Razvoj robusnog sigurnosnog okvira za CMS platforme

Juras, Ivan

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Ivan Juras

RAZVOJ ROBUSNOG SIGURNOSNOG OKVIRA ZA CMS PLATFORME

ZAVRŠNI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Ivan Juras

Matični broj: 0016156344

Studij: Informacijski i poslovni sustavi

RAZVOJ ROBUSNOG SIGURNOSNOG OKVIRA ZA CMS PLATFORME

ZAVRŠNI RAD

Mentor:

Doc. dr. sc. Igor Tomičić

Varaždin, rujan 2024.

Ivan Juras

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Razvoj sigurnosnog okvira je ključan dio internetske sigurnosti jer sadrži skup alata, resursa, dokumentacije i smjernica dizajniran kako bi maksimizirao sigurnost web stranica zasnovanih na CMS-u, a sve na temelju prethodnog početnog istraživanja i studija, penetracijskog testiranja i analize ranjivosti.

Provedeno je početno istraživanje postojećih sigurnosnih protokola i uobičajenih ranjivosti unutar najčešće korištenih CMS platformi. Isto tako provedeno je penetracijsko testiranje kako bi se identificirale ranjivosti platforme koje mogu ugroziti sigurnost sustava. Na temelju tih nalaza, razvijen je sigurnosni okvir koji obuhvaća automatizirane alate za otkrivanje ranjivosti, sigurnosne module i preporuke za najbolju praksu pri konfiguriranju i održavanju CMS platformi.

Praktični dio rada uključuje konfiguraciju popularne CMS platforme WordPress s ciljem da bude ranjiva te zatim slijedi analiza ranjivosti i penetracijsko testiranje pronađenih nedostataka. Na kraju primijenjen je sigurnosni okvir na CMS platformu, s ciljem procjene njegove učinkovitosti u smanjenju sigurnosnih rizika.

Ključne riječi: CMS, Sigurnosni okvir, Penetracijski test, Ranjivosti

Sadržaj

1. Uvod

U današnje vrijeme sve više ljudi i organizacija se koristi CMS platformama, istraživanja su pokazala da od svih stranica na internetu CMS platforme čine 68.7% tih stranica. Baš zato što su ove platforme lake za korištenje jer su prirode otvorenog koda i jer su toliko rasprostranjene na internetu, česta su meta hakerskih napada.

CMS platforme poput WordPress-a, Joomla i Drupal-a imaju dodatke i teme koje koriste, a upravo ti dodatci i teme su izvor ranjivosti koje platforma može sadržavati iako ponekad i sama verzija CMS platforme može sadržavati ranjivosti. Priroda tih ranjivosti je raznolika, ali najčešći napadi povodom ranjivosti su SQL injection, Cross-Site Scripting (XSS) i daljinsko izvršavanje koda (RCE). Ako se dogodi neki od ovih navedenih napada, mogu dovesti do značajnih sigurnosnih proboja, uključujući neovlašteni pristup i krađu podataka. Trendovi u 2024. godini su pokazali da napadači koriste automatizirane alate za otkrivanje i iskorištavanje ranjivosti, čime su CMS platforme na meti.

S obzirom na ključnu ulogu CMS platformi u web infrastrukturi ovaj rad se bavi razvojem sigurnosnog okvira za njih. Nadalje ovaj rad sadrži osim sigurnosnog okvira, praktično testiranje popularne CMS platforme WordPress korištenjem različitih alata i tehnika za otkrivanje ranjivosti, analizu ranjivosti i primjenu sigurnosnog okvira.

2. CMS platforme

CMS je skraćenica od „Content management system“ tj. sustav za upravljanje sadržajem i razlog velike popularnosti tih sustava je upravo činjenica da pomaže korisnicima da kreiraju, upravljaju i mijenjaju sadržaj na web stranici bez potrebe za ikakvim tehničkim znanjem. CMS platforme sadrže sučelje za lako i intuitivno korištenje same platforme i s pomoću njih se mogu napraviti različite vrste web stranica poput: Statične web stranice, Blogovi, E-trgovine, Forumi, Društvene mreže, Online tečajevi, Portfolio stranice i tako dalje [1].

