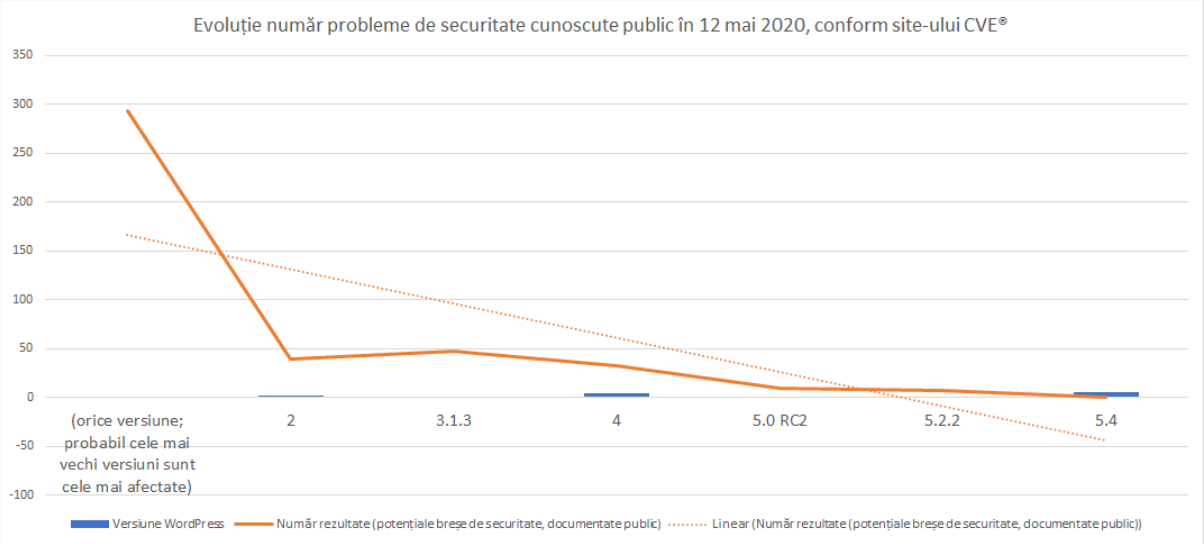


Figura 1.3.2. Evoluție număr probleme de securitate cunoscute public în 12 mai 2020, conform site-ului CVE® (Wordpress: CVE security vulnerabilities, versions and detailed reports, fără dată)

Tendințe ale vulnerabilităților în timp

An	# Vulnerabilități	DoS	Executare cod	Overflow	Corupere memorie	SQL Injection	XSS	Navigare directoare	Împărțire răspuns Http	Ocolire a unui lucru	Obținere informații	Obținere privilegii	CSRF	Includere de fișiere	# exploit-uri
2004	2						1		1						
2005	10		5			3	2				3				
2006	16	1	2			1	5	1			3				
2007	40	2	13			7	19			3	5		2		1
2008	27	2	4			3	9	4		1	2		2		
2009	12	3	1				3			1	3	1			2
2010	2		1			1									
2011	11					1	2				4				
2012	24	2	2			2	9			5	3		3		7
2013	19	1	1				8			3	2		1		
2014	29	3	3			1	8	1		6	2		3	1	
2015	11	1	2			1	7			1	1		1		
2016	20	1					9			6	1		1		
2017	43	1	1			4	14	4		5	2		5		
2018	17	1	4				5	1		3	1				
2019	11		2				7	1			1		1		
Total	294	18	41			24	108	12	1	34	33	1	19	1	10
% din toate		6.1	13.9	0.0	0.0	8.2	36.7	4.1	0.3	11.6	11.2	0.3	6.5	0.3	

Avertisment: Vulnerabilitățile cu date de publicare înainte de 1999 nu sunt incluse în acest tabel și în grafic. (Deoarece nu sunt foarte multe, și fac ca pagina să arate rău și este posibil să nu fie publicate în acei ani.)

Figura 1.3.3. Tendințe ale vulnerabilităților [WordPress] în timp, conform site-ului CVE® (Wordpress: CVE security vulnerabilities, versions and detailed reports, fără dată)

Hannes Trunde și Edgar Weippl (University of Applied Sciences Technikum Wien, Viena, Austria) au realizat o analiză a exploit-urilor (vulnerabilități) disponibile public. A fost utilizată funcția de căutare din Exploit-DB pentru a găsi toate exploit-urile referitoare la nucleul (core), plugin-urile și temele WordPress. În total, 488 de înregistrări de tip exploit între anii 2003 și 2014. 405 din cele 488 de înregistrări vizau plugin-uri. După prima lansare WordPress din

2003, majoritatea vulnerabilităților au fost găsite în nucleu / core. Această tendință s-a schimbat în 2008, așa cum este descris în figura de mai jos. (Trunde și Weippl, 2015)

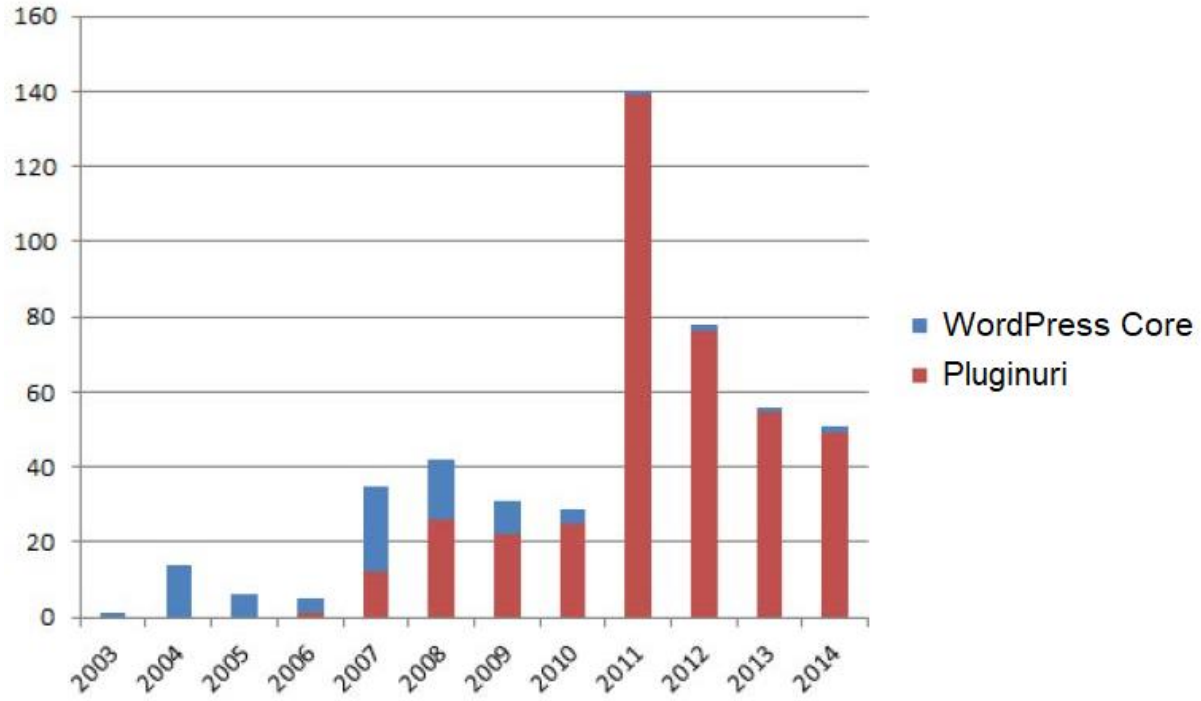


Figura 1.3.4. Distribuirea exploit-urilor pe 15 categorii de exploatări, WordPress, 2003-2014 (Trunde și Weippl, 2015)

Într-o comparație a celor mai populare sisteme de management al conținutului (CMS) - WordPress, Joomla!, Drupal, Magento -, se observă o tendință de creștere a infecțiilor pe WordPress. Din suma infecțiilor (cunoscute) ale celor 4 platforme, WordPress a înregistrat o creștere de la 90% la 94% în intervalul 2018-2019. Datele provin din "2019 Website Threat Research Report", raport al atacurilor din 2019. Pentru a înțelege o imagine de ansamblu asupra datelor strânse, peste 170 de milioane de încercări de atacuri au fost oprite folosind Sucuri Firewall. (Sucuri - Website Threat Report 2019, fără dată)

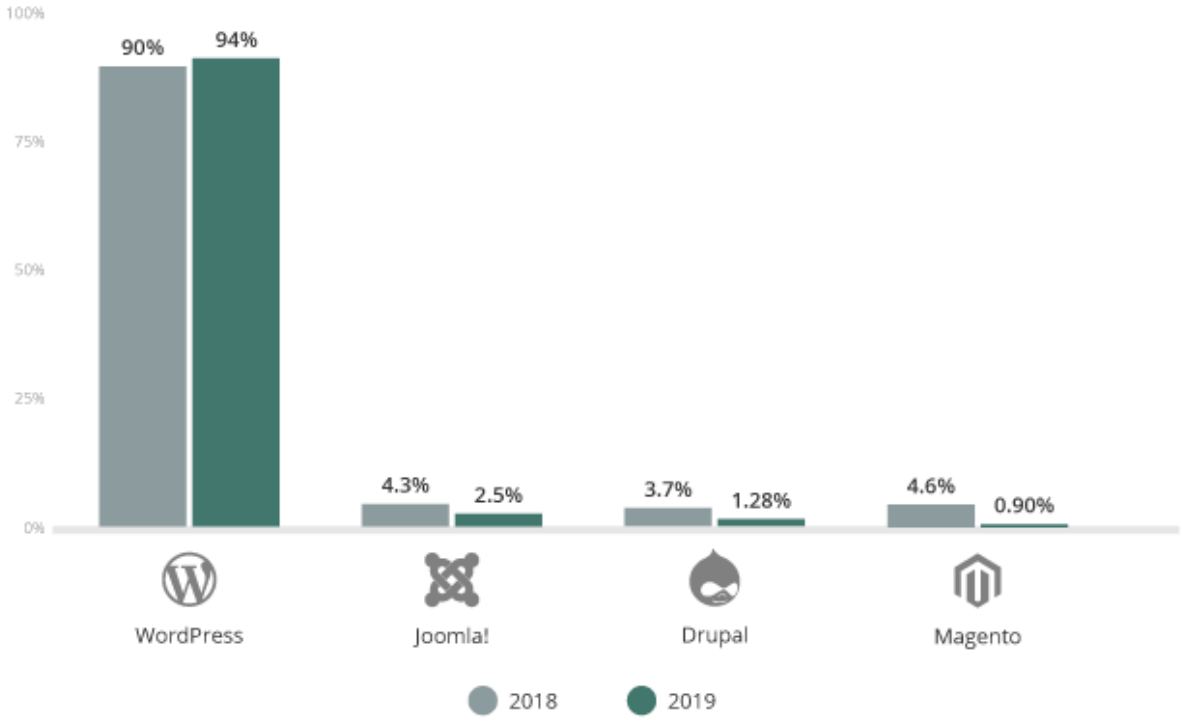


Figura 1.3.5. Comparație între sisteme de management al conținutului, 2018/2019 (Sucuri - Website Threat Report 2019, fără dată)

1.4. Cele mai mari riscuri de securitate pentru aplicații web, în general

Conform OWASP, cele mai mari 10 riscuri de securitate pentru aplicații web sunt: (OWASP Top Ten Web Application Security Risks | OWASP, fără dată)

1. SQL injection. SQL, Structured Query Language, este un limbaj de interogare structurat pentru sisteme de manipulare a bazelor de date relaționale.
2. Autentificare necorespunzătoare.
3. Expunerea datelor sensibile.
4. Entități externe XML (XXE). XML, Extensible Markup Language, este un limbaj de marcare pentru crearea de alte limbaje de marcare. XXE, XML external entity injection, este o vulnerabilitate de securitate web care permite unui atacator să interfereze cu prelucrarea unei aplicații de date XML.
5. Control de acces necorespunzător.
6. Configurare greșită de securitate.

7. XSS pentru scripturi de site-uri. XSS, Cross Site Scripting - introducere cod HTML sau JavaScript în pagini în mod automat. HTML, HyperText Markup Language, este un limbaj pentru creare pagini web.
8. Deserializare nesigură.
9. Utilizarea componentelor cu vulnerabilități cunoscute.
10. Logare și monitorizare insuficiente.

1.5. De ce ar proteja cineva o instanță de WordPress?

Dacă un utilizator are un site foarte nesigur, atunci script kiddies (hackeri care folosesc scripturi, bucăți de cod, standard de atac asupra unui site) vor avea accesul asupra unui site până la eliminarea accesului. Pe de altă parte, un site bine securizat este un motiv de descurajare pentru un hackerul obișnuit. De reținut că dacă un hacker este motivat să aibă acces asupra site-ului, și este bun în ceea ce face, va face mari demersuri în a găsi puncte slabe. Acest gen de hackeri sunt genul despre care nu se poate ști vreodată dacă au intrat pe site. Pot intra, obțin ceea ce au nevoie și nu lasă urme. (Canavan, 2011)

1.6. Creare mediu de testare - două instalări standard WordPress

Am instalat pe serverul https://olivian.ro două versiuni de WordPress (5.0 și 5.4) și le-am pus în două foldere (directare) ușor de reținut: https://olivian.ro/wp50/ și https://olivian.ro/wp54/.

Instalarea a fost cu setările implicite (limba română). Nu a fost niciun plugin suplimentar instalat, doar plugin-urile standard - "Akismet anti-spam" și "Hello, Dolly!", dar nici acestea nu au fost activate.

Tema de WordPress pentru 5.0 a fost "Twenty Nineteen, Versiunea: 1.0" (varianta implicită), iar pentru 5.4 "Twenty Twenty Versiunea: 1.2" (la fel, varianta implicită).

Am creat un singur utilizator, cu drepturi de administrator, pe ambele instalări.

Nu am făcut setări suplimentare în admin, de exemplu nu am modificat legăturile / URL-urile permanente (permalinks), nu am modificat data sistemului, alte asemenea.

Totuși, am dezactivat actualizările automate în WordPress adăugând această linie de cod în fișierul wp-config.php: (How to Disable Automatic Updates in WordPress, 2015)

```
define( 'WP_AUTO_UPDATE_CORE', false );
```

Acest lucru a dezactivat actualizările automate de WordPress. Am prevenit astfel eventuala actualizare de versiuni WordPress la alte versiuni minore, lucru care ar fi făcut dificilă testarea platformei WordPress în condiții similare de la un test la altul. (How to Disable Automatic Updates in WordPress, 2015)

Pentru baza de date, am lăsat prefixul standard sugerat de WordPress, "wp_", deși acest lucru nu este recomandat, din motive de securitate.

În cadrul instalării, am ales o combinație de nume utilizator și parolă complicate, de asemenea am ales utilizator, parolă și nume baze de date complicate, problema la WordPress este că aceste lucruri sunt opționale. Da, la alegere parolă utilizator, platforma WordPress sugerează o parolă complicată, și utilizatorul este avertizat că alege o parolă slabă dacă face asta, dar este, totuși, relativ nesigur acest lucru.

Capitolul 2. Testare WordPress la nivel de server

2.1. Introducere

Despre scanarea vulnerabilităților web, în cartea "WordPress 3 Ultimate Security" se menționează că în plus față de verificarea generală a "sănătății" unui site, scanările asupra serverului au un nivel variabil, sau chiar și niciunul, de evaluare a aplicațiilor web, cum ar fi WordPress. În capitolul de față vom analiza scanări asupra serverului, la modul general. (Connelly, 2011)

Vom pune în secțiunea Anexe imagini (capturi ecran) cu testele efectuate.

2.2. Testare observatory.mozilla.org

Ca probleme întâmpinate conform site-ului observatory.mozilla.org, se pot menționa: (Mozilla Observatory :: Scan Results for olivian.ro, 2020)

- Nu a fost implementat headerul Content Security Policy (CSP) - Politica de securitate a conținutului (CSP) poate preveni atacuri cross-site (XSS) și clickjacking (deturnare de clickuri).
- Nu a fost implementat headerul HTTP Strict Transport Security (HSTS) - acest lucru face în mod implicit ca un browser web să viziteze site-ul prin HTTPS. În general, noi folosim un plugin de WordPress dedicat care face acest lucru.
- Nu a fost implementat headerul X-Content-Type-Options - acestea sugerează browserelor să nu ghicească tipurile MIME de fișiere pe care le furnizează serverul web.
- Nu a fost implementat headerul X-Frame-Options (XFO) - aceasta controlează dacă site-ul poate fi inclus în frame-uri, protejând împotriva atacurilor de tip clickjacking. În prezent, există și directiva frame-ancestors (Content Security Policy), dar se recomandă să fie încă folosită.
- Nu a fost implementat headerul X-XSS-Protection - protejează împotriva atacurilor de scripturi cross-site (XSS) reflectate în IE (Internet Explorer) și Google Chrome, dar a fost înlocuită de politica de securitate a conținutului. Este utilă, încă, pentru a proteja utilizatorii de browsere web mai vechi.

Au fost și alte rezultate, dar în principal acestea sunt observațiile la care au fost lucruri de îmbunătățit.

Ca interpretare a acestor testări, considerăm că problemele acestea sunt suficient de importante pentru a le acorda atenție și rezolvare.

2.3. Testare securityheaders.com

Dintre problemele observate - Strict-Transport-Security; Content-Security-Policy; X-Frame-Options; X-Content-Type-Options; Referrer-Policy; Feature-Policy -, menționăm doar ultimele două aspecte, care nu au fost menționate în testul observatory.mozilla.org: (Helme, fără dată)

- Referrer Policy (politica pentru referreri, sursă) este un antet nou care permite unui site să controleze cât de multe informații include browser-ul când transmite informații în afara site-ului. Se consideră că ar trebui să fie setate de toate site-urile.
- Feature Policy (politica pentru caracteristici) este un antet nou care permite unui site să controleze ce caracteristici și API-uri pot fi utilizate în browser (API, Application Programming Interface, set de definiții programare).

Despre aceste rezultate putem afirma că sunt un risc de securitate relativ modern și nou, care ar ajuta să îl rezolvăm pe site-ul testat.

2.4. Testare ssltrust.com.au

Folosind SSLTrust (instrument gratuit de verificare a securității site-urilor web), au fost verificate mai mult de 60 de baze de date de la companii precum Google, Comodo, Opera, Securi ș.a. S-a testat împotriva malware, spam, am obținut inclusiv un raport de încredere a site-ului. Din cele 80 de teste realizate, au fost 0 pozitive și 8 pentru care testarea nu a avut date în baza de date. Așadar, un test cu rezultate pozitive (72/80 de teste) sau cel mult neutre (8/80). (Free Website Safety & Security Check, fără dată)

Ca interpretare a rezultatelor, considerăm că rezultatele sunt pozitive (cel mult neutre).

2.5. Testare ssllabs.com

Certification Authority Authorization - Autorizarea Autorității de Certificare (CAA), specificată în RFC 6844 în 2013, este o propunere de îmbunătățire a rezistenței ecosistemului

PKI cu un nou control pentru a restricționa care CA pot emite certificate pentru un anumit nume de domeniu. Deși CAA a fost în starea standard propusă de mai bine de 4 ani, doar o sută sau două sute de site-uri au adoptat-o. Forumul CA / Browser (CA / Browser Forum) sprijinul CAA ca parte a cerințelor de bază standard de eliberare a certificatelor. CAA creează un mecanism DNS care permite proprietarilor de nume de domeniu să caute lista albă a cărora li se permite să emită certificate pentru numele lor de găzduire (hosting). (Ristic, 2020) Exemplu folosire:

```
example.org. CAA 128 issue "letsencrypt.org". (Ristic, 2020)
```

Mai jos, SSL (Secure Sockets Layer) este un protocol criptografic pentru comunicații sigure pe Internet, la fel ca TLS (Transport Layer Security), IE (Internet Explorer) este un browser web realizat de compania Microsoft Corporation, Win (Microsoft Windows) este un sistem de operare, la fel ca iOS (pentru telefoane inteligente, smartphone) sau OS X (desktop).

TLS a evoluat dintr-un protocol de criptare anterior, SSL, care a fost dezvoltat de Netscape. Versiunea TLS 1.0 a început de fapt dezvoltarea ca SSL versiunea 3.1, dar numele protocolului a fost schimbat înainte de publicare pentru a indica faptul că acesta nu mai era asociat cu Netscape. Datorită acestui istoric, termenii TLS și SSL sunt uneori folosiți în mod interschimbabil. Criptarea TLS poate ajuta la protejarea aplicațiilor web de atacuri precum încălcări de date și atacuri DDoS. În plus, HTTPS-ul protejat de TLS devine rapid o practică standard pentru site-uri web. (What is Transport Layer Security (TLS)?, fără dată)

Interpretare rezultate: considerăm că lipsa SSL ca metodă de criptare nu constituie un motiv de îngrijorare, dar este un lucru bine de știut despre propriul site.

Rezultatele scanării au arătat: (Ristic, 2020)

TLS 1.3	Da
TLS 1.2	Da
TLS 1.1	Nu
TLS 1.0	Nu
SSL 3	Nu
SSL 2	Nu
IE 11 / Win 7 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
IE 11 / Win 8.1 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
IE 11 / Win Phone 8.1 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
IE 11 / Win Phone 8.1 Update R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
Safari 6 / iOS 6.0.1	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
Safari 7 / iOS 7.1 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
Safari 7 / OS X 10.9 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
Safari 8 / iOS 8.4 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)
Safari 8 / OS X 10.10 R	Serverul a trimis o eroare fatală: handshake_failure (nu s-a putut face handshake, proces necesar)

Tabel 2.5. Exemple vulnerabilități găsite pe olivian.ro, conform ssllabs.com (Ristic, 2020)

2.6. Testare siteguarding.com

Am rulat acest test la nivel de server, și nu de URL (subfolder 5.0 sau 5.4 creat pe olivian.ro), pentru că la nivel de URL furnizează doar informațiile despre linkurile de pe site către alte site-uri. Ori, pe o instalare proaspătă de WordPress nu există aproape niciun link extern, nu sunt rezultate relevante, dovada și faptul că testarea linkurilor de pe site-ul olivian.ro nu a reflectat niciun link. Am considerat relevantă însă testarea la nivel de server.

În primul rând, conform analizei "Server - Name: LiteSpeed". Este util de știut că cu o simplă scanare se poate afla tipul de server al hostingului (găzduirii). (Website Security | Website Antivirus | Website Firewall | Website File Monitoring | Website Backup | Malware, Virus, Trojan Removal | Blacklist Removal | SiteGuarding, fără dată)

Un prim test este la nivel de "Global Blacklists" (liste negre, cu site-uri nocive, la nivel global), pentru care s-au interogat 83 de site-uri, referitor la site-ul olivian.ro, și apoi la nivel de "SPAM Blacklists" (liste de spamming, trimitere de mesaje comerciale nesolicitate, la nivel global), unde au fost interogate 95 de site-uri. Toate testele au fost pozitive, în sensul că nu a fost raportat nicio prezență de tip blacklist / spam. Așadar, toate bune. (Website Security | Website Antivirus | Website Firewall | Website File Monitoring | Website Backup | Malware, Virus, Trojan Removal | Blacklist Removal | SiteGuarding, fără dată)

Ca o concluzie, rezultatele testelor au fost pozitive.

2.7. Testare portswigger.net

Am folosit software-ul Burp Suite Enterprise Edition, o variantă de test disponibilă două săptămâni cu toate funcționalitățile, pentru a verifica site-ul olivian.ro. După instalarea locală a programului, am rulat mai multe teste.

Ca observații de corectat, și soluții, la final pentru fiecare problemă găsită, se pot menționa: (Scan Remediation Report, 2020)

- Securitatea strictă a transportului nu este aplicată. Problema e că site-ul nu împiedică utilizatorii să se conecteze la ea prin conexiuni necriptate. Deoarece HSTS este un protocol de „încredere în prima utilizare” ("trust on first use" - TOFU), un utilizator care nu a accesat niciodată aplicația nu a văzut niciodată antetul HSTS și, prin urmare, va fi în continuare vulnerabil la atacurile SSL. Am mai discutat despre aceste aspecte.

- Câmp de parolă cu completare automată activată (autocomplete). Majoritatea browserelor au capacitatea de a memora credențialele (datele de logare) utilizatorilor care sunt introduse în formulare HTML. Aceste date pot fi capturate de un atacator care are controlul asupra computerului utilizatorului. De asemenea, pot fi vulnerabilități separate a aplicației, cum ar fi scripturile de pe mai multe site-uri. Soluția ar fi aici introducerea atributului autocomplete="off" în eticheta FORM (pentru a proteja toate câmpurile de formular) sau în etichetele INPUT relevante (pentru a proteja câmpurile individuale specifice).
- Scurgeri de referral (referință) pentru mai multe domenii (cross-domain Referer leakage). Când un browser web face o solicitare pentru o resursă, de obicei se adaugă un antet HTTP, numit antet "Referer", care indică adresa URL a resursei sursă. Dacă resursa solicitată se află pe un domeniu diferit, atunci antetul Referer este încă în general inclus în cererea cu mai multe domenii. Dacă adresa URL de origine conține informații sensibile din șirul de interogare, cum ar fi un simbol de sesiune, atunci aceste informații vor fi transmise celuilalt domeniu. Dacă celălalt domeniu nu este pe deplin de încredere în aplicație, atunci acest lucru poate duce la un compromis de securitate. Soluția este utilizarea antetului HTTP Referer-Policy pentru a reduce șansa de a fi dezvăluite către terți.
- Cookie-uri fără HttpOnly flag. Dacă atributul HttpOnly este setat pe un cookie, atunci valoarea cookie-ului nu poate fi citită sau setată de JavaScript din partea clientului. Această măsură face ca anumite atacuri din partea clientului, cum ar fi scripturile încrucișate (cross-site scripting), să fie ușor mai greu de exploatat. Soluția este setarea flag-ului HttpOnly pe toate cookie-urile.
- Răspuns HTTPS în memoria temporară (care poate fi cache-uit). Browserele pot stoca o copie locală în cache a conținutului primit de la serverele web. Unele browsere, inclusiv Internet Explorer, fac caching inclusiv pentru site-uri la care se conectează prin HTTPS. Acestea ar putea fi accesate de alți utilizatori ai aceluiași calculator/dispozitiv. Soluția este adăugarea următoarelor HTTP headers:
    - `Cache-control: no-store`
    - `Pragma: no-cache`
- HTML nu specifică charset (codarea caracterelor, de forma UTF-8, de exemplu). Dacă un răspuns în headers afirmă că acesta conține conținut HTML, dar nu specifică un set de caractere (character set), atunci browserul poate analiza HTML-ul pentru a

determina ce set de caractere folosește. Prezența caracterelor non-standard oriunde în răspuns poate determina browserul să interpreteze conținutul folosind un set standard de caractere diferit (precum UTF-7, de exemplu). Există vulnerabilități de scripturi cross-site în care pot fi utilizate codări non standard precum UTF-7 pentru a ocoli filtrele de securitate ale unei aplicații. UTF-7 - 7-bit Unicode Transformation Format, este un format de transformare a caracterlor în Unicode, format de texte definit de către Unicode Consortium. Pentru fiecare răspuns care conține conținut HTML, aplicația ar trebui să includă în antet (headers) o directivă precum:

    - `charset=ISO-8859-1`

- Răspuns Frameable (potențial clickjacking) - acest aspect a fost discutat anterior.

2.8. Testare apptrana.com

Folosind soluția on-line (în cloud) https://apptrana.com/, am testat serverul olivian.ro. A rezultat o listă de 12 vulnerabilități cu importanță medie care necesită protecție folosind reguli personalizate. De asemenea, o listă de alte 30 vulnerabilități cu importanță tot medie, dar și scăzută, sau doar ca notă de informare.

Rezultatele testării, prezentăm doar cele mai relevante aspecte: (AppTrana, fără dată)

- Content Security Policy (CSP)/X-Frame-Options nesigură - am discutat despre acest aspect anterior, nu mai detaliem aici.
- Activarea cachingului browserului - un punct anterior discutat a fost că în headers nu am activat caching, răspunsul nostru a fost la acel moment că am activat browser caching. Ei bine, se pare că și browser caching poate fi o problemă. Exemple de lucruri care pot fi expuse prin browser caching: istoria navigării, HTTP headers, datelor introduse în formulare HTML, cookie-uri, istoric tranzacții și alte tipuri de date. Soluția ar fi dezactivarea pluginurilor de caching de pe olivian.ro.
- Formular HTML fără protecție CSRF - falsificarea cererii Cross-Site (Cross-Site Request Forgery - CSRF / XSRF) este o vulnerabilitate în care atacatorul îi păcălește pe victimă să facă o solicitare pe care victima nu a făcut-o. Se pot folosi pluginuri de WordPress care să rezolve acest aspect.
- Listing directoare - acest lucru înseamnă că oricine poate vedea conținutul directorului. Se poate proteja împotriva acestui aspect cu un fișier .htaccess în folderul respectiv. Afișarea fișierelor dintr-un folder poate include scripturi CGI, fișiere de date sau fișiere

de tip backup (CGI, Computer-Generated Imagery, se referă la aplicarea graficii computerizate pentru a crea fișiere imagini).

- Dezvăluirea adresei IP interne - exemplu concret fiind URL-ul https://olivian.ro/?attachment_id=http%3A%2F%2F127.0.0.1%3A80 - aici este o problemă faptul că datele afișate pot duce la atacuri asupra hostingului, aplicației web sau a utilizatorilor. Pe de altă parte, IP-ul respectiv afișat în situația curentă este considerat sigur, așadar nu este o problemă.
- Metoda HTTP OPTIONS activată - metoda HTTP OPTIONS furnizează o listă a metodelor care sunt acceptate de serverul web. Este o solicitare despre opțiunile de comunicare disponibile în procesul de solicitare / răspuns. Acest lucru poate expune informații sensibile. Un plugin de securitate ar putea rezolva acest aspect.
- Fișier de tip documentare detectat la adresa https://olivian.ro/wp-content/plugins/a3-lazy-load//readme.txt - aceste fișiere, de documentare a aplicației (exemplu: readme.txt, changelog.txt), pot include informații numele aplicației, versiunea. Este de preferat ca astfel de fișiere sa fie ascunse. Am fost surprinși că un plugin a lăsat disponibil acest fișier printr-o navigare tipică, însă dacă am șterge acest fișier, la prima actualizare a plugin-ului fișierul ar fi pus înapoi în mod automat. Soluția ar fi blocarea accesului către fișier prin fișierul .htaccess.
- Fișier de arhivat (posibil) detectate - a fost găsit în directorul hostingului un posibil fișier arhivă. În urma testării concrete a acestei situații, se observă că fișierul cu pricina - https://olivian.ro/wp-content/uploads// - este de fapt un folder, și are o funcție de caching activată, și de asemenea, compresie, dar este un folder tipic de WordPress, așadar nu este o problemă reală. Ar fi putut fi o problemă un astfel de fișier dacă ar fi conținut o arhivă a site-ului, există pluginuri de backup care fac astfel de fișiere și nu ascund suficient de bine calea către arhivă. În situația de față, însă, nu este o problemă.
- Posibile directoare / fișiere sensibile detectate, anume https://olivian.ro/info.php - acestea sunt genul de directoare / fișiere care pot expune informații sensibile. Într-adevăr, în urma verificării, a rezultat că este un fișier al serviciului de hosting, care dă informații despre varianta de PHP folosită, și ce funcții PHP sunt activate. Soluția ar fi blocarea accesului via .htaccess.

2.9. Testare detectify.com

Conform testului de pe detectify.com, au fost găsite 5 probleme cu importanță medie, 6 cu prioritate scăzută, și 8 (nu le vom lista pe toate, două sunt complet irelevante) cu titlu informativ: (Monitor Your Site's Security | Detectify, fără dată)

- Importanță medie - Falsificare Cross Site Request (CSRF / XSRF) - discutat anterior.
- Importanță medie - Dezvăluirea completă a căilor (Full Path Disclosure) - serverul afișează calea completă (URL-ul întreg) către un script. Combinată cu alte vulnerabilități, ar putea crește rata de succes a atacatorilor. Soluția ar fi ascunderea acestei căi.
- Importanță medie - Linkuri externe folosind `target='_blank'`. Acestea au acces parțial la pagina de legătură prin intermediul `window.opener`. Paginile (sursă și destinație) vor fi în legătură. Partea problematică în a găsi o soluție la această problemă e că soluția de a deschide linkurile externe în fereastră dedicată a fost adăugată de noi pe site-ul olivian.ro, în mod automat, printr-un plugin. În prezent, nu vedem o soluție asupra acestei probleme. Nu este o funcționalitate standard în WordPress, dar în general noi considerăm că este util vizitatorilor ca un click pe un link către alt site să deschidă pagina în fereastră de browser dedicată.
- Importanță medie - X-Frame-Options / Lipsă Header (Clickjacking) - discutat anterior.
- Importanță medie - Scurgeri de tokenuri prin HTTP GET - Tokenul este adus ca parte a adresei URL într-o solicitare GET. Am analizat problema semnalată, ține de serviciul de hosting newsletter folosit pe olivian.ro, nu considerăm că este o problemă gravă.
- Importanță scăzută - Dezvăluire folosire tehnologie Apache - serverul HTTP dezvăluie ce tip de tehnologie este folosită pe în hosting, ceea ce poate fi folosită pentru a căuta vulnerabilitățile cunoscute. Pe de altă parte, Apache este folosit, conform unei surse, de aproximativ 38,5% din site-urile al căror server este cunoscut. (Usage Statistics and Market Share of Apache, May 2020, fără dată) Așadar, și dacă serverul folosit ar fi ascuns vizitatorilor, tot ar fi o țintă potențială de testat.
- Importanță scăzută - Scurgerea token-ului de autentificare a canalului lateral (Side Channel Authentication Token Leakage) - problema apare la redarea CSS, iframe-uri și un atacator poate face acest lucru pe o pagină aflată sub controlul său. Acesta poate extrage textul din ea măsurând diferența de timp. Acest text include și tokenuri.

Problema ține, din nou, de serviciul de newsletter folosit pe site, și în prezent nu avem o soluție.

- Importanță scăzută - Atributul frame-ului de tip sandbox nu este implementat - tag-ul iframe are o valoare a atributului sandbox lipsă sau invalidă, ceea ce înseamnă că deoarece conținutul în iframe nu are restricții, acesta trebuie considerat ca fiind de încredere. Am testat aparițiile în site ale problemei, sunt referitoare la cod încorporat (embed) YouTube, în principiu considerăm că este destul de sigur.
- Importanță scăzută - Scurgerea informațiilor de tip metadate - fișierele pot conține informații precum EXIF, IPTC sau XMP (diferite tipuri de metadate, informații suplimentare stocate în fișiere). Am testat pe site și se referă la niște lucruri care nu conțin date sensibile.
- Importanță scăzută - Strict-Transport-Security / Lipsește antetul pentru aceasta - detaliat anterior.
- Importanță scăzută - Referrer-Policy / antet lipsă - detaliat anterior.
- Doar cu titlu informativ - Software cu amprentă digitală vizibilă - aplicația a găsit newsletterul și date despre hostingul folosit. Nu avem o soluție concretă, alta decât schimbarea furnizorilor de hosting și newsletter.
- Doar cu titlu informativ - X-Content-Type-Options / antet lipsă - discutat anterior.
- Doar cu titlu informativ - Comentarii HTML - comentariile pot conține, teoretic, informații care nu sunt destinate publicului, dar în situația concretă de pe olivian.ro nu este cazul.
- Doar cu titlu informativ - Resurse externe - acest aspect este firesc pentru majoritatea site-urilor cu un conținut numeros, este firesc ca o parte din resurse să fie găzduite pe alte site-uri.
- Doar cu titlu informativ - Hosting vizibil - în prezent, nu avem o soluție în a ascunde datele despre hostingul folosit, alta decât alegerea unui alt furnizor.
- Doar cu titlu informativ - Lipsește politica DMARC - Un atacator va putea să difuzeze e-mailuri provenind din orice subdomeniu având fie o înregistrare A, AAAA sau MX (tipuri de înregistrări care se pot face pentru serverul de email). Acest lucru este posibil indiferent dacă există sau nu politici SPF (Sender Policy Framework - framework pentru politica de trimitere, o metodă de autentificare emailuri).

Capitolul 3. Testare WordPress la nivel de versiune diferită (5.0 vs. 5.4, testat comparativ)

3.1. Testare pentest-tools.com

Legat de testul variante 5.0 WordPress, soluția pentru majoritatea problemelor este actualizarea WordPress. Erorile găsite: (Website Vulnerability Scanner - Online Scan for Web Vulnerabilities | Pentest-Tools.com, fără dată)

- Nivel de risc 7,5/10 - CVE-2018-20148 - În WordPress 5.x înainte de 5.0.1, se pot face atacuri de injectare de obiect PHP prin metadate elaborate într-un apel wp.getMediaItem - apel XMLRPC. Acest lucru este cauzat de gestionarea greșită a datelor serializate la phar:// din funcția `wp_get_attachment_thumb_file` din wp-includes/post.php.

- Nivel de risc 6,8/10 - CVE-2019-9787 - WordPress înainte de 5.1.1 nu filtrează în mod corespunzător conținutul comentariilor, ceea ce duce la executarea codului la distanță de către utilizatori neautentificați într-o configurație implicită. Protecția CSRF este folosită greșit. Eroarea este legată de wp-admin/includes/ajax-actions.php și wp-includes/comment.php.

- Nivel de risc 6,5/10 - CVE-2019-8942 - WordPress 5.x înainte de 5.0.1 permite executarea de la distanță a codului, deoarece o intrare post meta `_wp_attached_file` poate fi modificată într-un șir arbitrar. Exemplu: un string care se încheie cu .jpg?file.php. Un atacator cu privilegii de autor poate executa cod arbitrar.

- Nivel de risc 5,8/10 - CVE-2019-16220 - În WordPress înainte de 5.2.3, validarea și sanitizarea (sanitize) unui URL în `wp_validate_redirect` în includes/pluggable.php ar putea duce la o redirecționare deschisă.

- Nivel de risc 5,5/10 - CVE-2018-20147 - În WordPress 5.x înainte de 5.0.1, autorii ar putea modifica metadatele pentru a evita restricțiile prevăzute la ștergerea fișierelor.

Alte probleme au fost deja tratate anterior, nu le vom mai dezbate: "Listingul în directoare este activat", "Software și tehnologie server făcute publice", "Lipsesc antete / headers de securitate

HTTP" (sunt aproape 4 tipuri de anteturi, toate au fost discutate anterior), "Fișierul Robots.txt a fost găsit" (acesta considerăm că nu este un risc de securitate, din descrierea din raport și din folosire anterioară). (Website Vulnerability Scanner - Online Scan for Web Vulnerabilities | Pentest-Tools.com, fără dată)

În testul variantei 5.4 nu au fost găsite erori suplimentare, dar, în mod firesc, o parte din problemele specifice variantei 5.0 nu s-au mai regăsit aici.

3.2. Testare immuniweb.com

Am realizat două teste, unul pentru varianta 5.0 și unul pentru 5.4. Pentru 5.4 rezultatele au arătat, în plus față de erori menționate deja: (Website Security Test of olivian.ro (89.33.25.24), fără dată)

- Nivel eroare: 9,7/10 - critică - CVE-2019-8943 CWE-22 - Traversarea directorului (path traversal). Discutată anterior, e o eroare la nivel de server.
- Nivel eroare: 7,7/10 - ridicată - CVE-2020-11027 CWE-640 - Mecanism slab de recuperare a parolei pentru parola uitată. Există soluții teoretice pentru astfel de probleme.
- Nivel eroare: 5,7/10 - medie - CVE-2019-17673 CWE-20 - Atac de tip spoofing, o situație în care persoană sau un program se identifică ca altul, pentru a obține beneficii. Am discutat despre probleme similare.
- Nivel eroare: 4,8/10 - medie - CVE-2019-17671 CWE-284 - Control necorespunzător pentru acces. Face parte din modelul de securitate AAA (Autentificare, Autorizare, Evidență - Accounting). Varianta 5.0 de WordPress are probleme la acest aspect.
- Nivel eroare: 4,7/10 - medie - CVE-2019-17675 CWE-20 - validare incorectă a introducerii datelor. Soluții teoretice pentru astfel de probleme există, și pot fi implementate.

Testarea variantei 5.4 nu a relevat informații suplimentare. (Website Security Test of olivian.ro (89.33.25.24), fără dată)

3.3. Testare sucuri.net

Ca observații pentru testarea WordPress 5.0: (https://olivian.ro/wp50/, fără dată)

- Monitorizarea site-ului web - Nu a fost detectată. Cei de la Sucuri oferă acest serviciu, așa că importanța acordată de site observației de securitate poate fi influențată (bias).

- Firewall găzduire / aplicație - Nu a fost detectat. Aceeași problemă. Soluțiile pentru aceste prime două puncte sunt folosirea unor soluții externe de securitate.
- Site-ul este neactualizat - Suntem conștienți pe deplin de această eroare. A fost una din condițiile de instalare, să păstrăm varianta de WordPress neactualizată, pentru a preveni rezultate diferite în urma testelor - desigur, WordPress ar trebui să fie cât mai similar de la o testare la alta.
- `display_errors` (PHP) este activat, această setare ar trebui să fie oprită în mediul de producție. Se poate utiliza `log_errors` în schimb. Da, este util de știut acest lucru pentru instalarea implicită.
- A fost prezentată și o eroare la o pagină care a returnat 404, dar nu este o eroare corectă, este firesc ca acea pagină să returneze 404, este o categorie cu zero rezultate (articole).
- Versiunea PHP afișată în anteturile HTTP - eroarea este cunoscută și la nivel de server.

Sunt și alte probleme depistate, dar au fost discutate deja anterior.

Scanarea 5.4 nu a dat erori suplimentare, unele din erorile de la 5.0 au fost remediate. (https://olivian.ro/wp54/, fără dată)

3.4. Testare upguard.com

Am testat ambele versiuni ale site-urilor - 5.0 (Free Website Security Scan | UpGuard, fără dată) și 5.4 (Free Website Security Scan | UpGuard, fără dată) - și, pentru ambele, am obținut nota F. În sistemul de notare american, asta e o notă foarte slabă. Pe de altă parte, considerăm că profilul specific al business-ului, care are tot interesul să vândă soluții de securitate, e posibil să influențeze acest rezultat. Nu au fost disponibile alte detalii despre nota obținută.

3.5. Testare webcookies.org

Pentru https://olivian.ro/wp50/, observația nouă a acestui test, față de ce deja am prezentat, arată că pagina încărcă 3 fișiere JavaScript ale unor terțe părți (third-party) și 10 fișiere CSS (Cascading Style Sheets - stiluri în cascadă, standard formatarea elemente HTML), dar nu utilizează Sub-Resource Integrity pentru a preveni un atac, dacă un CDN (content delivery network, or content distribution network - rețea de livrare de conținut, rețea internațională de servere proxy) terț este compromis. (olivian.ro | Privacy & security report #30424440, fără dată)

Soluția ar consta în utilizarea Sub-Resource Integrity.

În testul https://olivian.ro/wp54/ numărul de fișiere CSS devine 9. (olivian.ro | Privacy & security report #30424448, fără dată)

3.6. Testare nstalker.com

Am folosit suita de programe nstalker.com în mai multe testări pe cele două instanțe de WordPress - https://olivian.ro/wp50/ și https://olivian.ro/wp54/.

Am întâlnit, spre surprinderea noastră, probleme noi nu la nivel de folder (50 sau 54), ci la nivel de server, așadar comune ambelor instalări: (N-Stalker Free Edition Version X, fără dată)

- Serverul web impune propria ordine de criptare SSL/TLS - am fost puțin surprinși de această observație, deoarece am găsit și recomandarea ca ordinea de cifrare care pentru un server să fie implementată în mod obligatoriu într-o anumită ordine. (4.13. Hardening TLS Configuration Red Hat Enterprise Linux 7 | Red Hat Customer Portal, fără dată)
- Serverul web suportă SSL/TLS Forward Secrecy Cipher (PFS) - mai degrabă decât folosirea RSA pentru schimbul de chei de sesiune, ar trebui folosit schimbul de chei Diffie-Hellman (ECDHE). ECDHE este mult mai rapid decât DH (Diffie-Hellman) obișnuit.

Note:

- SSL, Secure Sockets Layer - protocol criptografic pentru comunicații sigure pe Internet.
- TLS, Transport Layer Security - Succesor SSL - protocol criptografic pentru comunicații sigure pe Internet.
- FS / PFS - forward secrecy sau perfect forward secrecy - funcții pentru transferul cheilor criptografice.
- RSA, Rivest–Shamir–Adleman - algoritm criptografic cu chei publice.
- ECDH, Elliptic-curve Diffie–Hellman - protocol pentru schimbul cifrat al cheilor cripotgrafice.
- DH, Diffie–Hellman key exchange - metodă de transfer chei cripotgrafice.

3.7. Testare hackertarget.com

Nu am obținut rezultate noi, în plus față de testele anterioare. (28 Online Vulnerability Scanners & Network Tools | HackerTarget.com, fără dată)

3.8. Testare probely.com

Ca rezultate suplimentare, am obținut permisiunea browser-ului de a vedea anumite tipuri de conținut, ce nu ar trebui să fie accesibil - "Browser content sniffing allowed". Problema este comună ambelor instalări de WordPress - 5.0 (Probely, fără dată) și 5.4 (Probely, fără dată). Soluția constă în modificarea header-elor transmise.

3.9. Testare appscan.com

Folosind aplicația în cloud (online) appscan.com, am obținut 3 erori noi pentru WordPress 5.0 (HCL AppScan on Cloud, fără dată), acestea ne mai regăsindu-se și în raportul pentru 5.4 (HCL AppScan on Cloud, fără dată):

- Divulgare informații CMME (Content Management Made Easy - sistem facil de gestionare al conținutului unui site) - faptul că este ușor de detectat că site-urile folosesc platforma WordPress le predispune la mai multe vulnerabilități. (CMME Cross Site Scripting And Information Disclosure Vulnerabilities, fără dată)
- File Parameter Shell Command Injection (injecția comenzii shell în parametrii fișierului) - injecția comenzilor este un atac în care obiectivul este executarea comenzilor arbitrare asupra sistemului de operare gazdă printr-o aplicație vulnerabilă. În atac, comenzile furnizate de atacator sunt de obicei executate cu privilegiile aplicației vulnerabile. (Command Injection | OWASP, fără dată)
- Model de eroare de tipar al bazei de date găsit - în varianta gratuită a aplicației nu erau date detalii despre problemă, dar probabil eliminarea caracterelor potențial periculoase nu a fost efectuată corect la introducerea datelor utilizatorului. Acest lucru face posibilă vizualizarea, modificarea sau ștergerea intrărilor și tabelelor bazei de date. (HCL AppScan on Cloud, fără dată)

3.10. Testare rapid7.com

La testarea inițială, pe WordPress 5.0, am obținut o eroare nouă, "Scurgeri de memorie (Memory Leaks) în Javascript". (Rapid7, fără dată) Acestea pot cauza probleme ca încetinirea, oprirea completă, latența ridicată sau probleme cu alte aplicații. (Peyrott, 2016) Până aici, nicio surpriză.

Surpriza a venit la testarea variantei 5.4, unde am obținut erori pe care le-am obținut și în alte testări realizate pe site, dar erau, pentru aplicația de față, mult mai multe față de varianta 5.0,

și chiar am găsit o eroare nouă: colectarea informațiilor personale sensibile. Este o eroare de securitate mai puțin tehnică, dar importantă, totuși. De remarcat că acest tip de eroare este mai ușor de detectat într-o analiză umană decât în una automată. Și, totuși, analiza a găsit-o. Este prima observație semnificativă a variantei de WordPress 5.4, ca eroare în plus față de testele pe 5.0. (Rapid7, fără dată)

3.11. Testare zaproxy.org

Folosind OWASP ZAP 2.9.0, rulat local, am găsit o singură eroare, în ambele instanțe de WordPress - 5.0 și 5.1 - "Dezvăluire timpului folosit de server - Unix". Considerăm însă că problema este departe de a fi una importantă, așa cum a fost marcată și în raportul primit. (OWASP ZAP, fără dată)

3.12. Testare qualys.eu

Nu am găsit probleme noi față de testele anterioare. Am avut anumite probleme în a rula testul pe varianta 5.0, la un moment dat returna puține date, deși scanarea rula. Pe 5.4 a funcționat fără probleme aparente. Punem în anexe imaginile cu rezultatele, nu e nimic nou. (Qualys Security and Compliance Suite Login, fără dată)

3.13. Testare hackertarget.com

Folosind un instrument online disponibil pe site-ul hackertarget.com, am testat ambele variante ale site-ului - WordPress 5.0 și 5.4. O eroare nou obținută, în ambele teste, a fost aceea de "Enumerare utilizatori platformă". Cu alte cuvinte, în WordPress se afișează lista de utilizatori ai platformei. Desigur, această practică nu este recomandată. (WordPress Security Scan | HackerTarget.com, fără dată)

Capitolul 4. Potențiale soluții - cum se poate securiza WordPress?