CMS platforme sastoje se od dva dijela:

- **Aplikacija za upravljanje sadržajem** (CMA) – ovo je dio koji vam omogućuje da zapravo dodate i upravljate sadržajem na svojoj stranici [1].
- **Aplikacija za isporuku sadržaja** (CDA) – ovo je pozadina, proces iza scene koji preuzima sadržaj koji unesete u CMA, pravilno ga pohranjuje i čini vidljivim vašim posjetiteljima [1].

2.1. Podjela CMS platformi

Prema Deane Barkeru [2, str 36.] postoje CMS platforme otvorenog koda, isto tako postoje komercijalne CMS platforme i softver kao usluga za korištenje CMS platformi te isto tako tvrtke koje imaju razvojne timove mogu razviti svoju CMS platformu.

CMS platforme otvorenog koda su ujedno i najkorištenija vrsta CMS platformi, neke od njih su WordPress, Drupal i Joomla, one su besplatne za korištenje, točnije ne treba se plaćati nikakva licenca. Iako se ne plaća licenca korisnici će ovisno o potrebi trebati plaćati hosting web stranice. Ove platforme imaju jako dobre zajednice korisnika koje mogu odgovoriti na mnoga pitanja brzo i točno. Međutim, to može biti razlog nedostatka profesionalne podrške. Obični softver je besplatan, ali postoji opcija plaćanja koja omogućava pristup više funkcionalnosti. Nekada je besplatna verzija potpuno dobra, a nekada je samo loša probna verzija kako bi natjerala korisnike da plate za potpunu verziju softvera. Mnogi pružatelji usluga CMS platformi nude upravljani hosting za sustave otvorenog koda koje oni razvijaju, prednost toga je hosting okruženje koje je dizajnirano specifično za taj sustav i eksperti tog sustava za hosting su na raspolaganju u slučaju ako dođe do kakvih problema [2, str. 38].

Komercijalne CMS platforme iako nisu besplatne one se predstavljaju kao formalni poslovni subjekt i taj dio profesionalnosti i formalnosti je bitan nekim organizacijama. Komercijalni pružatelji usluga generalno se pridržavaju većem standardu kvalitete i funkcionalnosti, jer trebaju održavati korisnike koji plaćaju za njihovu uslugu sretnima i upravo ti korisnici su izvor njihove zarade. Kao i bilo koja generalizacija ovo nije skroz točno jer postoje platforme otvorenog koda koje imaju jednake, ako ne i bolje funkcionalnosti i standarde nego komercijalne platforme. Posljednjih godina došlo je do velike razlike između platformi otvorenog koda i komercijalnih platformi, platforme otvorenog koda su pretežno zadužene za upravljanje sadržajem, dok su komercijalne platforme zadužene za sadržajni marketing, a to su alati i značajke koje pomažu poboljšati sadržaj nakon što se objavi. Cijena komercijalne CMS platforme ovisi o mnogim značajka kao što su: broj korisnika koji uređuju sadržaj, broj servera na kojima se platforma pokreće, broj web stranica, dodatci, količina sadržaja pod upravljanjem. Pojedine komercijalne platforme osim inicijalne cijene kupnje softvera imaju i pretplate [2, str. 42].

Softver kao usluga za korištenje CMS platformi funkcionira tako da korisnici umjesto da plaćaju i instaliraju CMS platformu, radije plaćaju mjesečnu pretplatu i pokreću web stranicu unutar velikog sustava upravljanog od strane pružatelja usluge. Time korisnici postanu jedan od mnogih korisnika koji pokreću web stranice unutar istog sustava. Prednosti su brzo vrijeme pokretanja i nema problema s hostingom. Još jedna prednost je da je platforma uvijek na najnovijoj verziji. Budući da pružatelj usluge upravlja hosting okruženjem, kada izdaju novu verziju, korisnik je dobiva odmah [2, str. 45]. U zadnje vrijeme došlo je do pada korisnika, jer korisnici žele određeno prilagođavanje stranice, a pružatelji usluge to ne mogu pružiti.