4.1. De ce securizare WordPress?

Alături de avantajele popularității WordPress, există un dezavantaj important: este o țintă pentru hackeri. Pe de altă parte, tot popularitatea a determinat comunitatea WordPress să identifice amenințările și să existe soluții pentru probleme. (Onishi, 2013)

4.2. Potențiale soluții securizare WordPress

4.2.1. Mutarea fișierului wp-config.php

Dacă WordPress se află pe un server care are un director pentru fișierele publice (cum ar fi public_html sau www), se poate muta fișierul wp-config.php în afara directorului public, și fără nicio modificare, WordPress va funcționa în continuare bine. Este necesară o modificare în wp-load.php. (Onishi, 2013)

4.2.2. Mutarea directorului wp-content

Acest director nu este esențial pentru WordPress, deoarece nu conține fișiere de bază care ar avea nevoie de actualizare. Doar temele și pluginurile se actualizează. Ar trebui adăugate în fișierul wp-config.php următoarele constante:

```
define('WP_CONTENT_DIR', $_SERVER['DOCUMENT_ROOT'] . '/content');
define('WP_CONTENT_URL', 'http://review.dev/content');
define('WP_PLUGIN_DIR',          $_SERVER['DOCUMENT_ROOT']          .
'/content/plugins');
define('WP_PLUGIN_URL', 'http://review.dev/content/plugins');
define('PLUGINDIR', $_SERVER['DOCUMENT_ROOT'] . '/content/plugins');
```

4.2.3. Adăugarea de pluginuri care pot preveni atacuri de tip DoS (denial-of-service attack)

Există posibilitatea ca un site să nu fie spart (hacked), dar atacurile de tip refuz-de-serviciu (DoS sau DDoS) pot opri forțat serverul web. Blogurile WordPress sunt foarte vulnerabile deoarece sunt site-uri-tip care au de multe ori aceeași structură. (Wordpress: Learn Wordpress In A DAY! - The Ultimate Crash Course to Learning the Basics of Wordpress In No Time

(Wordpress, Wordpress Course, Wordpress ... Wordpress Books, Wordpress for Beginners) Kindle Edition, 2015)

DDoS este executat folosind prin mai multe locații de pe glob. În fiecare an milioane de dolari sunt irosiți din cauza acestui atac notoriu de securitate web. Un plugin de securitate poate opri unele din astfel de atacuri (de exemplu, plugin de tip firewall). De asemenea, un serviciu de găzduire / hosting poate fi util în acest sens. (Rehman, 2020)

4.2.4. Alegerea unui plugin de backup corespunzător

Un aspect analizat în lucrarea "Security Evaluation of WordPress Backup" (Evaluarea securității WordPress backup - notă: se referă la pluginuri de backup) privește securitatea pluginurilor de backup create pentru WordPress din perspectiva securității - date sensibile, module vulnerabile, greșeli frecvente și impact. Una din concluzii a fost că majoritatea pluginurilor de rezervă nu folosesc metode criptografice puternice în crearea numelui backupului. Acest lucru face ca numele fișierului de rezervă să fie previzibil și de aici probleme de securitate. (Cernica, Popescu și Tiganoaia, 2019)

4.2.5. Adăugarea unui plugin de filtrare și prevenție comentarii SPAM

Există spammeri care au programe automate care scanează și încearcă să identifice site-uri WordPress, apoi lasă comentarii cu link-uri pentru reclame. Acest lucru poate fi destul de enervant, din cauza notificărilor prin e-mail. Un plugin dedicat poate rezolva acest lucru, inclusiv cel standard al WordPress, Akismet Anti-Spam. (Kelsey, 2012)

4.2.6. Folosirea unor pluginuri care pot crește securitatea site-ului

Dacă se folosește un plugin de securitate WordPress, se pot seta setări mai simplu unele elemente importante și recomandate de securitate. (Chahal, 2020)

Exemple de pluginuri care pot ajuta la securizarea unui blog:

- Akismet Anti-Spam - unul din cele mai populare pluginuri, cu peste 5 milioane de instalări active în mai 2020, și care vine în mod implicit cu WordPress, deși dezactivat. Pluginul verifică comentarii și formulare de contact împotriva spamului.
- Conditional CAPTCHA - un plugin mult mai puțin folosit, acesta poate afișa un cod de verificare împotriva spamului (CAPTCHA) numai dacă Akismet identifică un comentariu drept spam. Este un plugin care face mai bun pluginul anterior menționat.

- Cookie Notice for GDPR & CCPA - permite să informarea utilizatorilor unui site despre cookie-uri. Ajută pentru respectarea legilor privind cookie-urile GDPR (Uniunea Europeană) și CCPA (SUA). GDPR, General Data Protection Regulation - Regulamentul general al Uniunii Europene privind protecția datelor. CCPA, The California Consumer Privacy Act - Legea privind confidențialitatea consumatorilor din California, un statut privind drepturile de confidențialitate și protecția consumatorilor rezidenților din California, Statele Unite ale Americii.

- Really Simple SSL – detectează automat setările și configurează un site să ruleze pe https.

- Protect Your Admin – ascunde URL-ul de administrator WP, redenumind URL-ul admin din /wp-admin sau /wp-login.php în altceva.

- Limit Login Attempts Reloaded – limitează numărul de încercări de conectare posibile. În mod implicit WordPress permite încercări nelimitate.

- Simple Trackback Validation - Pluginul efectuează un test simplu, pe toate trackback-urile primite, pentru a opri spamul de tip trackback (un fel de link de la un blog la altul; dacă blogul X adaugă un link la blogul Y, blogul Y poate afișa un trackback al linkului, motivând alte bloguri să adauge linkuri către blogul Y).

- Advanced noCaptcha & invisible Captcha (v2 & v3) - Afișează un de verificare formularul de comentarii, formulare de contact, logare, înregistrare, parolă pierdută, resetare parolă.

4.2.7. Evitarea instalării de plugin-uri nenecesare

Există numeroase plugin-uri WordPress disponibile pe web. Un utilizator ar trebui să se gândească de două ori înainte de a instala unele dintre pluginuri mai puțin populare sau pluginuri care nu au fost testate de utilizatori. Dacă un plugin nu este esențial, poate fi mai bine să nu fie folosit. (Król, 2019)

4.2.8. Eliminarea plugin-urilor și teme nefolosite

Plugin-urile și temele sunt o componentă de bază în WordPress. Este o procedură bună inspectarea unui site, găsirea pluginurilor și temelor inutile (nefolosite, sau fără utilitate) și

eliminarea lor. Este indicată eliminarea din interfața WordPress, dar și verificarea ulterioară că nu există date rămase în baza de date WordPress. (Bari, fără dată)

4.2.9. Evitarea de teme și pluginuri piratate

Diferite teme și pluginurile WordPress piratate (versiuni modificate ale unor teme premium) oferă acces gratuit la versiuni în mod normal premium. Acest lucru are riscuri de securitate, deoarece acestea conțin adesea malware. (Banu, 2020)

4.2.10. Oprirea logării direct prin wp-login.php

Pentru unele atacuri, ale unor boți mai puțin sofisticați, se poate bloca solicitarea POST la wp-login.php. (Messenlehner et al., 2014)

1. Sa adaugă o regulă de rescriere în fișierul .htaccess: (Messenlehner et al., 2014)

```
RewriteRule ^adresa-noua$ wp-login.php
```

2. În functions.php sau un plugin personalizat: (Messenlehner et al., 2014)

```
function schoolpress_wp_login_filter( $url, $path, $orig_scheme ) {
      $old = array( "/(wp-login\.php)/" );
      $new = array( "adresa-noua" );
      return preg_replace( $old, $new, $url, 1 );
}
add_filter( 'site_url', 'schoolpress_wp_login_filter', 10, 3 );
function schoolpress_wp_login_redirect() {
if ( strpos( $_SERVER["REQUEST_URI"], 'adresa-noua' ) === false ) {
      wp_redirect( site_url() );
      exit();
} }
add_action( 'login_init', 'schoolpress_wp_login_redirect' );
```

4.2.11. Actualizarea completă - WordPress, pluginuri și teme

Nu mult timp după lansarea WordPress 5.0, versiune pe care am testat-o și în lucrarea de față, un mesaj oficial al WordPress afirma: "WordPress 5.0.1 este disponibil acum. Aceasta este o versiune de securitate pentru toate versiunile de la WordPress 3.7 încoace. Vă recomandăm cu căldură să vă actualizați site-urile imediat." (Dunn, 2018)

Echipa din spatele WordPress rezolvă problemele de securitate la scurt timp după ce sunt găsite. Prin urmare, dacă dashboard-ul (interfața de administrare) are notificări că există actualizări WordPress, este utilă instalarea cât mai curând posibil. (Williams, 2019)

Conform statisticilor oficiale ale WordPress, 42,6% dintre utilizatori utilizează în continuare diverse versiuni mai vechi de WordPress. (Rehman, 2020)

4.2.12. Adăugarea unei funcții de tip wp_nonce_field pentru pluginuri

Funcția `wp_nonce_field` face parte dintr-o serie de măsuri de securitate care fac ca datele trimise să fie verificate că provin din paginile de administrare WordPress și nu dintr-o sursă externă. Prin adăugarea acestui apel funcțional, se adaugă un câmp text ascuns în formularul de configurare a unui plugin, cu informații care vor fi verificate la recepția datelor de tip POST: (Lefebvre, 2012)

```
wp_nonce_field( $action, $name, $referer, $echo );
```

Nonce este utilizat în scopuri de securitate pentru a proteja împotriva cererilor neașteptate sau repetate care ar putea provoca modificări nedorite. Mai exact, un nonce este un simbol unic generat de un site web pentru a identifica cererile viitoare către acel site (requests). (Ratnayake, 2013)

4.2.13. Implementarea unui cod de verificare (CAPTCHA) în formulare

O măsură uzuală de securitate pentru formulare este utilizarea codurilor CAPTCHA. CAPTCHA, Completely Automated Public Turing test to tell Computers and Humans Apart, este un Test Turing public, complet automat, pentru a distinge calculatoarele de oameni. Este o metodă automată de a determina dacă persoana care face testul este om sau bot. (Lefebvre, 2012)

4.2.4. Utilizarea unui manager de parole

Atâta timp cât un utilizator o parolă foarte bună pentru a intra în managerul de parole, celelalte parole vor fi în siguranță. (BDM's: The WordPress Guidebook, 2017)

4.2.15. Instalarea manuală de WordPress

Dacă un utilizator alege instalarea rapidă sau automată, mai degrabă decât instalarea manuală, e posibil să acorde prea multă încredere serviciului de hosting (găzduire). Sunt prea multe posibilități pentru găzduire / hosting de a nu oferi un serviciu optim - parolă insuficient de sigură, prefixul tabelelor bazei de date, codurile de autentificare specificate în wp-config.php

ș.a. La o instalare manuală, se pot configura cu ușurință aceste lucruri. De asemenea, se obține o înțelegere mai bună asupra a WordPress prin instalarea manuală. (Guruli, 2017)

4.2.16. Prevenirea injecțiilor de tip SQL

SQL (Structured Query Language) este un limbaj de interogare structurat pentru sisteme de manipulare a bazelor de date relaționale. Injecția SQL este un atac în care codul SQL este inserat sau anexat în parametrii de aplicației / funcțiile de introducere date ale utilizatorului care sunt transmise ulterior la un server SQL de bază pentru analiză și execuție. Dacă o aplicație Web, cum e WordPress, nu reușește să sanitizeze / igienizeze în mod corespunzător parametrii care sunt trecuți la instrucțiunile SQL create dinamic (chiar și atunci când se utilizează tehnici de parametrizare), este posibil ca un atacator să modifice construcția instrucțiunilor SQL back-end (în codul sursă al site-ului). În WordPress, exemple de locuri unde se poate încerca SQL injection sunt câmpurile de comentarii, paginile de logare în site, formularele de contact, formularele de newsletter, câmpul de căutare, sau datele din admin / backend / administrare (de exemplu, un utilizator cu roluri de administrare mai mici poate încerca operații specifice unui administrator cu permisiuni mai mari). (Clarke, 2012)

4.2.17. Folosirea unor parole sigure

Este necesar ca pentru toți utilizatorii cu drepturi de administrare a unei platforme WordPress, pentru conturile de administrare a serverului, FTP și baze date să fie alese parole care să fie greu de ghicit. Uneori, însă, nu se întâmplă acest lucru. Mai jos, o listă cu 25 de parole foarte populare ale anului, analizate după cât de des au apărut în scurgeri de date (data leaks). Datele au fost prezentate de site-ul SplashData. Acestea sunt: "123456", "123456789", "qwerty", "password", "1234567", "12345678", "12345", "iloveyou", "111111", "123123", "abc123", "qwerty123", "1q2w3e4r", "admin", "qwertyuiop", "654321", "555555", "lovely", "7777777", "welcome", "888888", "princess", "dragon", "password1", "123qwe". (Douglas, 2019)

Probabil, o listă de astfel de parole pentru România ar arăta diferit, este util însă de notat că folosind baze de date cu parole publicate anterior în scurgeri de informații, hackerii pot porni de la niște baze de date cu parole folosite anterior, în încercarea de a obține acces la un site.

4.2.18. Ascunderea mesajelor de eroare de logare

Implicit, la logare, WordPress afișează un mesaj care alertează dacă s-a introdus în mod eronat numele de utilizator sau parola greșite. Cu funcția de mai jos, mesajul devine generic (se adaugă în functions.php sau într-un plugin personalizat): (Messenlehner et al., 2014)

```
add_filter( 'login_errors', create_function('$a', '"Utilizator sau parole greșite.";') );
```

4.2.19. Alegerea de nume utilizator dificil de ghicit

Unii utilizatori de site-uri web se axează pe parole puternice și ignoră numele de utilizator. Dar dacă numele de utilizator este ușor de ghicit, se elimină una din ținte (combinația utilizator și parolă). În trecut, WordPress i-a încurajat pe oameni să folosească „admin" ca nume de utilizator. Este bine să se evite numele de utilizator precum admin, admin124 etc. De asemenea, numele de utilizator ar trebui să fie diferit de numele afișat în mod public pe site. (Banu, 2020)

Pentru a ascunde numele de administrator și pe site, se poate folosi o funcție dedicată. Aceasta este necesară pentru că un hacker poate adăuga ?author=1 după URL-ului și probabil va obține în acest fel utilizatorul cu drepturi de administrare. Rămâne doar un pas, astfel, cel de a ghici parola. Soluția este o funcție care trebuie adăugată în functions.php. (Attard, 2020)

```
add_action('template_redirect', 'bwp_template_redirect');
function bwp_template_redirect(){
  if (is_author())  {
      wp_redirect( home_url() ); exit;
  }
}
```
(Attard, 2020)

4.2.20. Evitarea de roluri de utilizator greșit alese

La crearea unui site WordPress, un cont administrativ este creat în implicit. Ulterior, se pot crea noi conturi de utilizator, cu diferite roluri. Fiecare rol are un set de drepturi și responsabilități. O greșeală poate fi atribuirea de roluri de administrator tuturor utilizatorilor. (Banu, 2020) Drepturile de administrare pe platforma WordPress sunt: (Banu, 2020)

- Administrator - Are acces complet.
- Editor - Poate gestiona și publica toate postările.
- Autor - Poate publica și gestiona postările proprii.

- Contribuitor - Poate scrie și postări, dar nu le poate publica.
- Abonat - Poate gestiona propriul profil.

4.2.21. Oprirea posibilității de a rula cod în dosare (foldere) necorespunzătoare

Un site WordPress este format din sute de fișiere și foldere, unele din ele în locuri comune cu alte site-uri web WordPress - exemplu, folderul Uploads. Este bine să se blocheze executarea de cod în foldere necunoscute, prin adăugarea următorului fișier .htaccess în folderele care trebuie blocate: (Banu, 2020)

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L] RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L] </IfModule>
# END WordPress
```

De asemenea, se poate bloca accesul în fișierul .htaccess principal:

```
<FilesMatch "\.(php|php\.)$">
Order Allow, Deny
Deny from all
</FilesMatch>
```

4.2.22. Rularea site-ului pe https

HTTP este considerat nesigur, pentru că permite interceptarea și furtul datelor. Pentru a proteja site-ul de astfel de atacuri de hack-uri, este utilă trecerea la HTTPS. HTTPS - Secure Hyper Text Transfer Protocol - este un protocol de comunicații sigure. HTTPS face ca datele care circulă între browserul unui vizitator și serverul site-ului să fie criptate. (Banu, 2020)

4.2.23. Alegerea unui hosting (găzduire) de calitate

În ceea ce privește găzduirea (hostingul) unui site pe WordPress, sunt opțiuni de hosting partajat (shared hosting), gestionat (managed hosting) și alte variante, precum VPS, Virtual Private Server - server virtual privat. Întărirea serverului (server hardening) este o componentă importantă a întăririi securității WordPress. Este importantă alegerea unui serviciu de găzduire

fiabil și sigur. De exemplu, o găzduire shared (partajată între mai multe site-uri) implică niște riscuri de securitate, chiar dacă prețul e mai mic, de obicei. (Chahal, 2020)

Un hosting WordPress sigur are de obicei: (Agrawal, 2019)

- Firewall la nivel de server pentru a atenua atacurile DDoS.
- Cel mai recent hardware și centru de date (datacenter) de top pentru securitate fizică.
- Sistemul de operare actualizat și cu cele mai noi actualizări de securitate.
- Sisteme de detectare a intruziunilor.

Printre furnizorii recomandați pe plan internațional: SiteGround, Bluehost, WPEngine, Kinsta hosting. (Agrawal, 2019)

4.2.24. Activarea autentificării WordPress în doi pași / cu doi factori

Cu autentificarea în doi pași / cu doi factori, în afară de parolă, se va introduce, pentru logare, în mod obligatoriu un cod suplimentar într-un timp dedicat. Metoda face ceva mai dificilă spargerea (hackingul) unei conectări la WordPress și minimizează reușita atacurilor de forță brută de succes. (Chahal, 2020)

4.2.25. Prevenirea atacurilor la nivel de server

Hackerii nu au nevoie să obțină acces la tabloul de bord WordPress pentru crea probleme, este suficient accesul la serverele pe care este găzduit un site. Soluție: hosting de încredere, parolele sigure, conexiuni SFTP pentru transferul datelor. SFTP, SSH File Transfer Protocol, este un protocol de rețea utilizat pentru transferul de fișiere sigur pe shell-ul securizat, iar SSH, Secure Shell, este un protocol de rețea criptografică pentru operarea serviciilor de rețea în siguranță printr-o rețea nesecurizată. (Siarto, 2010)

4.2.26. Dezactivarea metodei de urmărire HTTP (HTTP Trace)

Cross Site Tracing (XST) și Cross Site Scripting (XSS) atacă sisteme țintă care au funcționalitatea HTTP TRACE. XST, Cross-Site Tracing - este tracing printr-o întrepătrundere a unor site-uri, implică folosirea Cross-site Scripting (XSS) și a metodelor TRACE sau TRACK HTTP. HTTP TRACE este o caracteristică implicită pe majoritatea serverelor. Prin atacarea lor, se pot obține cookie-uri și alte informații folosind cereri de antet (header requests).

(Chahal, 2020) Se poate dezactiva funcționalitatea prin adăugarea în fișierul .htaccess a: (Chahal, 2020)

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

4.2.27. Blocarea fragmentelor de text (strings) potențial periculoase

Dacă se adaugă următorul cod în fișierul .htaccess, acest lucru ajută la prevenirea atacurilor XSS. Regulile elimină solicitările URL ale unor atacuri de injecții periculoase. (Chahal, 2020)

```
<IfModule mod_rewrite.c>
RewriteCond %{REQUEST_METHOD} ^(HEAD|TRACE|DELETE|TRACK) [NC]
RewriteCond %{QUERY_STRING} ../ [NC,OR]
RewriteCond %{QUERY_STRING} boot.ini [NC,OR]
RewriteCond %{QUERY_STRING} tag= [NC,OR]
RewriteCond %{QUERY_STRING} ftp: [NC,OR]
RewriteCond %{QUERY_STRING} http: [NC,OR]
RewriteCond %{QUERY_STRING} https: [NC,OR]
RewriteCond %{QUERY_STRING} mosConfig [NC,OR]
RewriteCond %{QUERY_STRING} ^.*([|]|(|)|<|>|'|"|;|?|*).* [NC,OR]
RewriteCond    %{QUERY_STRING}    ^.*(%22|%27|%3C|%3E|%5C|%7B|%7C).*
[NC,OR]
RewriteCond %{QUERY_STRING} ^.*(%0|%A|%B|%C|%D|%E|%F|127.0).* [NC,OR]
RewriteCond                                              %{QUERY_STRING}
^.*(globals|encode|config|localhost|loopback).* [NC,OR]
RewriteCond                                              %{QUERY_STRING}
^.*(request|select|insert|union|declare|drop).* [NC]
RewriteRule ^(.*)$ - [F,L]
</IfModule>
```

4.2.28. Blocarea încercărilor de conectare eșuate repetate

Dacă un utilizator are prea multe încercări de autentificare eșuate, acesta va fi blocat pentru o perioadă. Este o caracteristică disponibilă cu multe pluginurilor de securitate din WordPress și reprezintă una dintre apărările împotriva atacurilor cu forțe brute (brute-force attacks). (Chahal, 2020)

Poate fi utilă și utilizarea de CAPTCHA-uri (metodă automată de a determina dacă persoana care face testul este om sau bot) de pre-autentificare. Caracteristică este utilă pentru a opri bot-urile automate de la accesarea tabloului de bord WordPress, precum și pentru a trimite spam-ul nedorit prin formulare. Pentru aceasta e poate adăuga un plugin precum "Advanced noCaptcha & invisible Captcha (v2 & v3)".

4.2.29. Securizarea stației de lucru (computer sau laptop)

Prevenirea hackingului WordPress începe cu stația de lucru, un element care poate fi ușor de trecut cu vederea: computerul sau laptopul administratorului. Acesta ar trebui să fie fără malware și viruși. Un keylogger poate contribui la spargerea și a unora din cele mai bine protejate dintre site-uri web. Sistemul de operare, software-ul folosit pe el, inclusiv browserele de pe computer trebuie să fie nevirusate. (Attard, 2020)

4.2.30. Securizarea folderului wp-includes

Folderul wp-includes este o componentă centrală a WordPress și este important să nu fie accesibil potențialilor hackeri. Soluția este un cod pentru fișierul .htaccess. (Attard, 2020)

```
# Blocare fișiere de forma incude-only.
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteBase /
  RewriteRule ^wp-admin/includes/ - [F,L]
  RewriteRule !^wp-includes/ - [S=3]
  RewriteRule ^wp-includes/[^/]+\.php$ - [F,L]
  RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
  RewriteRule ^wp-includes/theme-compat/ - [F,L]
</IfModule>
# BEGIN WordPress
```
(Attard, 2020)

4.2.31. Setarea unor permisiuni de fișiere corespunzătoare

Regula generală pentru permisiuni de fișiere este de drepturi de tip 755 (acces de citire și execuție pentru oricine și acces scriere pentru proprietarul directorului) pentru directoare și 644 (fișierele pot fi citite și scrise de proprietarul fișierului și citite de utilizatori) pentru fișiere. Setarea poate fi făcută de hosting, sau manual cu comenzile de mai jos. (Attard, 2020)

Pentru directoare: (Attard, 2020)

```
find /path/to/your/wordpress/install/ -type d -exec chmod 755 {} \;
```

Pentru fișiere: (Attard, 2020)

```
find /path/to/your/wordpress/install/ -type f -exec chmod 644 {} \;
```

4.2.32. Permiterea accesului la wp-admin numai prin IP-uri filtrate

O modalitate foarte simplă și elegantă de a restricționa accesul la pagina de conectare și zona de administrare este prin filtrarea IP. IP address, Internet Protocol address, este un protocol de

Internet pentru transmiterea datelor. (Attard, 2020) Pentru a filtra accesul pe baza IP-ului, este util codul de mai jos pentru .htaccess (via Sucuri): (Attard, 2020)

```
<Files wp-login.php>
Order Deny, Allow
Deny from All
Allow from [Aici se adaugă adrese IP]
</Files>
```

Pentru IP-urile dinamice: (Attard, 2020)

`Allow from [Aici se adaugă domeniul]` (Attard, 2020)

Pentru directorul wp-admin: (Attard, 2020)

```
<FilesMatch ".*">
Order Deny, Allow
Deny from All
Allow from [Se adaugă IP-ul]
</FilesMatch>
```

Pentru IP-uri dinamice: (Attard, 2020)

```
Allow from [Aici se adaugă domeniul]
```

4.2.33. Actualizarea PHP la o versiune cât mai recentă

PHP este un element de bază pentru WordPress. PHP, Hypertext Preprocessor, pre-procesor de hipertext, este un limbaj de programare. La data de 14 mai 2020 a fost lansat PHP 7.4.6. Multe versiuni de WordPress nu solicită în mod obligatoriu actualizarea la cea mai recentă de PHP. Acest lucru este un risc, deoarece echipa din spatele PHP oferă asistență de securitate oricărei versiuni stabile de PHP doar pentru 2 ani. (Agrawal, 2019)

Aproximativ 71,8% din site-urile care rulează WordPress utilizează PHP-ul învechit. Este în general indicată actualizarea la ultima versiune de PHP, cu atenție, totuși, pentru că pluginuri sau tema neactualizate ar putea crea conflicte. (Agrawal, 2019)

4.2.34. Ascunderea versiunii de WordPress

Pentru situația în care administratorii nu actualizează periodic core-ul (fișierele principale) de WordPress, o vulnerabilitate este că, uneori, în mod implicit se afișează în codul sursă al site-ului versiunea de WordPress folosită. Pentru a evita acest lucru, se poate adăuga în functions.php linia de mai jos. (Agrawal, 2019)

`<?php remove_action('wp_head', 'wp_generator'); ?>` (Agrawal, 2019)

4.2.35. Setarea unei alerte Google pentru paginile indexate

Există un procedeu mai puțin cunoscut - se pot seta alerte Google pentru a primi o atenționare pe email de fiecare dată când Google indexează o nouă pagină pe domeniul unui site. Uneori, hackerii WordPress adaugă noi pagini și postări care nu sunt afișate în meniuri, dar este aceste pagini sunt indexate în Google. Cu o alertă setată, se poate primi o avertizare pe email. (Agrawal, 2019)

O variație a acestei metode este înregistrarea domeniului în servicii cum e Google Search Console sau Bing Webmaster Tools, și activarea notificărilor.

4.2.36. Ascunderea directorului plugin (pluginuri)

Folderul (directorul) pentru pluginuri (add-on-uri) /wp-content/plugins/ nu ar trebui să arate lista de directoare și fișiere din interiorul lui. Uneori, această listă e vizibilă. Pentru a anula acest lucru, este necesară crearea unui fișier .htaccess în directorul de pluginuri. (Agrawal, 2019)

```
# BEGIN WordPress
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
# Previne listing director
IndexIgnore *
# END WordPress
```
(Agrawal, 2019)

4.2.37. Dezactivarea afișărilor erorilor de bază de date

În versiunile mai vechi ale WordPress, dacă existau erori în baza de date MySQL, s-ar afișa eroarea exactă în browser, oferind informații potențial periculoase. Soluția este actualizarea versiunii WordPress la o versiune recentă, care prezintă doar un mesaj de eroare general, precum „eroare de conectare a bazei de date”. (Agrawal, 2019)

4.2.38. Protejarea împotriva atacurilor de tip SQL Injection

Un set de reguli .htaccess poate împiedica decodarea multor tipuri de SQL Injection (injecție de cod malițios în baza de date). SQL, Structured Query Language, este un limbaj de interogare

structurat pentru sisteme de manipulare a bazelor de date relaționale. Sursa codului: shane.kinsch.com. (Common WordPress Security Vulnerabilities 2020 With Fixes, 2019)

```
<IfModule mod_rewrite.c>
# Enable rewrite engine
RewriteEngine On
# Blochează cererei suspecte
RewriteCond %{REQUEST_METHOD} ^(HEAD|TRACE|DELETE|TRACK|DEBUG) [NC]
RewriteRule ^(.*)$ - [F,L]
# Blochează hack-ul WP timthumb
RewriteCond                                                %{REQUEST_URI}
(timthumb\.php|phpthumb\.php|thumb\.php|thumbs\.php) [NC]
RewriteRule . - [S=1]
# Blochează user agents și requests suspecte
RewriteCond                %{HTTP_USER_AGENT}                (libwww-
perl|wget|python|nikto|curl|scan|java|winhttp|clshttp|loader)
[NC,OR]
RewriteCond    %{HTTP_USER_AGENT}    (<|>|'|%0A|%0D|%27|%3C|%3E|%00)
[NC,OR]
RewriteCond                                                %{HTTP_USER_AGENT}
(;|<|>|'|"|\)|\(|%0A|%0D|%22|%27|%28|%3C|%3E|%00).*(libwww-
perl|wget|python|nikto|curl|scan|java|winhttp|HTTrack|clshttp|archiv
er|loader|email|harvest|extract|grab|miner) [NC,OR]
RewriteCond %{THE_REQUEST} \?\ HTTP/ [NC,OR]
RewriteCond %{THE_REQUEST} \/\*\ HTTP/ [NC,OR]
RewriteCond %{THE_REQUEST} etc/passwd [NC,OR]
RewriteCond %{THE_REQUEST} cgi-bin [NC,OR]
RewriteCond %{THE_REQUEST} (%0A|%0D) [NC,OR]
# Block MySQL injections, RFI, base64, etc.
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=http:// [OR]
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=http%3A%2F%2F [OR]
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=(\.\.//?)+ [OR]
RewriteCond %{QUERY_STRING} [a-zA-Z0-9_]=/([a-z0-9_.]//?)+ [NC,OR]
RewriteCond    %{QUERY_STRING}    \=PHP[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-
f]{4}-[0-9a-f]{4}-[0-9a-f]{12} [NC,OR]
RewriteCond %{QUERY_STRING} (\.\./|\.\.) [OR]
RewriteCond %{QUERY_STRING} ftp\: [NC,OR]
RewriteCond %{QUERY_STRING} http\: [NC,OR]
RewriteCond %{QUERY_STRING} https\: [NC,OR]
RewriteCond %{QUERY_STRING} \=\|w\| [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)/self/(.*)$ [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)cPath=http://(.*)$ [NC,OR]
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} (<|%3C)([^s]*s)+cript.*(>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} (\<|%3C).*iframe.*(\>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} (<|%3C)([^i]*i)+frame.*(>|%3E) [NC,OR]
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [NC,OR]
RewriteCond %{QUERY_STRING} base64_(en|de)code[^()]*\([^)]*\) [NC,OR]
```

```
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2}) [OR]
RewriteCond %{QUERY_STRING} ^.*(\[|\]|\(|\)|<|>).* [NC,OR]
RewriteCond %{QUERY_STRING} (NULL|OUTFILE|LOAD_FILE) [OR]
RewriteCond %{QUERY_STRING} (\./|\../|\.../)+(motd|etc|bin) [NC,OR]
RewriteCond %{QUERY_STRING} (localhost|loopback|127\.0\.0\.1) [NC,OR]
RewriteCond %{QUERY_STRING} (<|>|'|%0A|%0D|%27|%3C|%3E|%00) [NC,OR]
RewriteCond %{QUERY_STRING} concat[^\(]*\( [NC,OR]
RewriteCond %{QUERY_STRING} union([^s]*s)+elect [NC,OR]
RewriteCond %{QUERY_STRING} union([^a]*a)+ll([^s]*s)+elect [NC,OR]
RewriteCond                                              %{QUERY_STRING}
(;|<|>|'|"|\)|%0A|%0D|%22|%27|%3C|%3E|%00).*(/\*|union|select|insert
|drop|delete|update|cast|create|char|convert|alter|declare|order|scr
ipt|set|md5|benchmark|encode) [NC,OR]
# Vulnerabilitea PHP-CGI
RewriteCond %{QUERY_STRING} ^(%2d|\-)[^=]+$ [NC,OR]
#proc/self/environ? no way!
RewriteCond %{QUERY_STRING} proc\/self\/environ [NC,OR]
RewriteCond %{QUERY_STRING} (sp_executesql) [NC]
RewriteRule ^(.*)$ - [F,L]
</IfModule>
```
(Common WordPress Security Vulnerabilities 2020 With Fixes, 2019)

4.2.39. Restrângerea hotlinking-ului

Hotlinking se referă la încărcarea unei imagini de pe un server direct de pe adresa URL a altui hosting / altei găzduiri. Imaginea va fi încărcată de pe hostingul A, dar afișată pe site-ul B, ceea ce este un fel de furt, deoarece se utilizează lățimea de bandă a site-ului A. Este și un oarecare risc de securitate. Prevenirea hotlinking în Apache se face rapid, adăugând un cod în .htaccess. (Bari, fără dată)

```
RewriteEngine on
RewriteCond   %{HTTP_REFERER}   !^$   RewriteCond   %{HTTP_REFERER}
!^http(s)?://(www\.)?yourdomain.com [NC]
RewriteRule  \.(jpg|jpeg|png|gif)$  http://drive.google.com/hotlink-
placeholder.jpg [NC,R,L]
ReescrieEngine activat
RewriteCond% {HTTP_REFERER}! ^ $ RewriteCond% {HTTP_REFERER}! ^ Http
(e)?: // (www \.)? Yourdomain.com [NC]
RewriteRule    \.    (Jpg    |    jpeg    |    png    |    gif)    $
http://drive.google.com/hotlink-placeholder.jpg [NC, R, L]
```
(Bari, fără dată)

4.2.40. Modificarea prefixului tabelelor de baze de date implicite

Instalarea implicită WordPress folosește "wp_" ca prefix pentru tabelele din baza de date. Dacă acesta se schimbă, site-ul va fi mai puțin vulnerabil în fața hackerilor care încearcă injecții SQL și testează după prefixul generic - "wp_". (Messenlehner et al., 2014)

4.2.41. Evitarea uploadului de fișiere SVG

Fișierele SVG sunt extrem de nesigure. SVG, Scalable Vector Graphics, sunt un tip grafică vectorială proporționabilă, SVG este un limbaj pentru imagini 2D (folosește XML). Există soluții pentru a permite upload-ul de SVG-uri în WordPress, dar această funcție nu este indicat să fie activată. SVG nu este obligatoriu un format de imagine, ci poate fi un format de document. Fișierele SVG permit încorporarea JavaScript, iar browser-ul poate să îl ruleze. Este așadar un risc de securitate upload-ul de fișiere SVG pe server. (Johansen, 2016)

4.2.42. Eliminarea HTML-ului personalizat

WordPress poate folosi HTML personalizat / nefiltrat / custom pentru diverse funcții, dar dacă acest lucru nu este absolut necesar, acesta se poate dezactiva, adăugând în fișierul wp-config.php: (9 Easy WordPress Security Tips: Hardening WordPress, 2011)

```
define( 'DISALLOW_UNFILTERED_HTML', true );
```

4.2.43. Evitarea ca site-ul să arate proaspăt lansat

Unul din lucrurile pentru a face ca un site să nu arate similar cu un site lansat recent este eliminarea postărilor și comentariilor implicite, cele cu care este furnizat inițial WordPress. (9 Easy WordPress Security Tips: Hardening WordPress, 2011)

De asemenea, este indicat să se elimine eticheta meta generator din template-ul (tema) site-ului. În fișierul header.php există această etichetă. În plus, se poate adăuga un filtru în functions.php. (9 Easy WordPress Security Tips: Hardening WordPress, 2011)

Este utilă și eliminrea din footer/subsol site a textului „Powered by WordPress” (realizat pe platforma WordPress), deoarece unii boți caută pe Internet în mod automat apariții ale acestei fraze și încearcă spargerea site-ului. (9 Easy WordPress Security Tips: Hardening WordPress, 2011)

4.2.44. Dezactivarea XML-RPC

XML-RPC permite unui site stabilirea de conexiuni cu aplicații mobile și plugin-uri WordPress, cum ar fi Jetpack. Pe de altă parte, este și un favorit al hackerilor WordPress, deoarece permite execuția unor comenzi simultan și poate da acces la un site. Dezactivarea poate fi făcută cu ajutorul unui plugin dedicat. (Stokes Barron, 2019)

4.2.45. Dezactivarea editării temei și pluginurilor prin tabloul de bord WordPress

Opțiunea de a edita fișierele temei și pluginurilor în tabloul de bord (dashboard-ul) WordPress este util pentru modificări rapide, dar este un risc de securitate. Se poate adăuga codul de mai jos în wp-config.php pentru dezactivare. (Stokes Barron, 2019)

```
// Oprește editarea fișierelor
define( 'DISALLOW_FILE_EDIT', true );
```
(Stokes Barron, 2019)

4.2.46. Modificarea cheilor de securitate WordPress

Cheile de securitate WordPress sunt responsabile de criptarea informațiilor stocate în cookie-urile utilizatorului. Se poate folosi un generator oferit chiar de WordPress pentru a le schimba. (Stokes Barron, 2019)

```
define('AUTH_KEY', 'put your unique phrase here');
define('SECURE_AUTH_KEY', 'put your unique phrase here');
define('LOGGED_IN_KEY', 'put your unique phrase here');
define('NONCE_KEY', 'put your unique phrase here');
define('AUTH_SALT', 'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT', 'put your unique phrase here');
define('NONCE_SALT', 'put your unique phrase here');
```

4.2.47. Dezactivarea raportării erorilor

Raportarea erorilor este utilă pentru a depista în detaliu o problemă. Pe de altă parte, în mod normal, când se raportează o eroare, se va afișa și calea serverului. Funcția se poate dezactiva adăugând codul de mai jos în wp-config.php. (Stokes Barron, 2019)