Razvoj vlastite CMS platforme nudi mnoge pogodnosti poput toga da nije potrebno plaćati licencu za korištenje platforme, osoba ili tim koji razvija platformu će biti ekspert u korištenju vlastitog sustava i time neće trebati prolaziti krivulju učenja te implementirati će samo potrebne funkcionalnosti, bez nepotrebnih dijelova i kompliciranja. Izvana CMS platforma izgleda kao jednostavni alat za upravljanje i objavljivanje sadržaja, ali na dubljoj razini kada korisnici počinju pitati za funkcionalnosti poput verzioniranja i više jezika tada se vidi koliko ih je teško za implementirati. Razvoj CMS platforme počne poprilično dobro i jednostavno dok imaju jednostavne CRUD operacije (Create, Read, Update, Delete) koje su jako brze za implementirati, ali druge funkcionalnosti će učiniti da se vrijeme razvoja značajno poveća [2. str, 48].

2.2. Vrste CMS platformi

Postoji 5 glavnih vrsta CMS platformi: Komponentni sustavi za upravljanje sadržajem (CCMS), Sustavi za upravljanje poslovnim sadržajem (ECM), Sustavi za upravljanje web sadržajem (WCMS), Sustavi upravljanja digitalnom imovinom (DAM) i Sustavi za upravljanje dokumentima (DMS) [3].

2.2.1. Komponenti sustavi za upravljanje sadržajem (CCMS)

Komponentni sustav upravljanja sadržajem (CCMS) razlikuje se od običnog CMS-a jer umjesto obrade materijala stranicu po stranicu, skuplja riječi, izraze, odlomke ili slike (komponente) i čuva ih u jednoj pohrani. Komponente se čuvaju samo jednom kako bi se povećala ponovna upotreba sadržaja. CCMS je stalni izvor za distribuciju informacija na brojnim uređajima, uključujući mobilne, PDF i ispisne uređaje.

Primjeri sustava za upravljanje sadržajem komponente su Xyleme, Paligo, Documentum.

Prednosti CMMS-a:

- **Ponovna iskoristivost**: Korištenje CCMS-a za ponovno korištenje sadržaja štedi vrijeme pisanja, uređivanja i objavljivanja.
- **Jedno izvorište**: CCMS omogućuje distribuciju sadržaja na brojne kanale, kao što su ispis, mobitel, web, chatbotovi i još mnogo toga.
- **Sljedivost**: CCMS omogućuje pregled izmjena sadržaja prikazujući identitet urednika, vrijeme izmjena i njihovu lokaciju.

2.2.2. Sustavi za upravljanje dokumentima (DMS)

Sustav za upravljanje dokumentima (DMS) rješenje je temeljeno na oblaku za upravljanje, pohranjivanje i praćenje dokumenata. Nudi automatizirano rješenje za učitavanje, obradu i distribuciju poslovnih dokumenata, eliminirajući potrebu za ispisom, kopiranjem ili skeniranjem.

Primjeri sustava za upravljanje dokumentima su Google Workspace, Dropbox, OneDrive i iCloud.

Prednosti DMS-a:

- **Ekološki**: Štedi papir i smanjuje otpad od papira.
- **Sigurnost**: DMS pruža više razina sigurnosti kako bi se osiguralo da povjerljivi sadržaj ostane u rukama ovlaštenih korisnika.

- **Mobilnost i rad na daljinu**: DMS omogućuje pristup i ažuriranje dokumenata s bilo kojeg mjesta [3].