```
error_reporting(0);
@ini_set('display_errors', 0);
```

4.2.48. Deconectarea utilizatorilor inactivi

Există un plugin dedicat, Inactive Logout, care permite deconectarea utilizatorilor inactivi după o perioadă de inactivitate. Acest lucru este necesar, deoarece fără această funcție, după o logare tipică, sesiunea poate fi deturnată, iar hackerii pot infecta site-ul. (Stokes Barron, 2019)

4.2.49. Monitorizarea funcționare permanentă (uptime) site

Pentru monitorizarea funcționării permanentă a site-ului (uptime-ul), se poate folosi o soluție externă site-ului, al cărei rol este să verifice în mod automat, periodic, că anumite pagini din site se încarcă corect. Există soluții, unele gratuite, altele plătite, precum Uptime Robot, Pingdom, care fac această verificare automat. (How to Monitor Your WordPress Website Server Uptime (Easy Way), 2019)

Concluzii

Am căutat, în demersul nostru, să identificăm probleme pe partea de securitate, cu o comparație între platforma WordPress - versiunile 5.0 și 5.4.

Am făcut întâi de toate o analiză la nivel de server (am analizat serverul olivian.ro).

Ca observație generală, testele automate la nivel de server dau răspunsuri predefinite. Ce poate testa un instrument de verificare a securității unui server sunt întotdeauna o serie de lucruri standard, cu o tipologie clară. Din acest motiv am inclus și o parte în care am căutat soluții manuale pentru problemele automate generate în această parte.

Legat de observațiile punct-cu-punct, nu avem o sinteză generală a testelor rulate. Testele au fost foarte variate. Există următorul paradox - se rulează foarte multe teste, cu multe micro-observații, cu grad diferit de dificultate rezolvare și de importanță, și, la final, este necesară prezentarea unei concluzii generale. Da, se poate spune că serverul olivian.ro nu este suficient de bine securizat, pe de altă parte a rezistat online fără să fie compromis ani buni de zile. Notele generale în urma analizelor sunt undeva în gama "note medii", aceasta ar fi nota generală.

O observație ar fi că hostingul (găzduirea) pe care îl folosesc are soluții de blocare a atacurilor, în mod implicit. Am fost blocat (banned) de câteva ori, a fost nevoie de schimbarea IP-ului. Unele teste, din acest motiv, a trebuit să le reiau. Nu am anunțat hostingul despre testele pe care le rulez, pentru că e posibil să fi influențat rezultatele.

Pe de altă parte, ce am observat este un grad ridicat de repetitivitate a rezultatelor obținute în diferite testări. Am întâlnit o serie de tipare clare ale analizelor de securitate la nivel de server, în principiu există numeroase teste care copiază ceea ce deja fac alte teste.

Am întâlnit însă și lucruri unice, specifice unui test sau altuia.

Testele au diferit și prin forma prezentării, explicarea problemelor, detalierea problemelor în articole și resurse de pe site-ul propriu.

Ulterior, am făcut o analiză la nivel de instanță WordPress (am comparat https://olivian.ro/wp50/ și https://olivian.ro/wp54/). Așteptările noastre presupuneau că nu vom găsi nicio eroare nouă în WordPress 5.4, lansat doar cu puțin timp în urma rulării testelor, și cu aproape nicio vulnerabilitate general cunoscută la data testării. Cu toate acestea, pe anumite teste specifice, am găsit observații de securitate în plus la instanța de WordPress 5.4. O parte

din acestea se pot explica prin faptul că versiunea 5.4 vine cu un template (temă) implicit mult diferit față de 5.0, deși ambele fac parte din ramura WordPress 5. Acest nou template apelează alte fișiere CSS / JavaScript. Totuși, a fost o surpriză să vedem în unele teste pe 5.4 rezultate mai slabe decât pe 5.0. Desigur, cum am spus, asta s-a întâmplat ca excepție, nu ca regulă.

În partea finală, am căutat soluții din literatura de specialitate (articole din bloguri online, din studii științifice, din cărți) pentru potențialele probleme ale platformei WordPress. Ne așteptam să găsim câteva puncte, dar rezultatele obținute au arătat că sunt numeroase lucruri care se pot face pentru a îmbunătăți securitatea pe platforma WordPress. Unele din observații au prioritate redusă, altele au grad de dificultate implementare ridicat, altele pur și simplu nu se aplică tuturor instanțelor de WordPress, dar, în ansamblu, au fost numeroase puncte ce pot fi un punct bun de pornirea în securizarea unei instanțe de WordPress.

Una din concluzii este că literatura în domeniul securității WordPress este foarte bogată. Există numeroase resurse care au abordat această temă, de la bloguri la articole științifice și cărți.

Printre concluzii se poate număra faptul că, în opinia noastră, WordPress nu își propune să ofere, în mod implicit, o securitate ridicată, în mod forțat, la instalare. Câteva exemple în acest sens - unul din pluginurile standard în orice instalare de WordPress este "Akismet anti-spam", care poate ajuta la reducerea comentariilor lăsate în mod automat de boți. Dar acest plugin nu este activat în mod implicit, ci utilizatorul trebuie să îl activeze, dacă i se pare util acest lucru. Apoi, parolele utilizatorilor sunt sugerate de WordPress, și, în general, sunt complexe și sigure. Dar dacă un utilizator dorește neapărat să pună o parolă considerată slabă, WordPress arată o avertizare, obligă vizitatorul să bifeze că a înțeles riscul, dar parola poate fi setată fără a fi în mod obligatoriu una complicată. Apoi, unele din funcțiile des întâlnite în pluginuri de securitate ar putea fi integrate în mod implicit în WordPress. În opinia noastră, WordPress își propune însă altceva - să aibă un cod simplu, robust, pe care să se poată pune diferite straturi (teme, pluginuri, customizări (particularizări) diferite la nivel de server / baze de date / fișiere PHP). Dacă WordPress ar fi mai complex, ar exista un risc să fie mai greu de întreținut. Oferind doar un schelet, pe care utilizatorii pot adăuga ce doresc, WordPress oferă, de fapt, mai multă libertate de particularizare a instalării, chiar dacă odată cu asta vin și riscuri inerente.

O altă concluzie, în urma folosirii îndelungate a platformei, ca utilizatori, dar și ca administratori, este că WordPress a făcut periodic îmbunătățiri constante, pe toate planurile importante (design, uzabilitate, funcționalități, securitate). Dacă se compară versiuni majore, treceri de la WordPress 2 la 3, de la 3 la 4 sau de la 4 la 5, se observă în general multe schimbări,

cu impact puternic pe multe planuri. Scriem acest lucru pentru că se întâmplă uneori ca cineva să facă o actualizare între versiuni minore de WordPress și să considere fie că nu s-a schimbat nimic, fie că au fost doar mici schimbări. Ei bine, într-o privire de ansamblu se pot observa și lucruri importante schimbate. Dacă am fi analizat WordPress 3.0 sau 4.0 (cum intenționasem inițial), așteptarea noastră e că am fi găsit numeroase probleme. Pe de altă parte, asta ar fi făcut lucrarea neactuală, deoarece instalările noi de WordPress se fac de obicei, probabil, pe variante recente, nu unele mai vechi.

De asemenea, WordPress, în mod implicit asigură o securitate de bază, care pentru mulți utilizatori poate fi suficientă. Ajută, desigur, ca un utilizator să fie cât mai bine documentat despre ce presupune securitatea în WordPress, dar și cu setările implicite, o instanță de WordPress poate fi considerată sigură. Exemple de lucruri implicite, care ajută la securitate - generare automată parole sigure, teme (template-uri) standard bine securizate și cu aspect care îndeamnă la folosirea lor, plugin de filtrare comentarii spam (Akismet anti-spam) care trebuie doar activat, este pre-instalat, roluri granulare utilizatori, funcție de activare politică confidențialitate ante-scrisă, ce trebuie doar customizată, funcție de permite comentarii care sunt destul de bine securizate, chiar din varianta standard. Poate cel mai important element al WordPress este, însă, că este bazat pe cod sursă deschis (open source) și un utilizator poate găsi ușor soluții gratuite pentru numeroase probleme, inclusiv cele legate de securitate.

Bibliografie

2012. *Ghid Pentru Securizarea Aplicațiilor Și Serviciilor Web. Versiunea 1.0 – 24 Februarie 2012*. 1st ed. [ebook] Centrul Național de Răspuns la Incidente de Securitate Cibernetică. Disponibil la: <https://www.cert.ro/vezi/document/ghid-securizare-aplicatii-web> [Accesat 12 mai 2020].

2015. *Wordpress: Learn Wordpress In A DAY! - The Ultimate Crash Course To Learning The Basics Of Wordpress In No Time (Wordpress, Wordpress Course, Wordpress ... Wordpress Books, Wordpress For Beginners) Kindle Edition*. 15th ed.

2017. *BDM'S: The Wordpress Guidebook*. Papercut Limited.

2020. *Scan Remediation Report*. United Kingdom: Burp Suite Enterprise Edition.

Agrawal, H., 2019. *Wordpress Security Guide: 14 Pro Tips To Secure A Wordpress Website*. [online] ShoutMeLoud. Disponibil la: <https://www.shoutmeloud.com/wordpress-security.html> [Accesat 14 mai 2020].

Attard, D., 2020. *7 Ways To Fix Wordpress Hacked Sites + 17 Security Steps To Protect*. [online] CollectiveRay. Disponibil la: <https://www.collectiveray.com/wordpress-hacked> [Accesat 14 mai 2020].

Banu, S., 2020. *7 Wordpress Security Vulnerabilities & How To Fix Them - Malcare*. [online] MalCare. Disponibil la: <https://www.malcare.com/blog/wordpress-vulnerabilities/> [Accesat 13 mai 2020].

Bari, S., fără dată. *Ultimate Wordpress Security Tips For Woocommerce | Basic To Advanced Guide*. [online] WPManageNinja. Disponibil la: <https://wpmanageninja.com/ultimate-wordpress-security-tips-for-woocommerce/> [Accesat 14 mai 2020].

Brazell, A., 2010. *Wordpress Bible*. Indianapolis, Ind.: Wiley.

Canavan, T., 2011. *CMS Security Handbook*. Indianapolis, IN: Wiley Pub.

Cernica, I., Popescu, N. și Tiganoaia, B., 2019. Security Evaluation of Wordpress Backup Plugins. *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, [online] Disponibil la: <https://ieeexplore.ieee.org/document/8744951> [Accesat 12 mai 2020].

Chahal, P., 2020. *Hardening Wordpress Security: For Beginners To Advanced*. [online] TemplateToaster Blog. Disponibil la: <https://blog.templatetoaster.com/hardening-wordpress-security/> [Accesat 14 mai 2020].

Clarke, J., 2012. *SQL Injection Attacks And Defense $C*. Sin Lugar: Syngress.

Cloud.appscan.com. fără dată. *HCL Appscan On Cloud*. [online] Disponibil la: <https://cloud.appscan.com/AsoCUI/serviceui/main/myapps/oneapp/c0033d45-d792-41aa-932a-46965ab1b665/issues/issue/advisory> [Accesat 17 mai 2020].

Cloud.appscan.com. fără dată. *HCL Appscan On Cloud*. [online] Disponibil la: <https://cloud.appscan.com/AsoCUI/serviceui/main/myapps/oneapp/cb654b66-b9c6-4345-b1ce-73e8c46a7812/scans> [Accesat 17 mai 2020].

Cloudflare. fără dată. *What Is Transport Layer Security (TLS)?*. [online] Disponibil la: <https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/> [Accesat 8 mai 2020].

Connelly, O., 2011. *Wordpress 3 Ultimate Security*. Birmingham, U.K.: Packt Open Source.

Cve.mitre.org. fără dată. *CVE -Common Vulnerabilities And Exposures (CVE)*. [online] Disponibil la: <https://cve.mitre.org/> [Accesat 12 mai 2020].

Cvedetails.com. fără dată. *Wordpress: CVE Security Vulnerabilities, Versions And Detailed Reports*. [online] Disponibil la: <https://www.cvedetails.com/product/4096/Wordpress-Wordpress.html?vendor_id=2337> [Accesat 12 mai 2020].

Detectify.com. fără dată. *Monitor Your Site's Security | Detectify*. [online] Disponibil la: <https://detectify.com/report/962a89862e75f6430a49a046afa45183/ccf634358bbfa4b27408f5f291a6fbb28fcaeb23/findinglist> [Accesat 15 mai 2020].

Douglas, N., 2019. *'Iloveyou' And The 24 Other Worst Passwords Of 2019*. [online] Lifehacker. Disponibil la: <https://lifehacker.com/iloveyou-and-the-24-other-worst-passwords-of-2019-1840491292> [Accesat 13 mai 2020].

Dunn, I., 2018. *Wordpress 5.0.1 Security Release*. [online] WordPress News. Disponibil la: <https://wordpress.org/news/2018/12/wordpress-5-0-1-security-release/> [Accesat 13 mai 2020].

Eu.appsec.insight.rapid7.com. fără dată. *Rapid7*. [online] Disponibil la: <https://eu.appsec.insight.rapid7.com/op/5D56656D3101C5FA9CB8/#/apps/82fc5553-49fd-45be-8b48-065034c5afac/configuration/de3e1d8a-f3b7-4ac1-bfc1-50968fde1711/scan/82f5cafd-d2de-4f07-855a-dc91a70e47a8> [Accesat 17 mai 2020].

Eu.appsec.insight.rapid7.com. fără dată. *Rapid7*. [online] Disponibil la: <https://eu.appsec.insight.rapid7.com/op/5D56656D3101C5FA9CB8/#/apps/2657f952-a3ac-49e2-8e43-1a018cd369e7/configuration/4ea91a95-0270-488d-83db-d48fc6190952/scan/a14ae53a-00d1-4f27-8794-961ba22773aa> [Accesat 17 mai 2020].

Guruli, N., 2017. *Wordpress Security Made Easy. Visual Step-By-Step Guide From Zero To Hero How To Install Secure Wordpress Site And Maintain It Cost Free And Without Turning Into A Geek*. Pasadena: CreateSpace Independent Publishing Platform; 1 edition (June 13, 2017).

HackerTarget.com. fără dată. *28 Online Vulnerability Scanners & Network Tools | Hackertarget.Com*. [online] Disponibil la: <https://hackertarget.com/> [Accesat 17 mai 2020].

HackerTarget.com. fără dată. *Wordpress Security Scan | Hackertarget.Com*. [online] Disponibil la: <https://hackertarget.com/wordpress-security-scan/> [Accesat 17 mai 2020].

Helme, S., fără dată. *Scan Results For Olivian.Ro*. [online] Securityheaders.com. Disponibil la: <https://securityheaders.com/?followRedirects=on&hide=on&q=olivian.ro> [Accesat 8 mai 2020].

https://olivian.ro/info.php. fără dată. *Phpinfo ()*. [online] Disponibil la: <https://olivian.ro/info.php> [Accesat 8 mai 2020].

https://portal.apptrana.com/. fără dată. *Apptrana*. [online] Disponibil la: <https://portal.apptrana.com/dashboard> [Accesat 8 mai 2020].

Immuniweb.com. fără dată. *Website Security Test Of Olivian.Ro (89.33.25.24)*. [online] Disponibil la: <https://www.immuniweb.com/websec/?id=YnywPnwQ> [Accesat 17 mai 2020].

Immuniweb.com. fără dată. *Website Security Test Of Olivian.Ro (89.33.25.24)*. [online] Disponibil la: <https://www.immuniweb.com/websec/?id=1NVsfnrG> [Accesat 17 mai 2020].

Infosec Resources. 2011. *9 Easy Wordpress Security Tips: Hardening Wordpress*. [online] Disponibil la: <https://resources.infosecinstitute.com/hardening-wordpress/> [Accesat 14 mai 2020].

Johansen, B., 2016. *SVG Uploads In Wordpress (The Inconvenient Truth)*. [online] {Bjørn:Johansen}. Disponibil la: <https://www.bjornjohansen.com/svg-in-wordpress> [Accesat 14 mai 2020].

Kelsey, T., 2012. *Getting Started With Wordpress*. Boston, MA: Course Technology.

Król, K. și Silver, A., 2013. *Wordpress 3.7 Complete*. Birmingham: Packt Publishing Ltd.

Król, K., 2019. *Wordpress 5 Complete*. Packt Publishing.

Lefebvre, Y., 2012. *Wordpress Plugin Development Cookbook*. Birmingham, U.K.: Packt Pub.

MacDonald, M., 2014. *Wordpress: The Missing Manual*. 1st ed. Sebastopol: O'Reilly Media, Inc.

Messenlehner, B., Coleman, J., Williams, B., Shelby, N., Comer, R. și Roumeliotis, C., 2014. *Building Web Apps With Wordpress*. Sebastopol: O'Reilly Media, Inc.

fără dată. *Ghid Întrebări Și Răspunsuri Cu Privire La Aplicarea Regulamentului (UE) 2016/679*. [ebook] Autoritatea Națională de Supraveghere a Prelucrării Datelor cu Caracter Personal. Disponibil la: <https://www.dataprotection.ro/servlet/ViewDocument?id=1650> [Accesat 12 mai 2020].

fără dată. *N-Stalker Free Edition Version X*. Brazilia: ORMELLE PARTICIPACOES LTDA.

fără dată. *OWASP ZAP*. OWASP.

fără dată. *Probely*. [online] Disponibil la: <https://app.probely.com/portal/2N9Z7NHvQq3e/findings/1/filter?state=notfixed> [Accesat 17 mai 2020].

fără dată. *Probely*. [online] Disponibil la: <https://app.probely.com/portal/2HCuuCbULi6c/findings/1/filter?state=notfixed> [Accesat 17 mai 2020].

Observatory.mozilla.org. 2020. *Mozilla Observatory :: Scan Results For Olivian.Ro*. [online] Disponibil la: <https://observatory.mozilla.org/analyze/olivian.ro> [Accesat 8 mai 2020].

Onishi, A., 2013. *Pro Wordpress Theme Development*. New York: Apress L.P.

Owasp.org. fără dată. *Command Injection | OWASP*. [online] Disponibil la: <https://owasp.org/www-community/attacks/Command_Injection> [Accesat 17 mai 2020].

Owasp.org. fără dată. *OWASP Top Ten Web Application Security Risks | OWASP*. [online] Disponibil la: <https://owasp.org/www-project-top-ten/> [Accesat 12 mai 2020].

Pentest-Tools.com. fără dată. *Website Vulnerability Scanner - Online Scan For Web Vulnerabilities | Pentest-Tools.Com*. [online] Disponibil la: <https://pentest-tools.com/website-vulnerability-scanning/website-scanner> [Accesat 26 April 2020].

Peyrott, S., 2016. *4 Types Of Memory Leaks In Javascript And How To Get Rid Of Them*. [online] Auth0 - Blog. Disponibil la: <https://auth0.com/blog/four-types-of-leaks-in-your-javascript-code-and-how-to-get-rid-of-them/> [Accesat 17 mai 2020].

Qualysguard.qg2.apps.qualys.eu. fără dată. *Qualys Security And Compliance Suite Login*. [online] Disponibil la: <https://qualysguard.qg2.apps.qualys.eu/> [Accesat 17 mai 2020].

Ratnayake, R., 2013. *Wordpress Web Application Development*. Birmingham: Packt Publishing.

Red Hat Customer Portal. fără dată. *4.13. Hardening TLS Configuration Red Hat Enterprise Linux 7 | Red Hat Customer Portal*. [online] Disponibil la: <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-hardening_tls_configuration> [Accesat 17 mai 2020].

Rehman, I., 2020. *Top 6 Most Common Wordpress Vulnerabilities (With Fixes)*. [online] Website Hosting Rating. Disponibil la: <https://www.websitehostingrating.com/most-common-wordpress-vulnerabilities/> [Accesat 14 mai 2020].

Ristic, I., 2020. *CAA Mandated By CA/Browser Forum | Qualys Blog*. [online] Qualys Blog. Disponibil la: <https://blog.qualys.com/ssllabs/2017/03/13/caa-mandated-by-cabrowser-forum> [Accesat 8 mai 2020].

Sabin-Wilson, L., fără dată. *Wordpress All-In-One For Dummies*. 3rd ed. New Jersey: John Wiley & Sons, Inc.

Securityfocus.com. fără dată. *CMME Cross Site Scripting And Information Disclosure Vulnerabilities*. [online] Disponibil la: <https://www.securityfocus.com/bid/30239/discuss> [Accesat 17 mai 2020].

Siarto, J., 2010. *Head First Wordpress*. Sebastopol, Calif.: O'Reilly.

Siteguarding.com. fără dată. *Website Security | Website Antivirus | Website Firewall | Website File Monitoring | Website Backup | Malware, Virus, Trojan Removal | Blacklist Removal | Siteguarding*. [online] Disponibil la: <https://www.siteguarding.com/> [Accesat 8 mai 2020].

Smith, B. și McCallister, M., 2010. *Wordpress In Depth*. Indianapolis, Ind: Que.

Ssltrust.com.au. fără dată. *Free Website Safety & Security Check*. [online] Disponibil la: <https://www.ssltrust.com.au/ssl-tools/website-security-check?domain=olivian.ro> [Accesat 8 mai 2020].

Stokes Barron, B., 2019. *22 Steps To Harden Your Wordpress Website Security | Websitesetup.Org*. [online] WebsiteSetup.org. Disponibil la: <https://websitesetup.org/wordpress-security/> [Accesat 16 mai 2020].

Sucuri Security. fără dată. *Https://Olivian.Ro/Wp50/*. [online] Disponibil la: <https://sitecheck.sucuri.net/results/https/olivian.ro/wp50/> [Accesat 17 mai 2020].

Sucuri Security. fără dată. *Https://Olivian.Ro/Wp54/*. [online] Disponibil la: <https://sitecheck.sucuri.net/results/https/olivian.ro/wp50/> [Accesat 17 mai 2020].

Sucuri. fără dată. *Sucuri - Website Threat Report 2019*. [online] Disponibil la: <https://sucuri.net/reports/2019-hacked-website-report/> [Accesat 13 mai 2020].

Trunde, H. și Weippl, E., 2015. WordPress Security: An analysis based on publicly available exploits. *iiWAS '15: Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, [online] Disponibil la: <http://dx.doi.org/10.1145/2837185.2837195> [Accesat 12 mai 2020].

Upguard.com. fără dată. *Free Website Security Scan | Upguard*. [online] Disponibil la: <https://www.upguard.com/webscan?c=https%3A%2F%2Folivian.ro%2Fwp50%2F> [Accesat 17 mai 2020].