2.2.3. Sustavi za upravljanje poslovnim sadržajem (ECM)

Sustav za upravljanje poslovnim sadržajem (ECM) je CMS sustav koji tvrtkama omogućuje stvaranje, upravljanje i distribuciju brojnih oblika sadržaja. Također jamči da se informacije i dokumenti šalju namijenjenoj osobi: zaposleniku, izvršnom direktoru, poslovnom partneru ili kupcu, ovisno cilju.

Zaposlenici mogu jednostavno dohvatiti pohranjeni sadržaj potreban za ispunjavanje svojih dnevnih dužnosti pomoću ECM-a. Također, ECM sustav automatski briše datoteke nakon roka zadržavanja kako bi oslobodio suvišni zauzeti prostor.

Primjeri sustava za upravljanje poslovnim sadržajem su DocuShare, Zoho Docs i eFileCabinet.

Prednosti ECM-a:

- **Prilagodljivost**: ECM omogućuje pohranjivanje sadržaja u bilo kojem formatu ili vrsti datoteke i automatsko rukovanje njime.
- **Automatizacija upravljanja dokumentima**: Sustav se automatski brine za upravljanje dokumentima.
- **Ekonomičnost pohrane**: ECM pohranjuje samo potrebne podatke dok briše ostale [4].

2.2.4. Sustavi za upravljanje web sadržajem (WCMS)

Sa sustavima za upravljanje web sadržajem korisnici mogu upravljati digitalnim komponentama web stranice bez prethodnog znanja programskih jezika ili web programiranja. WCMS olakšava upravljanje digitalnim sadržajem i web stranicama pružajući alate za suradnju, autorstvo i administraciju. Za razlike od većine drugih CMS-ova koji obrađuju i ispisni i web sadržaj, WCMS obrađuje samo mrežni sadržaj.

Primjeri sustava za upravljanje web sadržajem su Hubspot, WordPress, Webflow i Duda.

Prednosti WCMS-a:

- **Automatizacija**: Sustav upravljanja web sadržajem automatizira svakodnevne zadatke poput objavljivanja informacija.
- **Personalizacija**: WCMS prilagođava iskustvo na temelju ciljane publike i omogućuje korisnicima da personaliziraju stil i sadržaj web stranice.
- **Skalabilnost**: WCMS platforma podupire proširenja za rukovanje rastućim prometom ili količinama sadržaja bez ograničenja [4].

2.2.5. Sustavi upravljanja digitalnom imovinom (DAM)

Sustav upravljanja digitalnom imovinom (DAM) omogućuje proizvodnju, upravljanje, pohranjivanje, organiziranje i distribuciju digitalne imovine. DAM temeljen na oblaku centralizirana je biblioteka koja zaposleniku, klijentu ili izvođaču omogućuje praktičan pristup sadržaju s bilo koje lokacije. DAM se u početku koristio samo za pohranu medijskih podataka kao što su slike, filmovi i audio zapisi. Ovi sustavi sada podržavaju različite formate, uključujući logotipe, fontove, papire i još mnogo toga.

Primjeri sustava za upravljanje digitalnom imovinom su MediaValet, Brainfolder i Bynder.

Prednosti DAM-a:

- **Objavljivanje druge strane**: DAM sustavi omogućuju objavljivanje sadržaja distribucijskim sustavima trećih strana, portalima, kanalima društvenih medija i još mnogo toga.
- **Prva centralizirana knjižnica**: DAM sprema sadržaj u centralizirano spremište, omogućujući različitim korisnicima da mu lako pristupe.
- **Administracija**: Korisnici mogu jednostavno upravljati portalima sadržaja korištenje DAM funkcionalnosti [4].

2.3. Sigurnosni izazovi u CMS platformama

Sigurnosno okruženje za CMS platforme prepuno je izazova, prvenstveno zbog njihove široke primjene i raznolikog sustava dodataka i tema koje podržavaju. Najčešće sigurnosne ranjivosti uključuju:

- **SQL injection:** SQL injection je među najčešćim napadima na CMS-ove. SQL injection se provodi tako da napadač unese proizvoljan SQL kod u sloj baze podataka. To napadačima omogućuje izdavanje izravnih naredbi baze podataka i manipuliranje bazom podataka kao da je korisnik CMS-a. Zbog novih sigurnosnih mjera koje su implementirali

CMS-ovi i relativno lakih metoda za sprječavanje ovih napada, oni su svakim danom sve manje učinkoviti [5].