Upguard.com. fără dată. *Free Website Security Scan | Upguard*. [online] Disponibil la: <https://www.upguard.com/webscan?c=https%3A%2F%2Folivian.ro%2Fwp54%2F> [Accesat 17 mai 2020].

Valk, J., Berg, A., Heijmans, M., Rakt, M. și Valk, T., fără dată. *Optimize Your Wordpress Site*.

W3techs.com. 2020. *Usage Statistics And Market Share Of Wordpress, mai 2020*. [online] Disponibil la: <https://w3techs.com/technologies/details/cm-wordpress> [Accesat 12 mai 2020].

W3techs.com. fără dată. *Usage Statistics And Market Share Of Apache, mai 2020*. [online] Disponibil la: <https://w3techs.com/technologies/details/ws-apache> [Accesat 8 mai 2020].

webcookies.org. fără dată. *Olivian.Ro | Privacy & Security Report #30424440*. [online] Disponibil la: <https://webcookies.org/cookies/olivian.ro/30424440?158074> [Accesat 17 mai 2020].

webcookies.org. fără dată. *Olivian.Ro | Privacy & Security Report #30424448*. [online] Disponibil la: <https://webcookies.org/cookies/olivian.ro/30424448?566932> [Accesat 17 mai 2020].

Williams, A., 2019. *Wordpress For Beginners 2019. A Visual Step-By-Step Guide To Mastering Wordpress*.

WP Hacked Help Blog. 2019. *Common Wordpress Security Vulnerabilities 2020 With Fixes*. [online] Disponibil la: <https://secure.wphackedhelp.com/blog/wordpress-vulnerabilities-how-to-fix-guide-tools/> [Accesat 14 mai 2020].

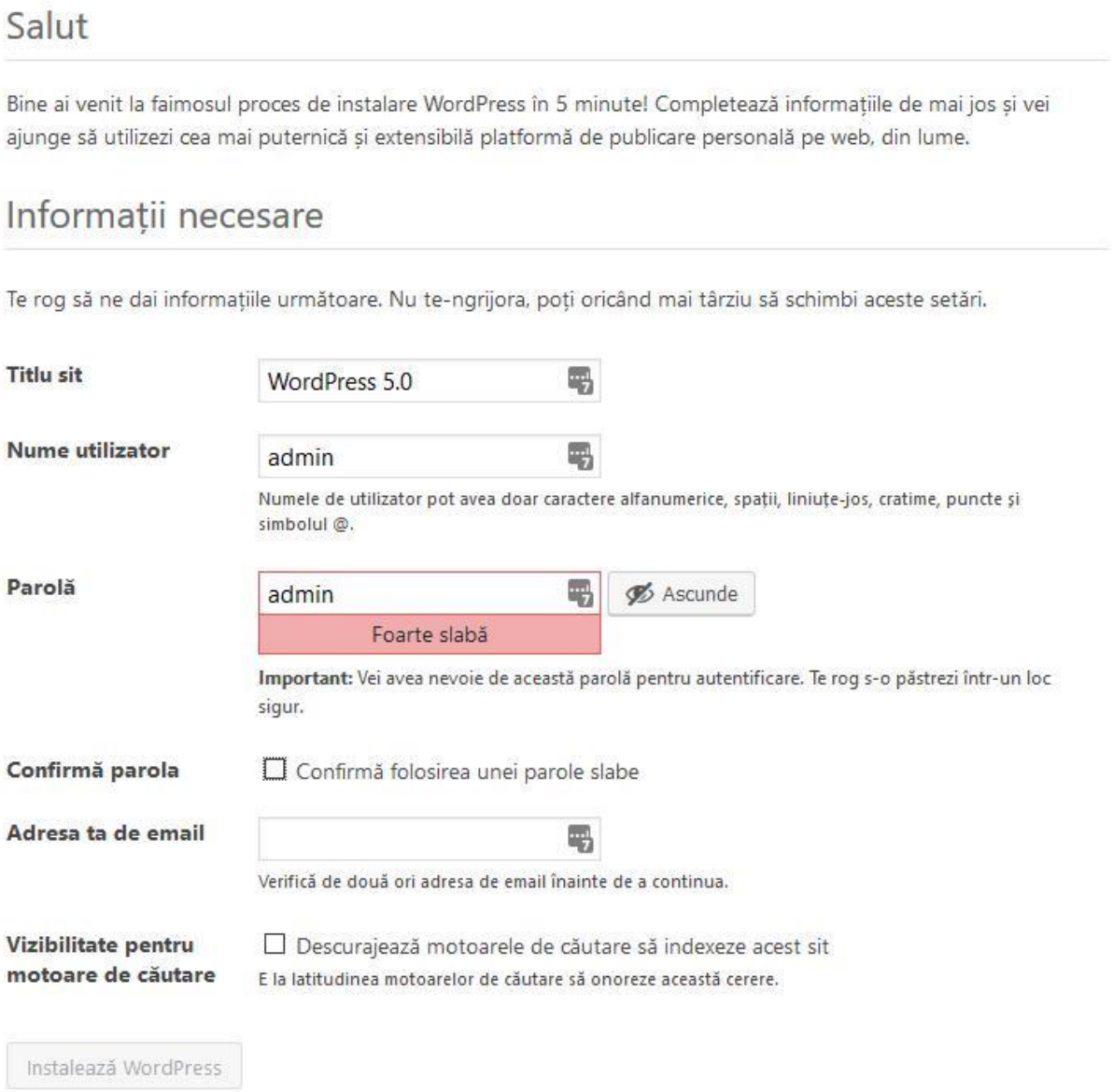
WPBeginner. 2019. *How To Monitor Your Wordpress Website Server Uptime (Easy Way)*. [online] Disponibil la: <https://www.wpbeginner.com/plugins/how-to-monitor-server-uptime-in-wordpress/> [Accesat 16 mai 2020].

Wpbeginner.com. 2015. *How To Disable Automatic Updates In Wordpress*. [online] Disponibil la: <https://www.wpbeginner.com/wp-tutorials/how-to-disable-automatic-updates-in-wordpress/> [Accesat 13 mai 2020].

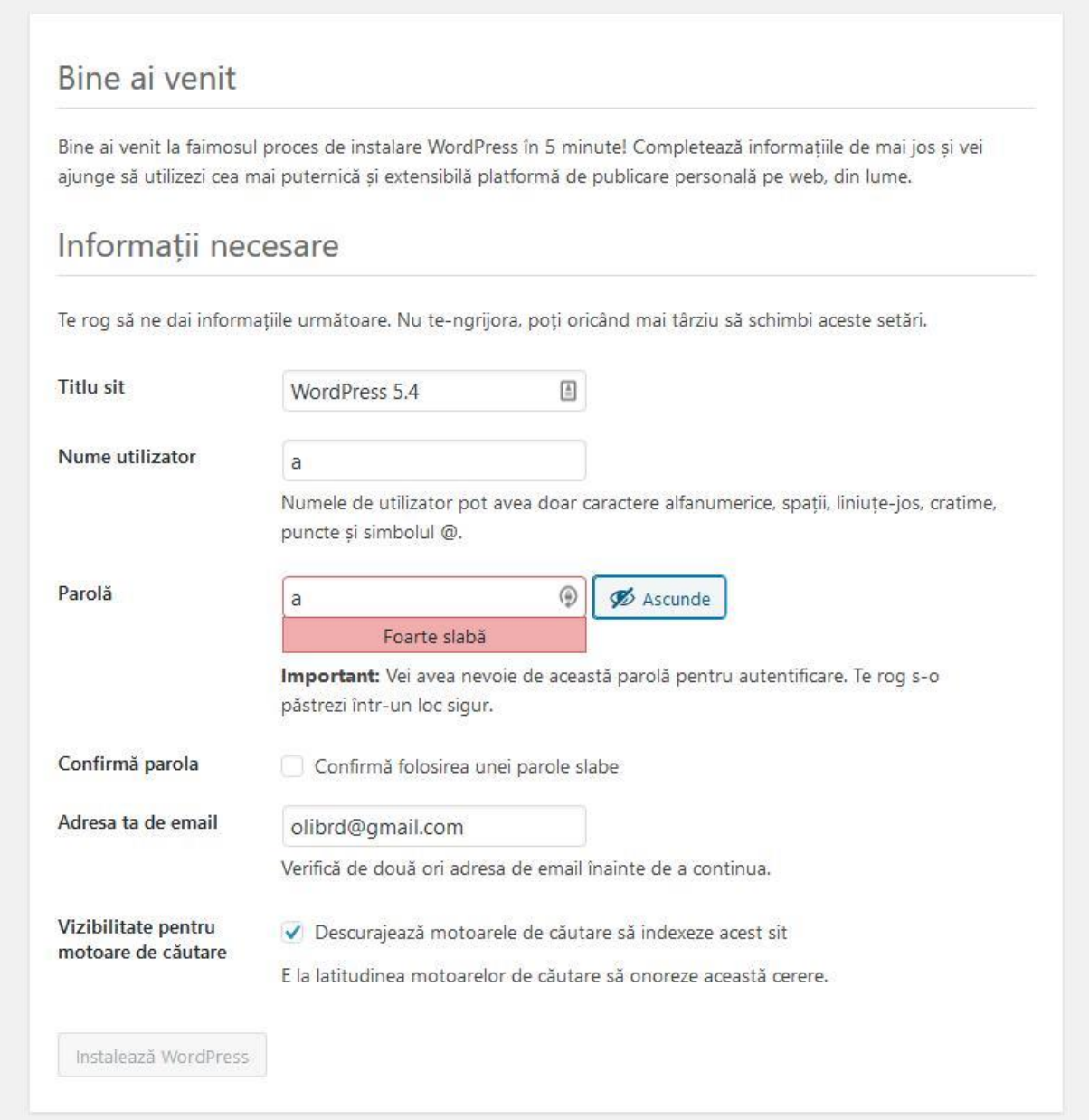
Anexe

1. Anexe instalare WordPress

1.1. Securitate slabă instalare WordPess 5.0

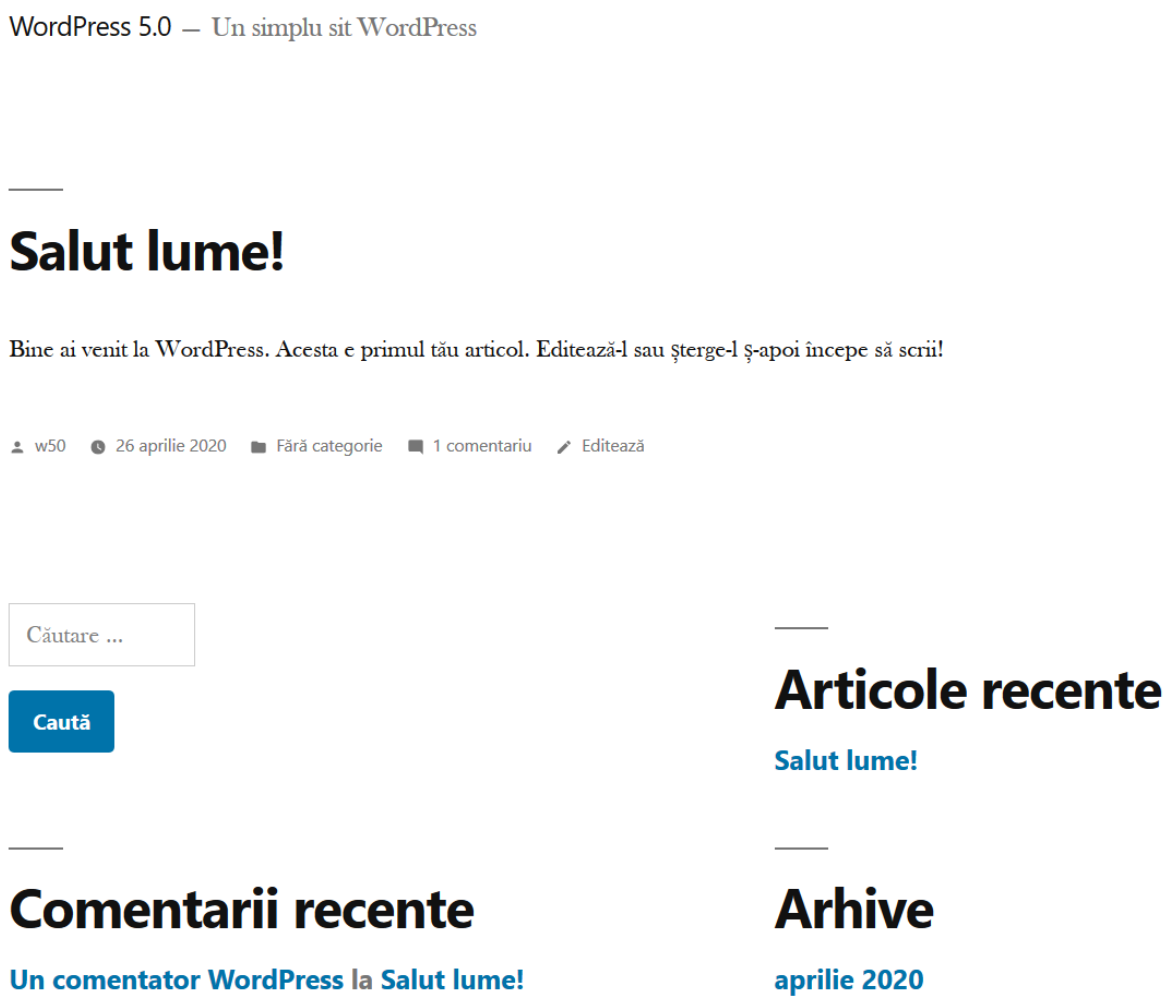


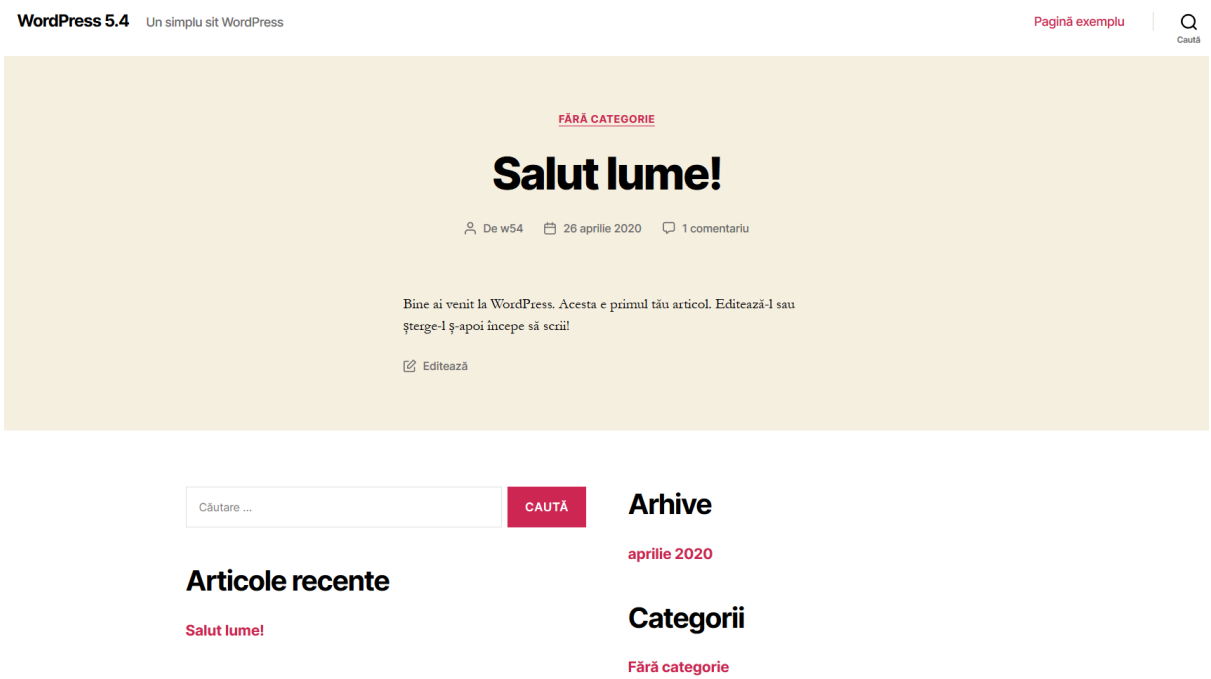
1.2. Securitate slabă instalare WordPess 5.4



2. Anexe Capitolul 2. - Testare WordPress la nivel de server

2.1. Cum arată site-urile demonstrative





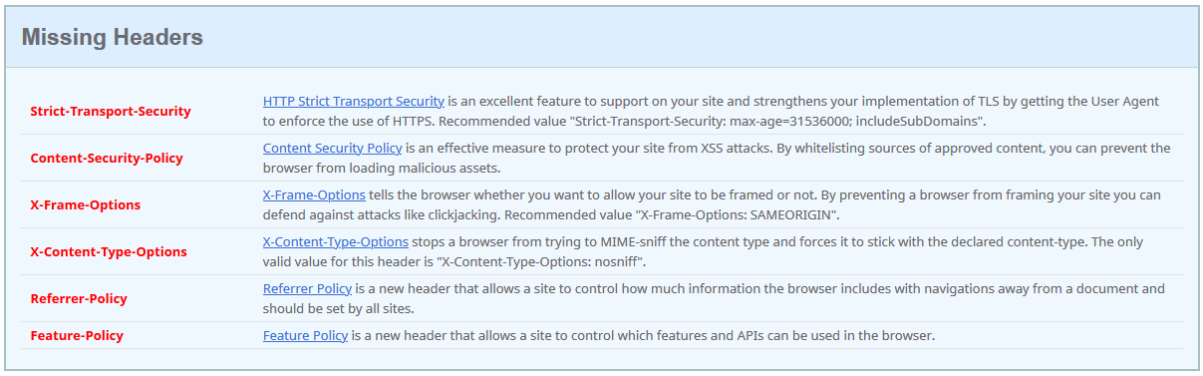
2.2. Testare observatory.mozilla.org

Test Scores

Test	Pass	Score	Reason	Info
Content Security Policy	✗	-25	Content Security Policy (CSP) header not implemented	
Cookies	–	0	No cookies detected	
Cross-origin Resource Sharing	✓	0	Content is not visible via cross-origin resource sharing (CORS) files or headers	
HTTP Public Key Pinning	–	0	HTTP Public Key Pinning (HPKP) header not implemented (optional)	
HTTP Strict Transport Security	✗	-20	HTTP Strict Transport Security (HSTS) header not implemented	
Redirection	✓	0	Initial redirection is to HTTPS on same host, final destination is HTTPS	
Referrer Policy	–	0	Referrer-Policy header not implemented (optional)	
Subresource Integrity	–	0	Subresource Integrity (SRI) not implemented, but all scripts are loaded from a similar origin	
X-Content-Type-Options	✗	-5	X-Content-Type-Options header not implemented	
X-Frame-Options	✗	-20	X-Frame-Options (XFO) header not implemented	
X-XSS-Protection	✗	-10	X-XSS-Protection header not implemented	

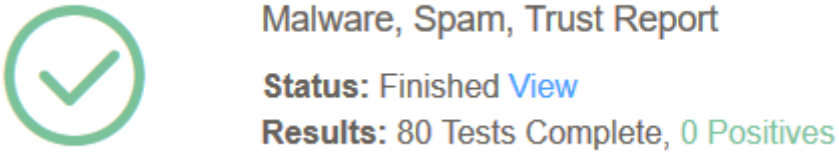
(Mozilla Observatory :: Scan Results for olivian.ro, 2020)

2.3. Testare securityheaders.com

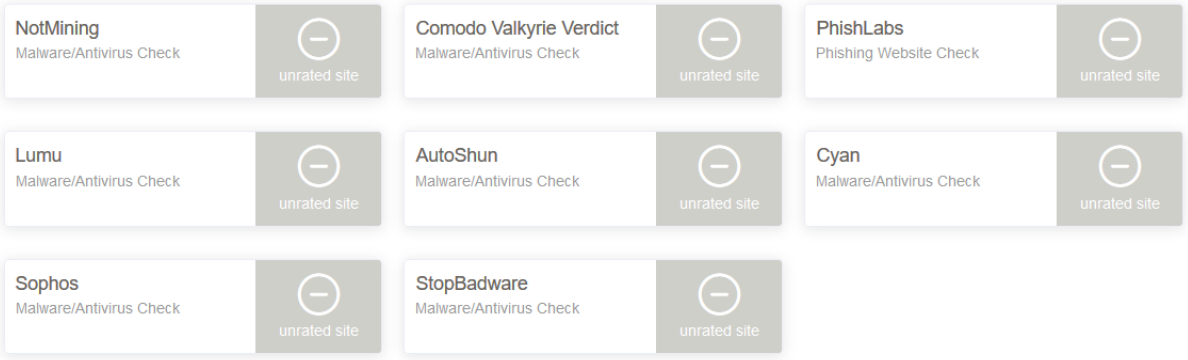


(Helme, fără dată)

2.4. Testare ssltrust.com.au

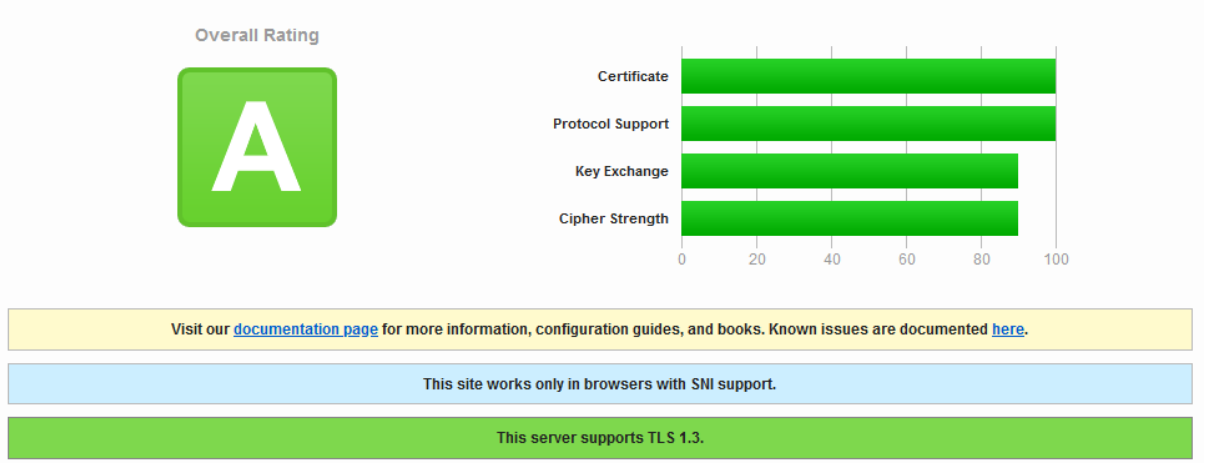


(Free Website Safety & Security Check, fără dată)



(Free Website Safety & Security Check, fără dată)

2.5. Testare ssllabs.com



(Ristic, 2020)

2.6. Testare siteguarding.com

Link Analyze

Your website loads images, javascript, css style files from these domains.

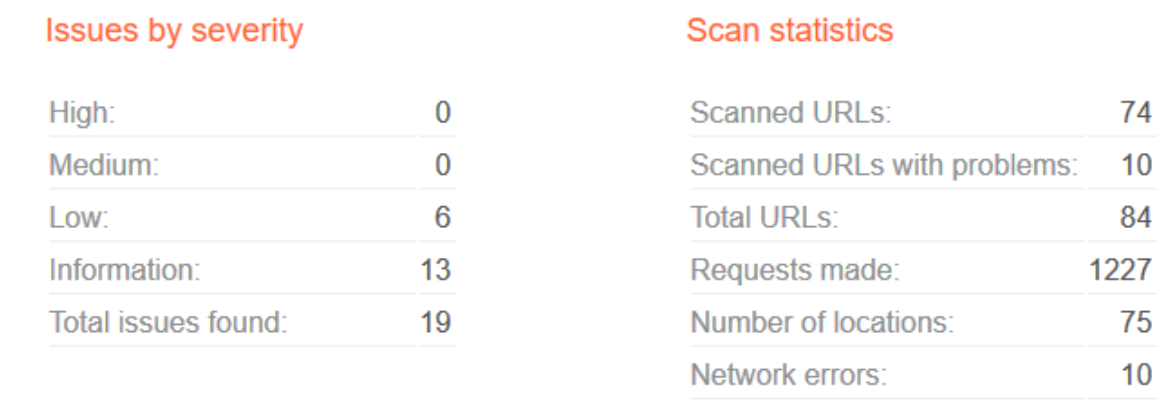
Total Domains: **36**

Total Blacklisted Domains: **0**

Domain URL	Found Links	Blacklist Status
olivian.ro	194	ok

(Website Security | Website Antivirus | Website Firewall | Website File Monitoring | Website Backup | Malware, Virus, Trojan Removal | Blacklist Removal | SiteGuarding, fără dată)

2.7 Testare portswigger.net



Issues by severity

High:	0
Medium:	0
Low:	6
Information:	13
Total issues found:	19

Scan statistics

Scanned URLs:	74
Scanned URLs with problems:	10
Total URLs:	84
Requests made:	1227
Number of locations:	75
Network errors:	10

(Scan Remediation Report, 2020)

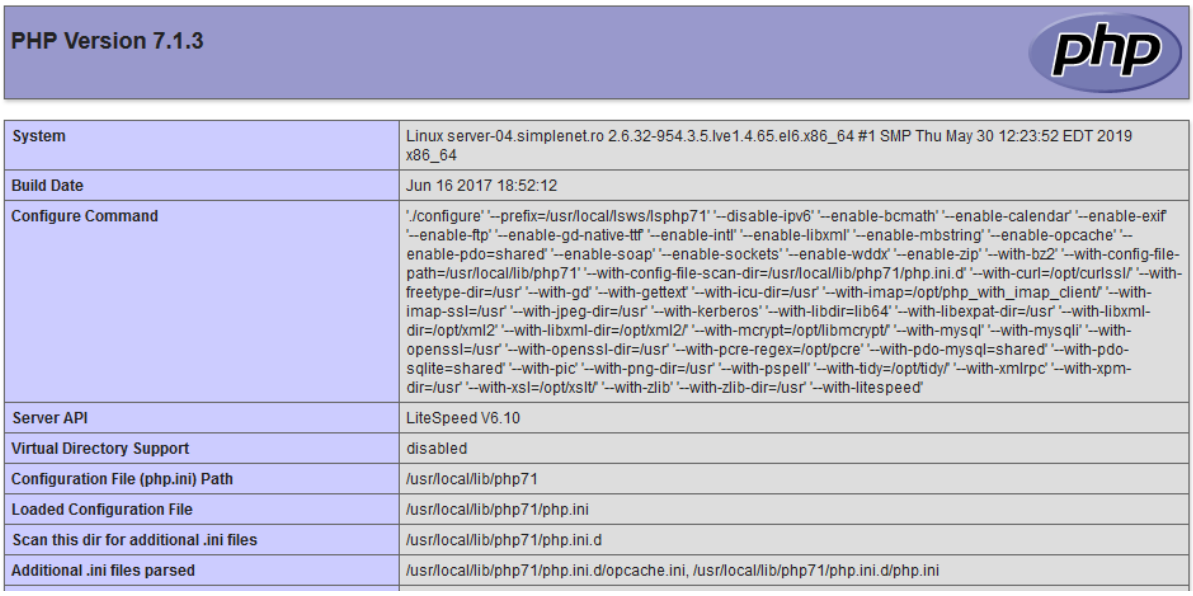
Issues found on https://olivian.ro

URLs By issue type	Severity	Confidence	More detail
Strict transport security not enforced [1]			
/	Low	Certain	>>
Password field with autocomplete enabled [5]			
/wp50/wp-admin/	Low	Certain	>>
/wp50/wp-includes/css/dashicons.min.css	Low	Certain	>>
/wp50/wp-login.php	Low	Certain	>>
/wp50/wp-login.php	Low	Certain	>>
/wp54/wp-login.php	Low	Certain	>>
Cross-domain Referer leakage [7]			
/	Info	Certain	>>
/	Info	Certain	>>
/wp50/	Info	Certain	>>
/wp50/index.php/2020/04/26/salut-lume/	Info	Certain	>>
/wp54/	Info	Certain	>>
/wp54/index.php/2020/04/26/salut-lume/	Info	Certain	>>
/wp54/wp-comments-post.php	Info	Certain	>>
Cookie without HttpOnly flag set [2]			
/	Info	Certain	>>
/wp54/wp-comments-post.php	Info	Certain	>>
Cacheable HTTPS response [1]			
/	Info	Certain	>>
HTML does not specify charset [2]			
/wp50/xmlrpc.php	Info	Certain	>>
/wp54/xmlrpc.php	Info	Certain	>>
Frameable response (potential Clickjacking) [1]			
/	Info	Firm	>>

(Scan Remediation Report, 2020)

2.8. Testare apptrana.com

(AppTrana, fără dată)



PHP Version 7.1.3

System	Linux server-04.simplenet.ro 2.6.32-954.3.5.lve1.4.65.el6.x86_64 #1 SMP Thu May 30 12:23:52 EDT 2019 x86_64
Build Date	Jun 16 2017 18:52:12
Configure Command	'./configure' '--prefix=/usr/local/lsws/lsphp71' '--disable-ipv6' '--enable-bcmath' '--enable-calendar' '--enable-exif' '--enable-ftp' '--enable-gd-native-ttf' '--enable-intl' '--enable-libxml' '--enable-mbstring' '--enable-opcache' '--enable-pdo=shared' '--enable-soap' '--enable-sockets' '--enable-wddx' '--enable-zip' '--with-bz2' '--with-config-file-path=/usr/local/lib/php71' '--with-config-file-scan-dir=/usr/local/lib/php71/php.ini.d' '--with-curl=/opt/curlssl/' '--with-freetype-dir=/usr' '--with-gd' '--with-gettext' '--with-icu-dir=/usr' '--with-imap=/opt/php_with_imap_client/' '--with-imap-ssl=/usr' '--with-jpeg-dir=/usr' '--with-kerberos' '--with-libdir=lib64' '--with-libexpat-dir=/usr' '--with-libxml-dir=/opt/xml2' '--with-libxml-dir=/opt/xml2/' '--with-mcrypt=/opt/libmcrypt/' '--with-mysql' '--with-mysqli' '--with-openssl=/usr' '--with-openssl-dir=/usr' '--with-pcre-regex=/opt/pcre' '--with-pdo-mysql=shared' '--with-pdo-sqlite=shared' '--with-pic' '--with-png-dir=/usr' '--with-pspell' '--with-tidy=/opt/tidy/' '--with-xmlrpc' '--with-xpm-dir=/usr' '--with-xsl=/opt/xslt/' '--with-zlib' '--with-zlib-dir=/usr' '--with-litespeed'
Server API	LiteSpeed V6.10
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/lib/php71
Loaded Configuration File	/usr/local/lib/php71/php.ini
Scan this dir for additional .ini files	/usr/local/lib/php71/php.ini.d
Additional .ini files parsed	/usr/local/lib/php71/php.ini.d/opcache.ini, /usr/local/lib/php71/php.ini.d/php.ini

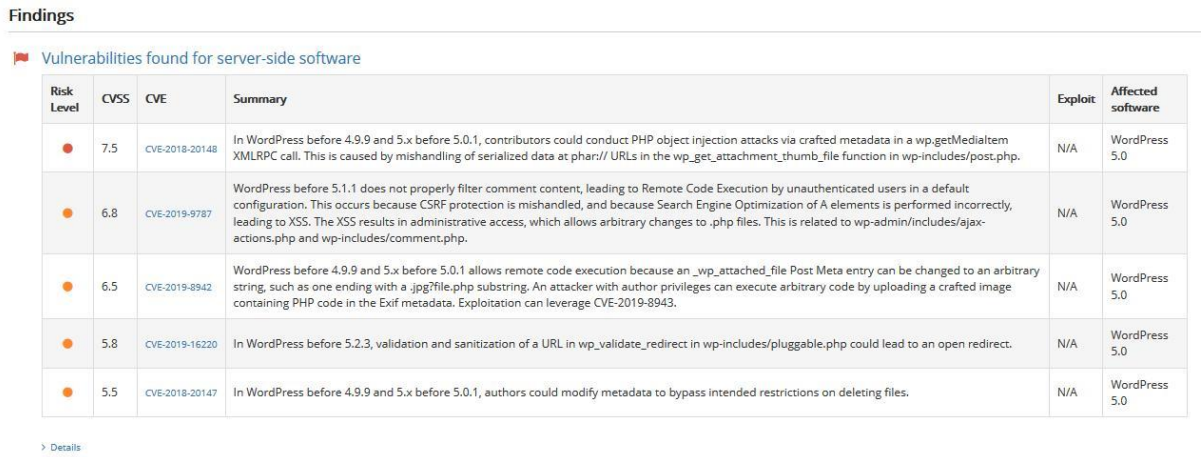
(phpinfo (), fără dată)

2.9. Testare detectify.com

(Monitor Your Site's Security | Detectify, fără dată)

3. Anexe Capitolul 3. Testare WordPress la nivel de versiune diferită (5.0 vs. 5.4, testat comparativ)

3.1. Testare pentest-tools.com



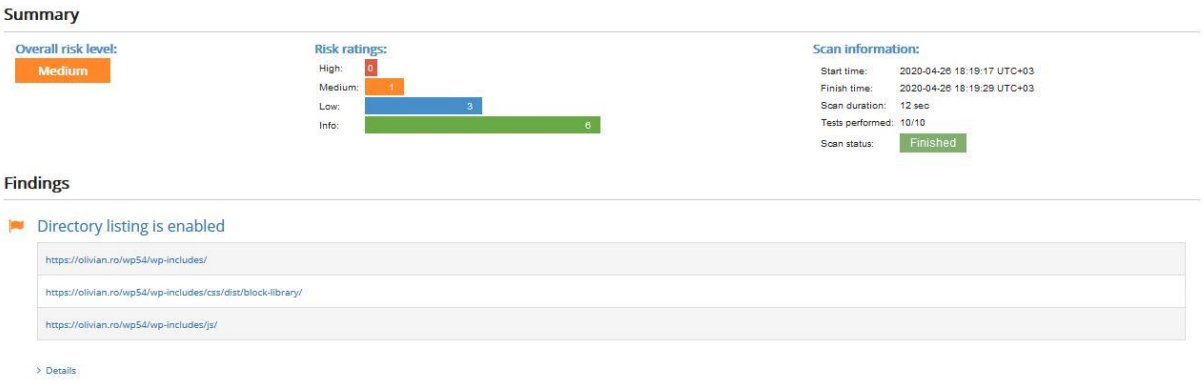
Findings

Vulnerabilities found for server-side software

Risk Level	CVSS	CVE	Summary	Exploit	Affected software
●	7.5	CVE-2018-20148	In WordPress before 4.9.9 and 5.x before 5.0.1, contributors could conduct PHP object injection attacks via crafted metadata in a wp.getMediaItem XMLRPC call. This is caused by mishandling of serialized data at phar:// URLs in the wp_get_attachment_thumb_file function in wp-includes/post.php.	N/A	WordPress 5.0
●	6.8	CVE-2019-9787	WordPress before 5.1.1 does not properly filter comment content, leading to Remote Code Execution by unauthenticated users in a default configuration. This occurs because CSRF protection is mishandled, and because Search Engine Optimization of A elements is performed incorrectly, leading to XSS. The XSS results in administrative access, which allows arbitrary changes to .php files. This is related to wp-admin/includes/ajax-actions.php and wp-includes/comment.php.	N/A	WordPress 5.0
●	6.5	CVE-2019-8942	WordPress before 4.9.9 and 5.x before 5.0.1 allows remote code execution because an _wp_attached_file Post Meta entry can be changed to an arbitrary string, such as one ending with a .jpg?file.php substring. An attacker with author privileges can execute arbitrary code by uploading a crafted image containing PHP code in the Exif metadata. Exploitation can leverage CVE-2019-8943.	N/A	WordPress 5.0
●	5.8	CVE-2019-16220	In WordPress before 5.2.3, validation and sanitization of a URL in wp_validate_redirect in wp-includes/pluggable.php could lead to an open redirect.	N/A	WordPress 5.0
●	5.5	CVE-2018-20147	In WordPress before 4.9.9 and 5.x before 5.0.1, authors could modify metadata to bypass intended restrictions on deleting files.	N/A	WordPress 5.0

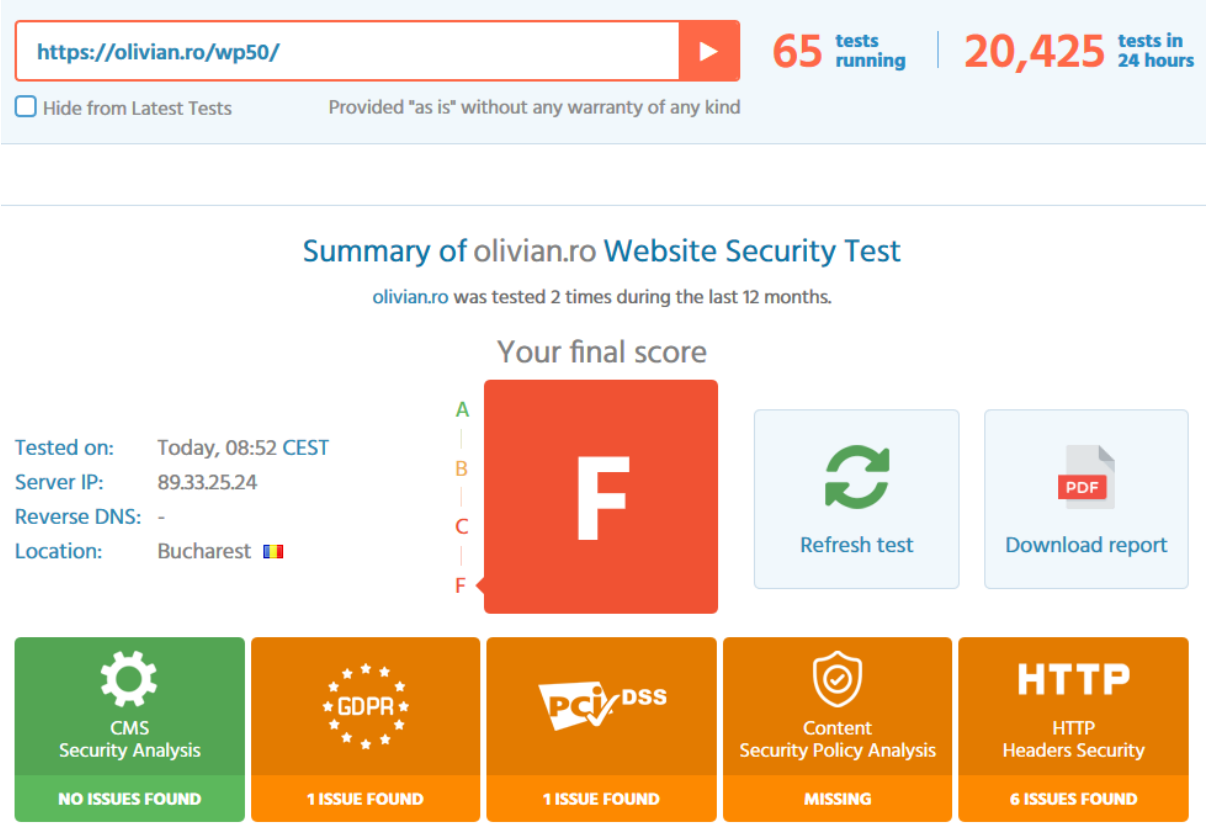
› Details

(Website Vulnerability Scanner - Online Scan for Web Vulnerabilities | Pentest-Tools.com, fără dată)

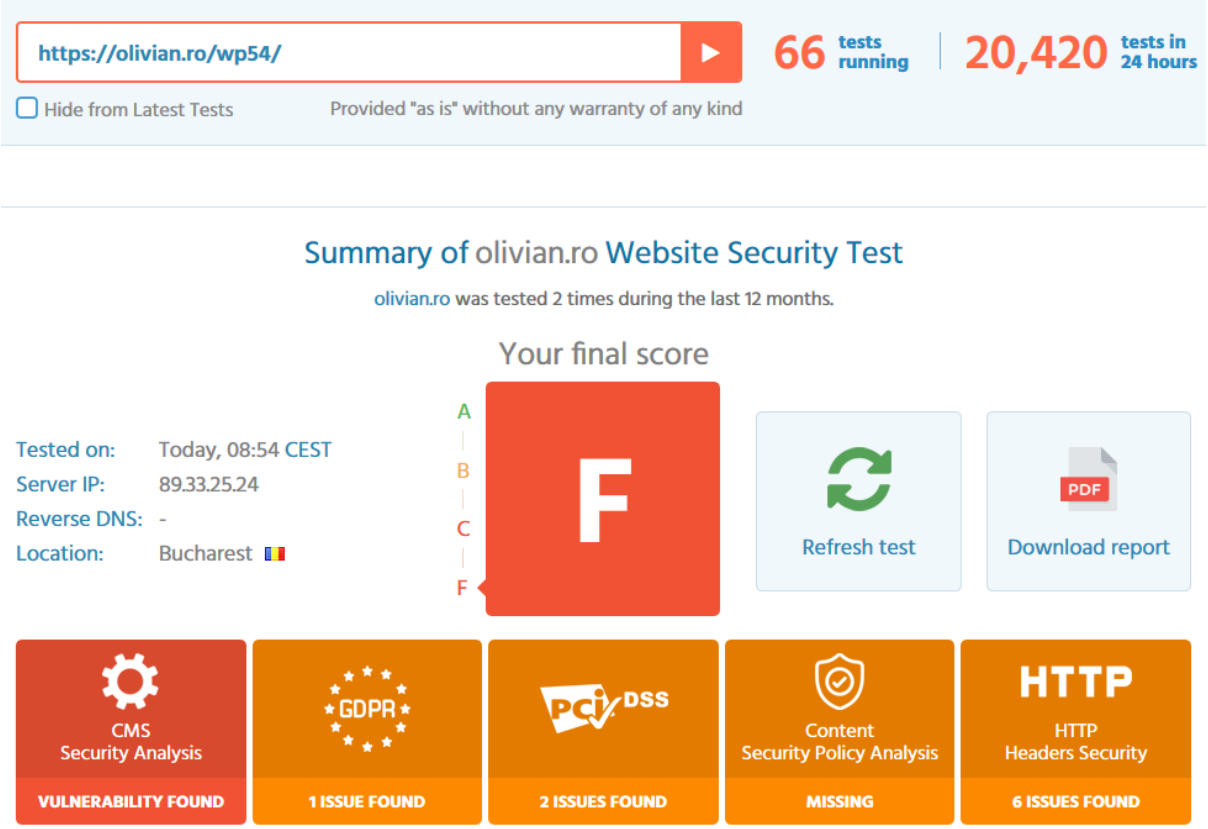


(Website Vulnerability Scanner - Online Scan for Web Vulnerabilities | Pentest-Tools.com, fără dată)

3.2. Testare immuniweb.com

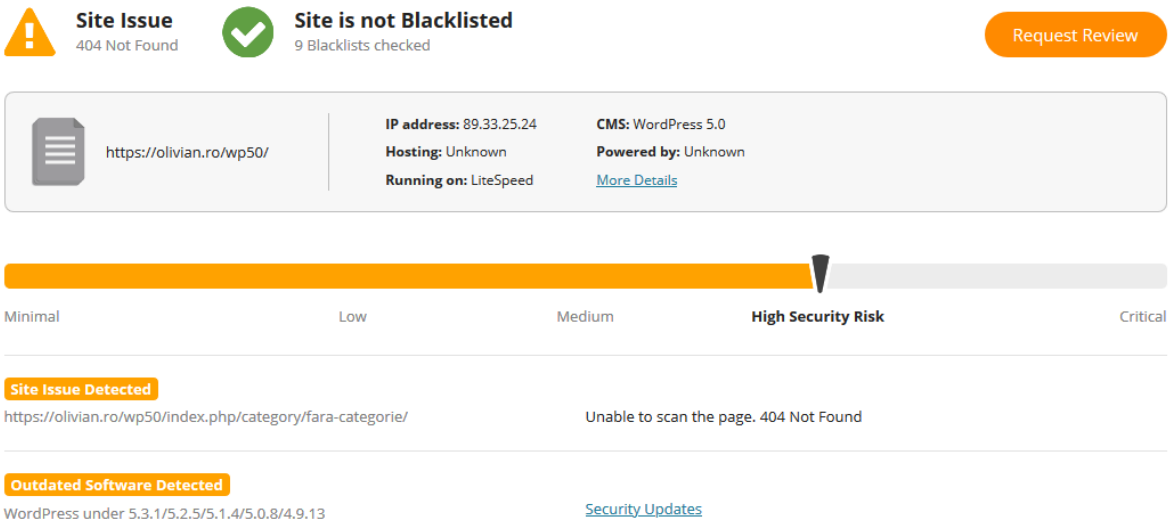


(Website Security Test of olivian.ro (89.33.25.24), fără dată)

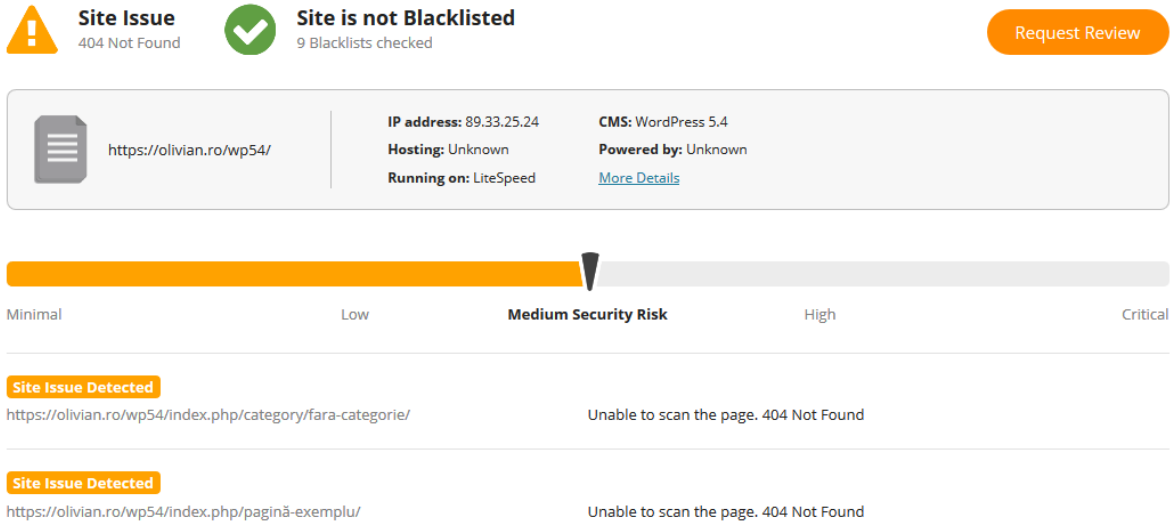


(Website Security Test of olivian.ro (89.33.25.24), fără dată)

3.3. Testare sucuri.net

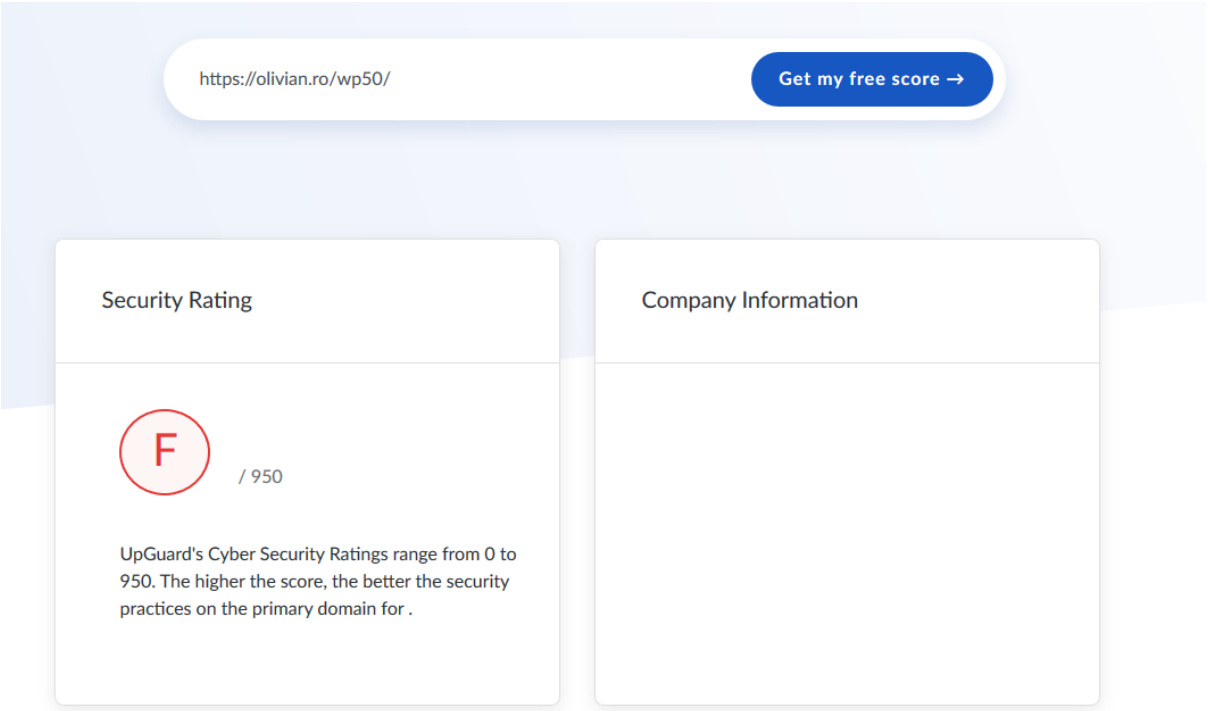


(https://olivian.ro/wp50/, fără dată)

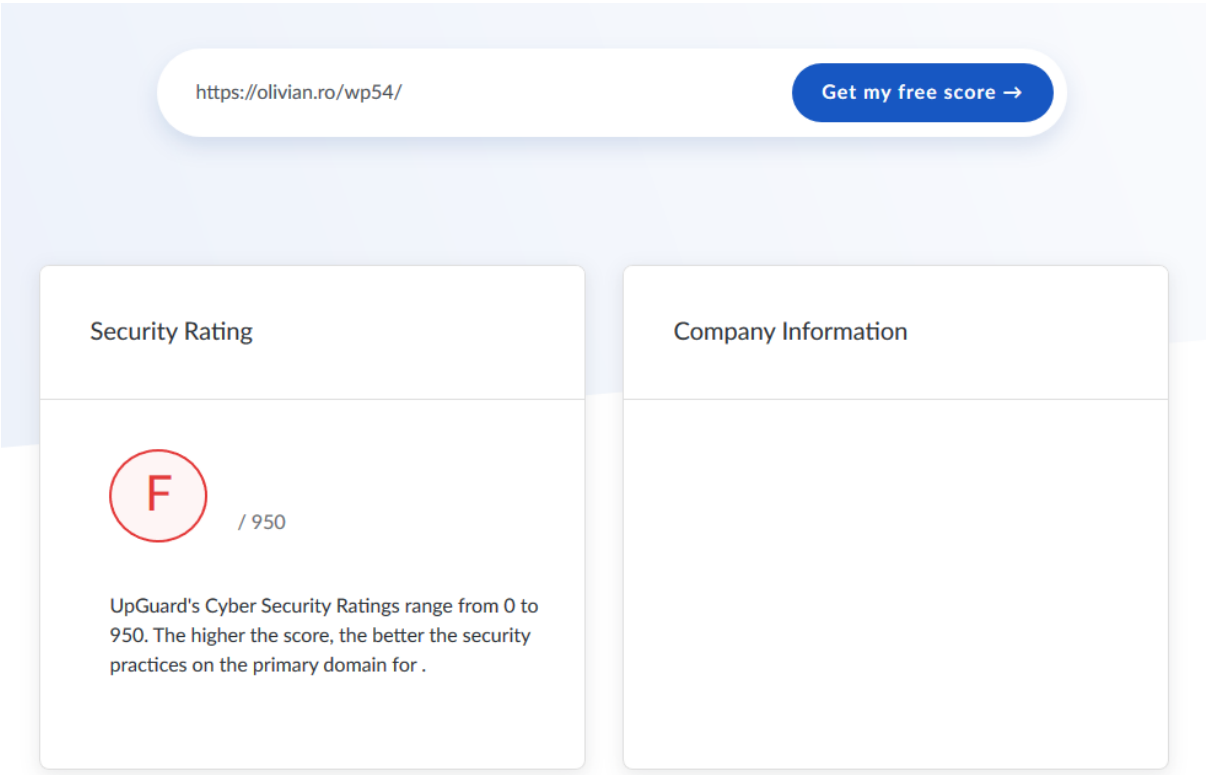


(https://olivian.ro/wp54/, fără dată)

3.4. Testare upguard.com

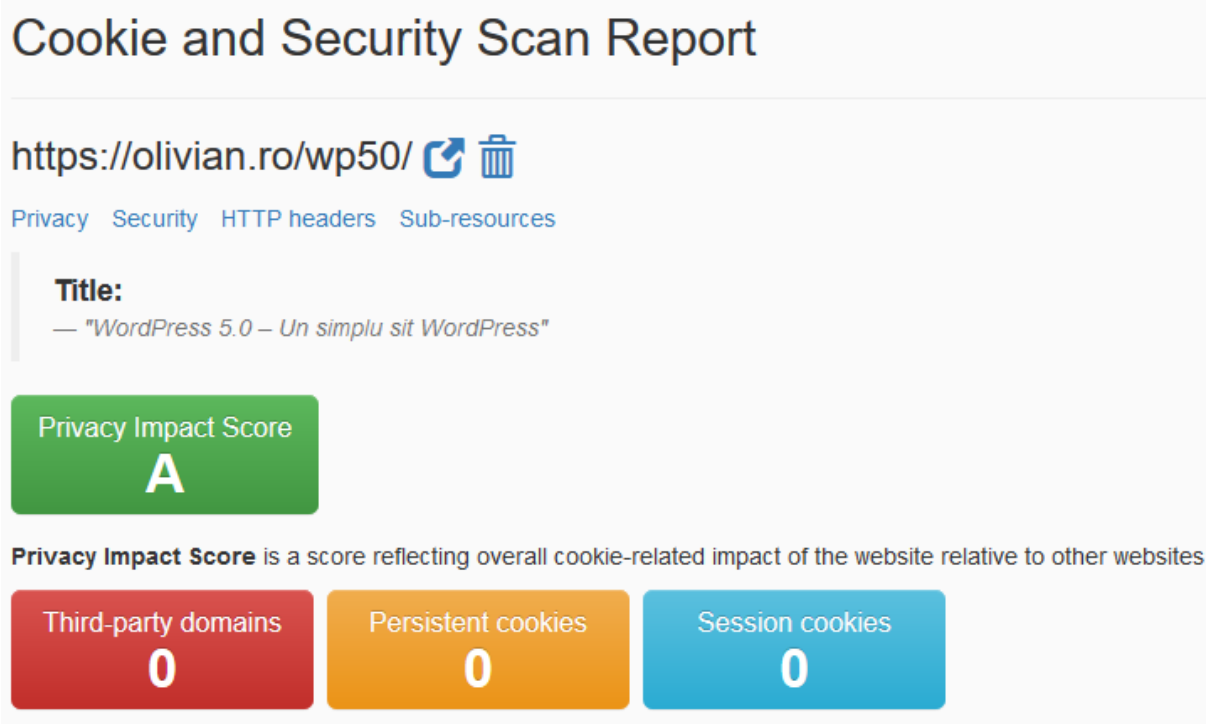


(Free Website Security Scan | UpGuard, fără dată)

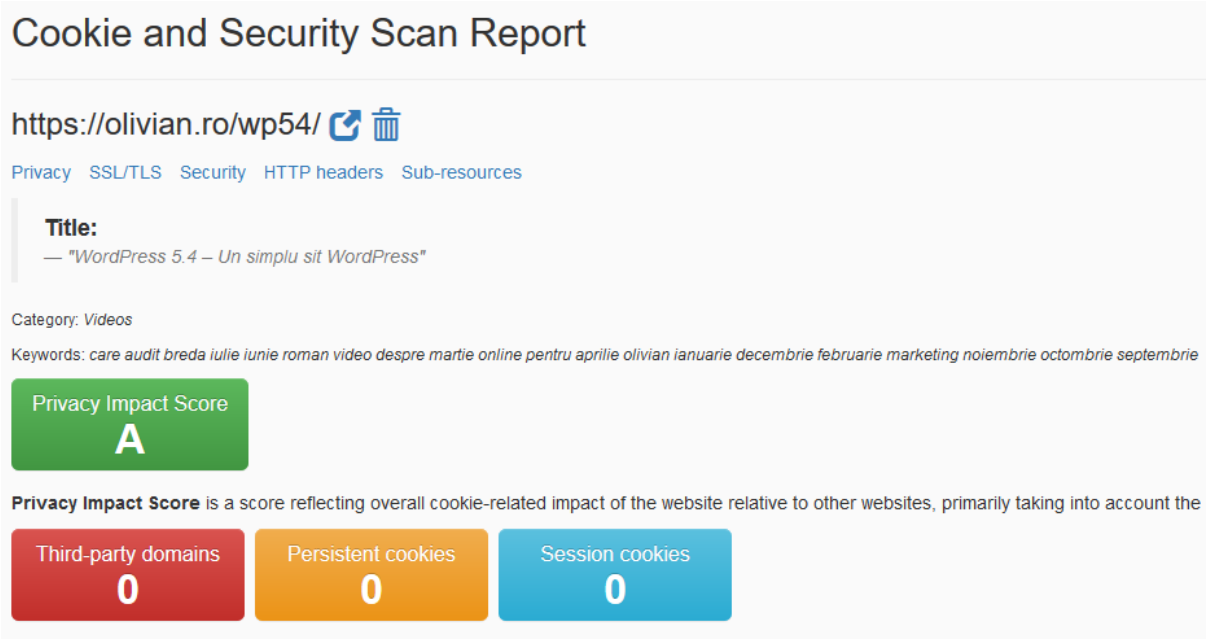


(Free Website Security Scan | UpGuard, fără dată)

3.5. Testare webcookies.org



(olivian.ro | Privacy & security report #30424440, fără dată)



(olivian.ro | Privacy & security report #30424448, fără dată)

3.6. Testare nstalker.com

3. Technical Summary

3.1. Scan Session Information

URL :	https://olivian.ro/wp50/
Date:	May 8, 2020 18:18:21
Scan Policy:	Webserver security (including SANS FBI)
SSL Cipher (Algorithm):	ECDHE-RSA-AES256-GCM-SHA384
Server Reported Banner:	LiteSpeed
Server Technology (Banner):	Unknown Server
Server Technology Detected:	Unknown Server
Server-side Technologies:	[PHP]

(N-Stalker Free Edition Version X, fără dată)

3.1. Scan Session Information

URL :	https://olivian.ro/wp54/
Date:	May 8, 2020 18:10:30
Scan Policy:	Webserver security (including SANS FBI)
SSL Cipher (Algorithm):	ECDHE-RSA-AES256-GCM-SHA384
Server Reported Banner:	LiteSpeed
Server Technology (Banner):	Unknown Server
Server Technology Detected:	Unknown Server
Server-side Technologies:	[PHP]

(N-Stalker Free Edition Version X, fără dată)

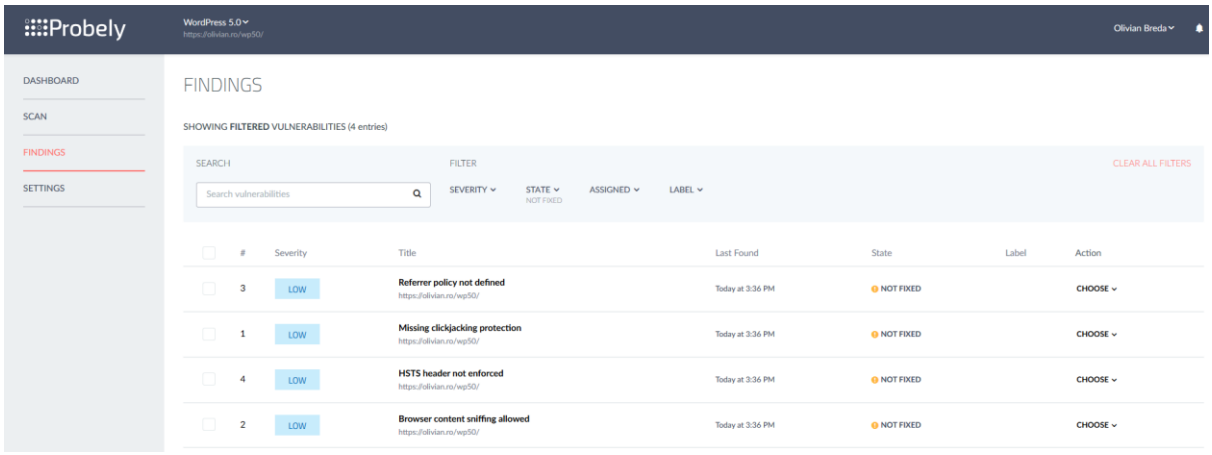
3.7. Testare hackertarget.com

Results of Passive WordPress Analysis

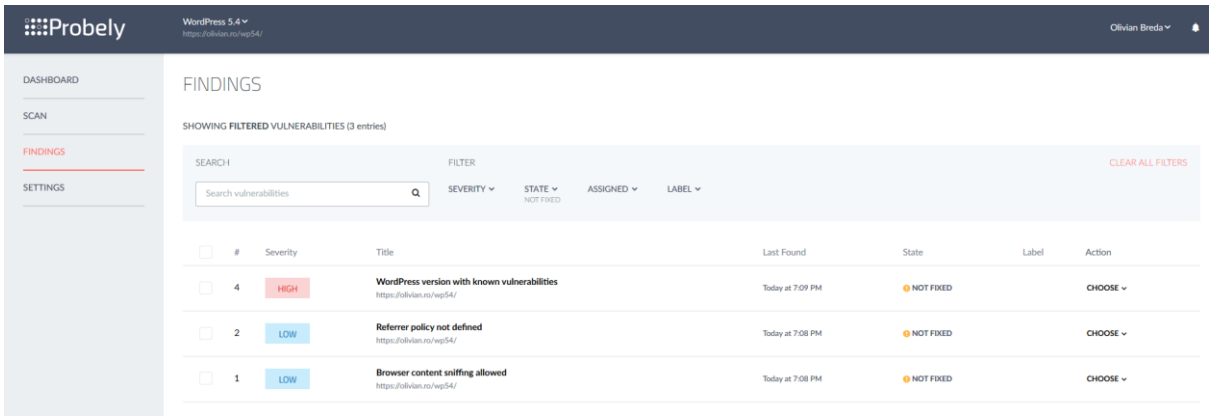
Site	Title	Version	Server	IP Address	ASN	Hosting	Location	Plugins / Theme
https://olivian.ro/wp50/	WordPress 5.0 – Un simplu sit WordPress	WordPress 5.0 (Meta Generator)	LiteSpeed	89.33.25.24	205275	ROMARG SRL	RO	**Theme:** twentynineteen
https://olivian.ro/wp54/	WordPress 5.4 – Un simplu sit WordPress	WordPress 5.4 (Meta Generator)	LiteSpeed	89.33.25.24	205275	ROMARG SRL	RO	**Theme:** twentytwenty

(28 Online Vulnerability Scanners & Network Tools | HackerTarget.com, fără dată)

3.8. Testare probely.com

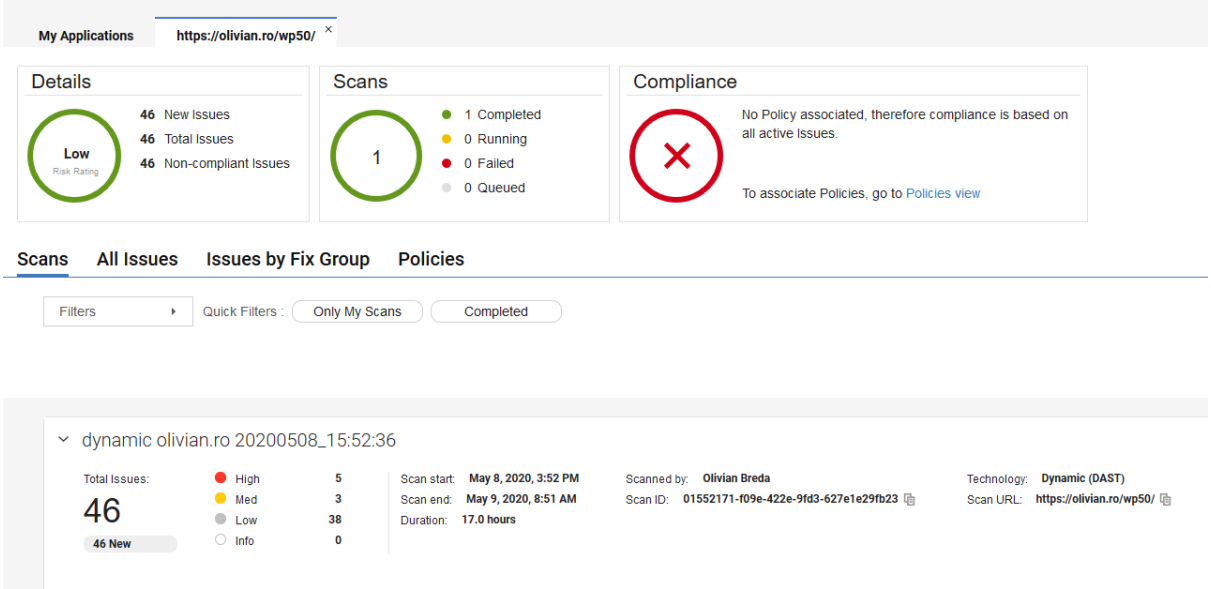


(Probely, fără dată)

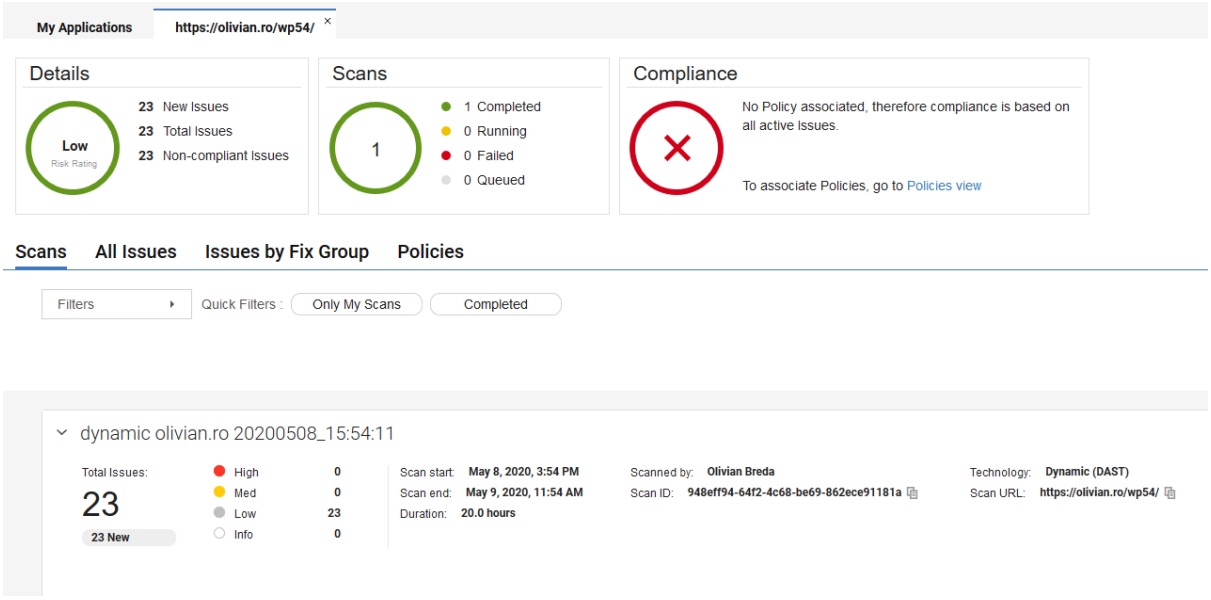


(Probely, fără dată)

3.9. Testare appscan.com



(HCL AppScan on Cloud, fără dată)



(HCL AppScan on Cloud, fără dată)

3.10. Testare rapid7.com

App Name: olivian.ro Scan Config: https://olivian.ro/wp50/

0d 0h 2m 30s

Duration of Scan

Scan Completed 05/08/20 4:10 PM

9

CrawledLinks

16

Vulns Discovered

843 Attacks Performed

Vulnerabilities by Type

Vulnerability Type	Amount
Content Security Policy Header	1
Http Headers	8
Javascript Memory Leaks	1
Vulnerability Type	Amount
X-content-type-options	2
X-frame-options	2
X-xss-protection	2

(Rapid7, fără dată)

RAPID7

Vulnerabilities Summary

olivian.ro/wp54

App Name: olivian.ro/wp54 Scan Config: https://olivian.ro/wp54

0d 0h 6m 40s

Duration of Scan

Scan Completed 05/10/20 9:53 AM

76

CrawledLinks

113

Vulns Discovered

13129 Attacks Performed

Vulnerabilities by Type

Vulnerability Type	Amount
Autocomplete Attribute	1
Browser Cache Directive (leaking Sensitive Information)	1
Collecting Sensitive Personal Information	2
Content Security Policy Header	2
Cookie Attributes	5
Http Headers	25

Vulnerability Type	Amount
Http Strict Transport Security	20
Javascript Memory Leaks	3
Sensitive Data Exposure	1
X-content-type-options	19
X-frame-options	13
X-xss-protection	21

(Rapid7, fără dată)

3.11. Testare zaproxy.org

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	5
Informational	2

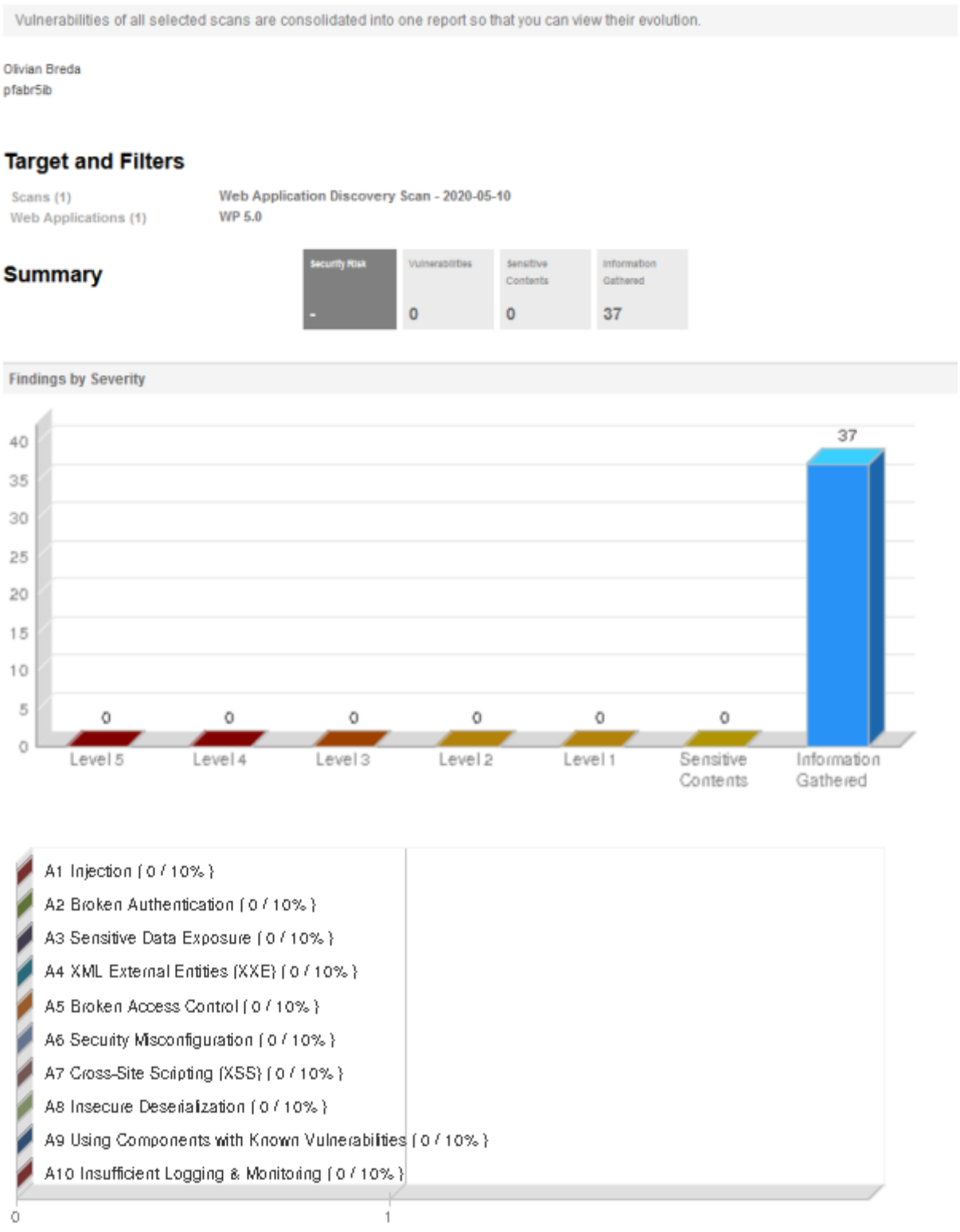
(OWASP ZAP, fără dată)

Summary of Alerts

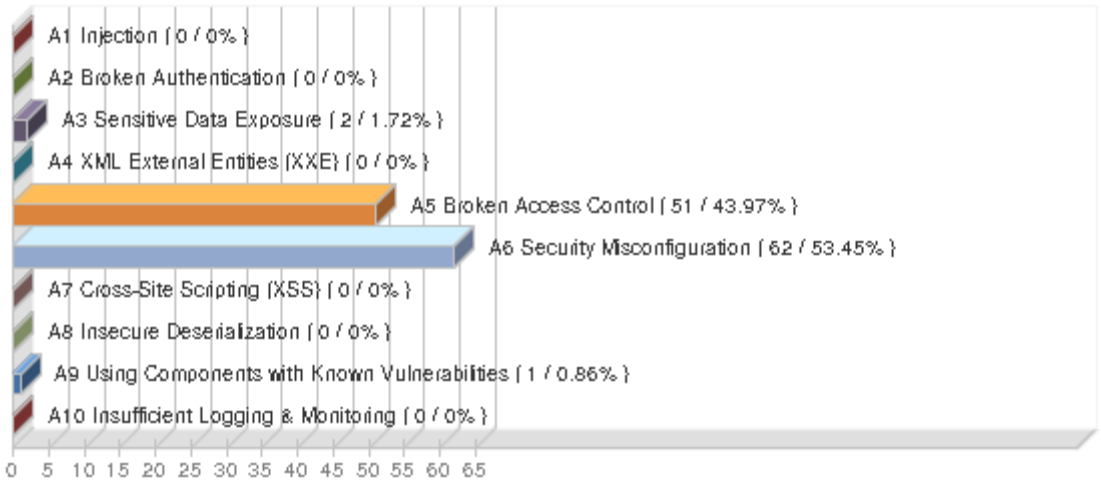
Risk Level	Number of Alerts
High	0
Medium	1
Low	5
Informational	2

(OWASP ZAP, fără dată)

3.12. Testare qualys.eu



(Qualys Security and Compliance Suite Login, fără dată)



Scan Report

Vulnerabilities of all selected scans are consolidated into one report so that you can view their evolution.

Olivian Breda
pfabr5ib

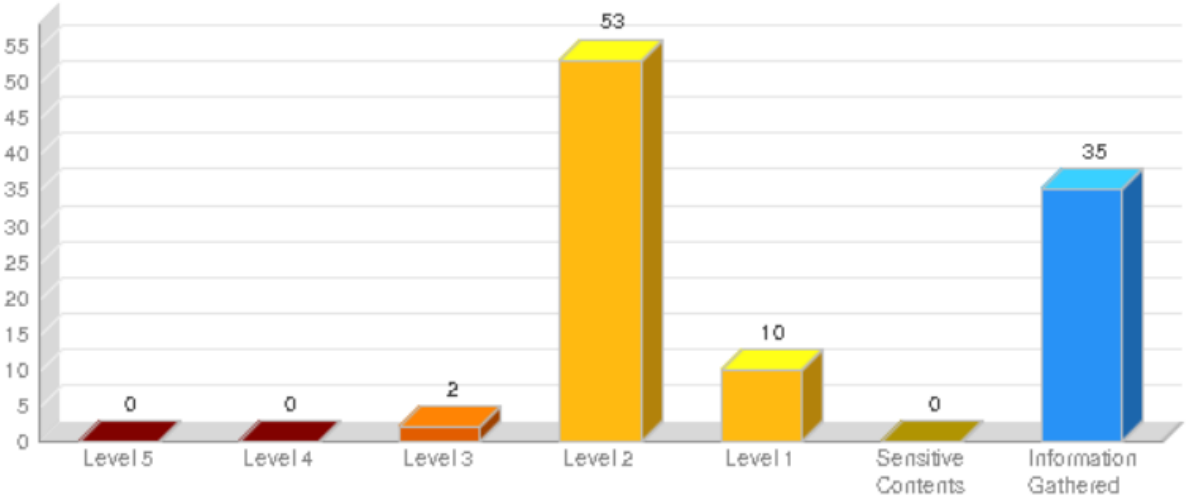
Target and Filters

Scans (1)	Web Application Vulnerability Scan - 2020-05-11
Web Applications (1)	WP 5.4

Summary

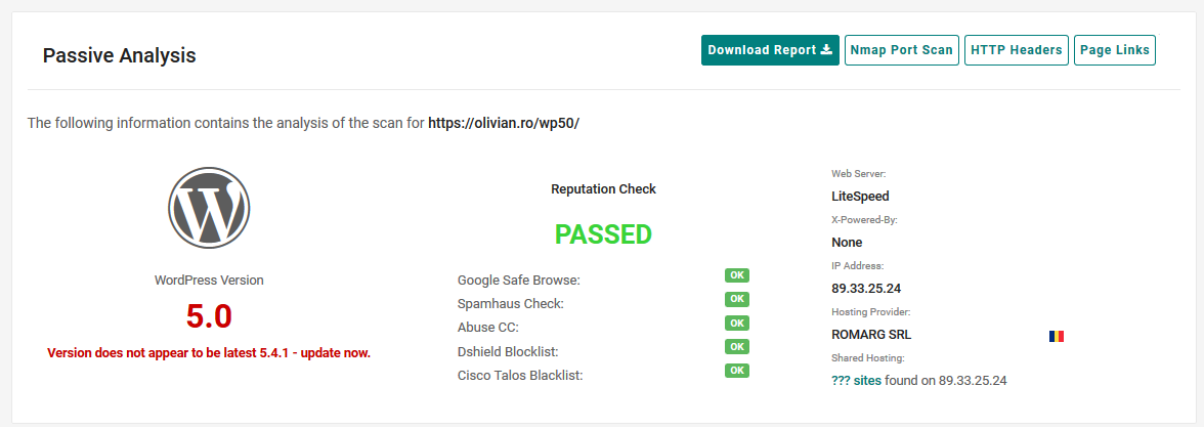
Security Risk	Vulnerabilities	Sensitive Contents	Information Gathered
MED	65	0	35

Findings by Severity

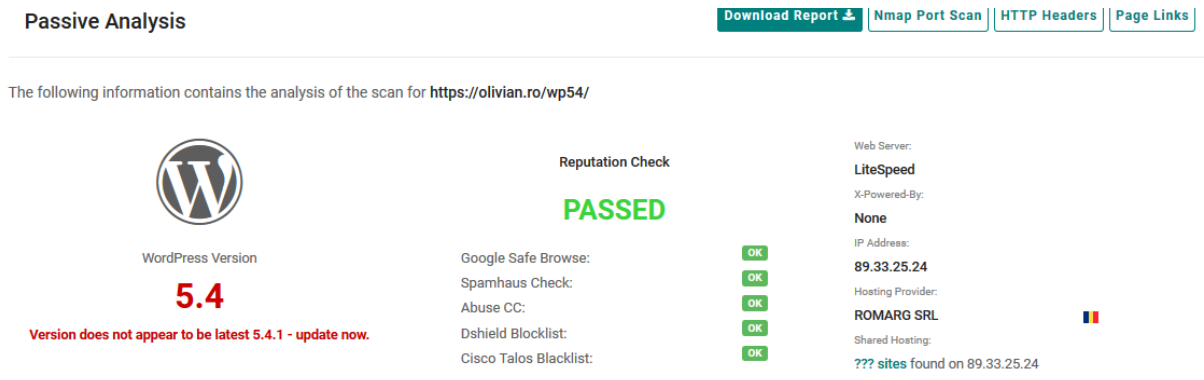


(Qualys Security and Compliance Suite Login, fără dată)

3.13. Testare hackertarget.com



(WordPress Security Scan | HackerTarget.com, fără dată)



(WordPress Security Scan | HackerTarget.com, fără dată)

4. Anexă - lista abrevierilor

2FA = Two-factor authentication - Autentificare în doi pași / cu doi factori.

AAA = Authentication, authorization, and accounting - Autentificare, Autorizare și Evidență - control drepturi utilizatori.

API = Application Programming Interface, set de definiții programare.

CAPTCHA = Completely Automated Public Turing test to tell Computers and Humans Apart - Test Turing public, complet automat, pentru a distinge calculatoarele de oameni. Este o metodă automată de a determina dacă persoana care face testul este om sau bot.

CCPA = The California Consumer Privacy Act - Legea privind confidențialitatea consumatorilor din California, un statut privind drepturile de confidențialitate și protecția consumatorilor rezidenților din California, Statele Unite ale Americii.

CDN = content delivery network, or content distribution network - rețea de livrare de conținut, rețea internațională de servere proxy.

CGI = Computer-Generated Imagery, aplicarea graficii computerizate pentru a crea fișiere imagini.

CMME = Content Management Made Easy - sistem facil de gestionare al conținutului unui site.

CMS = Content Management System - sistem de administrare a conținutului.

CSRF/XSRF = Cross-site request forgery - transmitere comenzi neautorizate de la un utilizator în care aplicația web are încredere.

CSS = Cascading Style Sheets - stiluri în cascadă, standard formatarea elemente HTML.

CVE® = Common Vulnerabilities and Exposures - vulernabilități și expuneri de date comun întâlnite, site care prezintă vulnerabilități de securitate cibernetică cunoscute public.

DH = Diffie–Hellman key exchange - metodă de transfer chei cripotgrafice.

DoS = Denial-of-service attack / DoS attack - atac cibernetic de tip oprire serviciu.

ECDH = Elliptic-curve Diffie–Hellman - protocol pentru schimbul cifrat al cheilor cripotgrafice.

EXIF = Exchangeable image file format - format comutabil pentru fișiere imagine.

FS / PFS = forward secrecy sau perfect forward secrecy - funcții pentru transferul cheilor criptografice.

GDPR = General Data Protection Regulation - Regulamentul general al Uniunii Europene privind protecția datelor.

GPL = General Public License - licența publică generală, utilă pentru transmitere drepturile de autor în anumite condiții.

HTML = HyperText Markup Language, limbaj pentru creare pagini web.

HTTP = Hypertext Transfer Protocol - metodă de a accesarea informații în online.

HTTPS = Secure Hyper Text Transfer Protocol - protocol de comunicații sigure.

IE = Internet Explorer - browser web realizat de compania Microsoft Corporation.

IIM = Information Interchange Model (IIM) - structură pentru fișiere și atribute de tip metadate.

iOS = iPhone OS (Operating System) - sitem de operare pentru telefonul mobile (smartphone).

IP address = Internet Protocol address - un protocol de Internet pentru transmiterea datelor.

IPTC eaders = Date IIM încorporate (incluse) în imagini.

NVD = National Vulnerability Database - bază de date națională a vulnerabilităților din S.U.A.

OS X = macOS (Operating System) - sistem de operare.

PHP = Hypertext Preprocessor - pre-procesor de hipertext, un limbaj de programare.

RSA = Rivest–Shamir–Adleman - algoritm criptografic cu chei publice.

SEO = Search Engine Optimization - optimizare pentru motoarele de căutare.

SFTP = SSH File Transfer Protocol - un protocol de rețea utilizat pentru transferul de fișiere sigur pe shell-ul securizat.

SPF = Sender Policy Framework - framework pentru politica de trimitere, o metodă de autentificare emailuri.

SQL = Structured Query Language - limbaj de interogare structurat pentru sisteme de manipulare a bazelor de date relaționale.

SSH = Secure Shell - un protocol de rețea criptografică pentru operarea serviciilor de rețea în siguranță printr-o rețea nesecurizată.

SSL = Secure Sockets Layer - protocol criptografic pentru comunicații sigure pe Internet.

SSL = Secure Sockets Layer - protocol criptografic pentru comunicații sigure pe Internet.

SVG = Scalable Vector Graphics - grafică vectorială proporționabilă, limbaj pentru imagini 2D (folosește XML).

TLS = Transport Layer Security - Succesor SSL - protocol criptografic pentru comunicații sigure pe Internet.

TLS = Transport Layer Security - Succesor SSL - protocol criptografic pentru comunicații sigure pe Internet.

TOFU = "trust on first use" - „încredere în prima utilizare”, protocol de transmitere date.

URL = A Uniform Resource Locator - localizator uniform de resurse.

UTF-7 = 7-bit Unicode Transformation Format, format de transformare a caracterlor în Unicode, format de texte definit de către Unicode Consortium.

UX = User Experience - uzabilitate.

VPS = Virtual Private Server - server virtual privat.

Win = Microsoft Windows - sistem de operare.

XML = Extensible Markup Language - limbaj de marcare pentru crearea de alte limbaje de marcare.

XMP = Extensible Metadata Platform (XMP) is an ISO standard, originally created by Adobe Systems Inc., for the creation, processing and interchange of standardized and custom metadata for digital documents and data sets.

XSS = Cross Site Scripting - introducere cod HTML sau JavaScript în pagini în mod automat.

XST = Cross-Site Tracing - Tracing printr-o întrepătrundere a unor site-uri, implică folosirea Cross-site Scripting (XSS) și a metodelor TRACE sau TRACK HTTP.

XXE = XML external entity injection - o vulnerabilitate de securitate web care permite unui atacator să interfereze cu prelucrarea unei aplicații de date XML.