- **Brute-force napadi:** Brute-force napade može izvesti svatko jer uključuju unos više kombinacija akreditacija korisnika, sve dok se napadač ne uspije prijaviti kao korisnik. Neki CMS-ovi prema zadanim postavkama ne ograničavaju broj pokušaja prijave, što znači da su korisnici koji koriste te CMS-ove izloženi zlonamjernim pokušajima koji mogu unijeti stotine ili tisuće akreditacija dok ne pronađu onu koja radi. Čak i ako brute-force napad ne uspije i dalje može izazvati kaos na poslužitelju jer će previše pokušaja preopteretiti i usporiti sustav [5].
- **DDoS:** Distribuirani denial-of-service je poboljšana verzija napada denial-of-service gdje zlonamjerni napadač šalje veliku količinu zahtjeva poslužitelju sa svrhom da se sruši ili učini nedostupnim korisnicima kojima je namijenjen. DDoS napadi često se izvode preko mnogo različitih strojeva, također poznatih kao botnetovi, koji skrivaju porijeklo zahtjeva [5].
- **Arbitrary Remote Code Execution (Proizvoljno daljinsko izvršavanje koda):** Iako proizvoljno ubacivanje koda zahtijeva više resursa nego druge vrste kibernetičkih napada, ubacivanje koda u web mjesto ili aplikaciju može imati opasne posljedice za privatnost i podatke korisnika. Proizvoljno daljinsko izvršavanje koda koristi bilo koju površinu za napad. Šalje dio PHP koda u udaljeno okruženje za izvršavanje. Bez odgovarajuće sigurnosti, pokrenut će se kao da je od korisnika, otvarajući udaljena stražnja vrata za napadače kako bi dobili pristup ciljanom okruženju [5].
- **Cross-Site Scripting (XSS):** Ova vrsta CMS ranjivosti iskorištava klijentsko okruženje unutar preglednika, što napadaču omogućuje ubacivanje proizvoljnog koda u ciljnu instancu i okolinu. Ovaj se napad događa na strani klijenta, što znači da može ugroziti osjetljive korisničke podatke i omogućiti manipulaciju bazama podataka i pohranjenim varijablama. Tradicionalne CMS platforme poput Drupala i WordPressa posebno su osjetljive na XSS ranjivosti zbog veće upotrebe okruženja na strani klijenta [5].
- **File Inclusion Exploitation (Iskorištavanje uključivanja datoteka):** Ranjivosti uključivanja datoteka često se nalaze na loše kodiranim stranicama. Ova vrsta ranjivosti javlja se kada web mjesto dopušta korisnicima unos ili učitavanje datoteka na poslužitelj. PHP kôd ne potvrđuje unos što rezultira isporukom zlonamjernih datoteka na poslužitelj. U eksploataciji uključivanja datoteka, korisnici mogu dobiti pristup osjetljivim podacima kada su poslužitelji pogrešno konfigurirani ili kada korisnik ima visoke privilegije [5].

- **Automatizirani napadi i prijetnje potpomognute umjetnom inteligencijom:** Sve veća zabrinutost u pogledu sigurnosti CMS-a je porast automatiziranih alata i napada potpomognutih umjetnom inteligencijom. Napadači sve više koriste modele strojnog učenja za skeniranje CMS platformi u potrazi za ranjivostima, što im omogućuje otkrivanje i iskorištavanje slabosti brže nego ikad prije. Prema izvješću iz 2024. o modelima velikih jezika u kibernetičkoj sigurnosti, napadači koji koriste alate umjetne inteligencije sposobni su zaobići tradicionalne sigurnosne mjere automatiziranjem zadataka kao što su otkrivanje ranjivosti i brutalno probijanje lozinki. Ovaj trend ukazuje na to da su CMS platforme sve osjetljivije na napade vođene umjetnom inteligencijom, koji mogu nadjačati osnovne sigurnosne okvire koji se oslanjaju na ručnu intervenciju. [6]
- **Napadi na lanac opskrbe putem CMS dodataka**: značajan porast napada na lanac opskrbe zabilježen je 2023., posebno na CMS platformama koje se uvelike oslanjaju na dodatke trećih strana. Studija o strategijama kibernetičke sigurnosti u modernim organizacijama naglašava kako napadači ugrožavaju CMS okruženja ciljajući na ranjive dodatke. Napadači dobivaju pristup putem nesigurnih ili zastarjelih dodataka, koji zatim djeluju kao pristupnik široj CMS infrastrukturi. [7]

2.4. Postojeće sigurnosne mjere

Kako bi se spriječile te ranjivosti, razvijeno je nekoliko sigurnosnih mjera:

- **Redovita ažuriranja i popravci:** Ažuriranja softvera moraju se instalirati čim se objave, jer sadrže najnovije sigurnosne zakrpe. Mnogi proizvođači nude automatska ažuriranja, stoga se savjetuje aktivirati tu funkciju [8].
- **Dvofaktorska autentifikacija:** Uz uobičajenu autentifikaciju (korisničko ime i lozinka) za pristup administrativnom području, preporučuje se korištenje dvofaktorske autentifikacije [8].
- **Ograničavanje administratorskog pristupa određenim IP adresama:** Moguće je ograničiti pristup administrativnoj stranici određenim IP adresama, rasponima IP adresa ili geolokaciji IP adrese. Odgovarajuća proširenja (plug-inovi) već postoje za širok raspon različitih sustava za upravljanje sadržajem [8].
- **Zaštita računala webmastera:** Web stranice i CMS-ovi često su ugroženi ukradenim FTP akreditacijama. Obično se dobivaju instaliranjem trojanaca na računalo webmastera. Stoga je potrebno osigurati da računalo koje se koristi ne sadrži zlonamjerni softver i da

je zaštićeno najnovijom zaštitom od virusa. Osim toga, FTP veza treba biti šifrirana gdje je to moguće (korištenje sFTP-a) [8].

- **Vatrozid web aplikacije (WAF):** Korištenjem vatrozida web aplikacije, web bazirani napadi na web stranice mogu se blokirati čak i prije nego dođu do aplikacije [8]. Postoje različiti sigurnosni dodatci za različite platforme koje nude ovu funkcionalnost.
- **Rano otkrivanje ranjivosti:** Cilj je identificirati potencijalne ranjivosti na web stranici korisnika prije nego što to učine napadači. Dostupna su različita besplatna rješenja, kao i usluge koje se plaćaju na internetu [8].
- **Sigurnosni dodaci:** Dostupni su brojni sigurnosni dodaci za CMS platforme koji nude značajke poput vatrozida, skeniranja zlonamjernog softvera i zaštite od prijave.
- **Cybersecurity Mesh Framework:** Jedan od nedavnih napredaka u kibernetičkoj sigurnosti, posebno primjenjiv na CMS platforme, je Cybersecurity Mesh Framework. Ovaj okvir decentralizira sigurnosne kontrole preko čvorova, omogućujući prilagodljiviju i otporniju obranu. Studija iz 2024. naglašava da mreža kibernetičke sigurnosti omogućuje CMS sustavima skaliranje sigurnosti kako rastu, dinamički se prilagođavajući promjenjivim prijetnjama. Integracija ovog okvira u sigurnost CMS-a pružila bi jaču obranu od distribuiranih napada, posebno onih koji ciljaju na velike CMS implementacije. [9]
- **Napredni sigurnosni alati utemeljeni na umjetnoj inteligenciji:** Korištenje sigurnosnih alata na temelju umjetne inteligencije još je jedan inovativan pristup. Ovi alati analiziraju obrasce ponašanja korisnika, prometne anomalije i zapisnike sustava u stvarnom vremenu, identificirajući potencijalne prijetnje prije nego što mogu iskoristiti ranjivosti. Nedavni napredak u analizi prometa temeljenoj na strojnom učenju omogućio je preventivno prepoznavanje sumnjivih aktivnosti na CMS platformama, omogućujući brže vrijeme odgovora i umanjujući rizike povezane s DDoS napadima, Brute-force napadima i ubacivanjem SQL-a. [10]

Unatoč ovim mjerama, mnogi korisnici ne uspijevaju implementirati najbolju sigurnosnu praksu, ostavljajući svoje stranice ranjivima na napade. Osim toga, brzi razvoj cyber prijetnji znači da se postojeća rješenja moraju stalno prilagođavati kako bi se pozabavila novim ranjivostima.

3. Metode i tehnike rada

U ovom radu korištene su različite metode i tehnike kako bi cilj razvoja robusnog sigurnosnog okvira bio ostvaren. Budući da je bilo potrebno provesti analizu nad pravom CMS platformom odabran je WordPress, razlog tomu je taj što je WordPress najrasprostranjenija CMS platforma

na internetu. Rad je podijeljen u nekoliko ključnih faza: postavljanje CMS platforme, prikupljanje podataka i analiza CMS platforme, identifikacija sigurnosnih prijetnji i penetracijsko testiranje CMS platforme.

3.1. Postavljanje CMS platforme

Kako bi se izbjegle sve nelegalne radnje potrebno je postaviti vlastitu CMS platformu te postoje dva načina. Prvi način je korištenje usluga web hostinga i kupovanja web domene, a drugi način je postavljanje mreže lokalno koji je korišten u ovom radu. Za podizanje stranice prvo je potrebno skinuti XAMPP. XAMPP je besplatan i višeplatformski paket otvorenog koda rješenja web poslužitelja koji su razvili Apache Friends koji se uglavnom sastoji od Apache HTTP poslužitelja, baze podataka MariaDB i tumača za skripte napisane u programskim jezicima PHP i Perl [11]. Potrebno je postaviti ranjivu CMS platformu i iz tog razloga odabrana je starija verzija XAMPP-a, odnosno verzija 7.4.33 koja sadrži 7.4. verziju PHP-a u sebi.

Nakon skidanja XAMPP-a preostalo je skinuti i samu CMS platformu. Zbog potreba ranjive CMS platforme skinut je WordPress s verzijom 5.1.19.

Prvo su pokrenuti Apache i MySQL s XAMPP kontrolne ploče i u SQL bazu dodana je tablica pod nazivom „wordpress“. Nakon toga ugašeni su Apache i MySQL te je dodan WordPress direktorij u „htdocs“ datoteku unutar xampp direktorija. Zatim su ponovno pokrenuti Apache i MySQL te praćenjem koraka prikazanih na ekranu uspješno je pokrenuta WordPress stranica. Pri koracima postavljanja WordPress-a stavljeno korisničko ime i lozinka je admin/admin kako bi stranica bila još ranjivija. Postavljeno je nekoliko „zastava“ u WordPress direktoriju za napadača da ih pronađe.

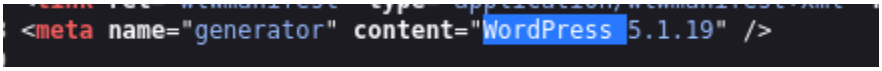
Kako bi CMS platformu napravili ranjivom potrebno je dodati i dodatke koji su ključan dio svake platforme. Dodane su starije verzije dodataka Mail Masta (1.0), Photo Gallery (1.5.34.) i Review Slider (6.1.) te je dodan Easy Updates Manager dodatak kako bi se onemogućila automatska ažuriranja dodataka i tema kao i verzije samog WordPress-a. Na kraju dodano je još nekoliko korisnika s različitim ulogama, to je postignuto dodavanjem korisnika s korisničkim akreditacijama sarah123/sunshine1 i johndoe/johndoe.

3.2. Prikupljanje podataka i analiza CMS platforme

Kako bi se prikupili podaci i analizirala postavljena CMS platforma, provodi se enumeracija WordPress stranice. Enumeracija stranice odnosi se na postupak prikupljanja informacija o različitim aspektima WordPress instalacije, gdje se prikupljaju informacije o korisnicima, temama,

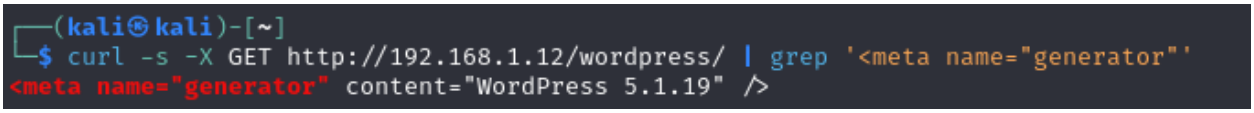
dodatcima i verziji WordPressa [12]. Enumeracija se može provoditi ručno i preko različitih alata koji nam mnogo olakšavaju posao. Osim enumeracije provedena je analiza portova i njihovih ranjivosti koristeći alat Nmap te isto tako je provedena i analiza mašine na kojoj je postavljena CMS platforma koristeći alat OpenVAS koji služi za procjenu sveukupne ranjivosti sustava.

Prvo je enumeracija provedena ručno tako što je pristupljeno početnoj stranici WordPress-a, a pritiskom desnog klika miša odabrana je opcija „View Page Source“. U prikazanom izvoru stranice otkriveno je da se radi o verziji WordPress-a 5.1.19, kao što je prikazano na slici 1.



Slika 1. Verzija WordPress-a [Izvor: vlastita izrada]

Također se može koristiti komandna liniju s curl i grep komandama, kao što je prikazano na slici ispod. Ovim se putem pomoću curl-a vrši GET zahtjev na određenu WordPress stranicu, a zatim se uz pomoć grep-a pretražuje HTML kod kako bi se pronašla verzija WordPress-a, što je u ovom slučaju 5.1.19.



Slika 2. Verzija WordPress-a preko komandne linije [Izvor: vlastita izrada]

Nakon enumeracije verzije, preostalo je izvršiti enumeraciju dodataka i tema koje se koriste. Oni se mogu enumerirati na isti način kao i verzija WordPress-a, putem opcije „View Page Source“ ili komandne linije. Enumeracija je provedena preko komandne linije jer je preglednija i brža.

Slika 3. Enumeriranje dodataka [Izvor: vlastita izrada]

Može se primijetiti da se koriste dodatci photo-gallery, mail-masta i wp-google-places-review-slider, koji su upravo oni dodaci koji su prethodno instalirani.

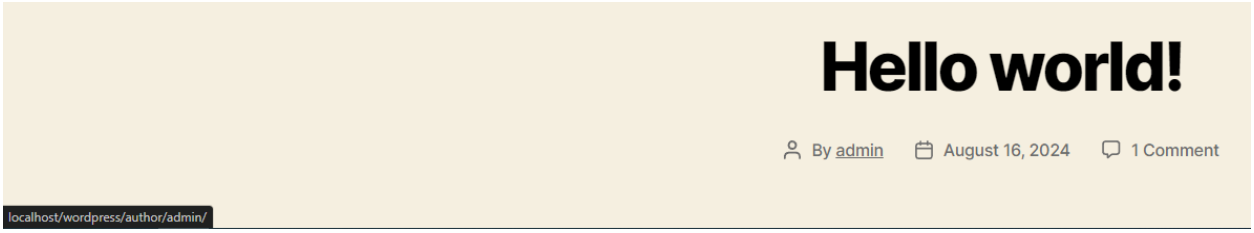
Nakon toga provedena je enumeracija tema kao što je prikazano na slici ispod.

```
┌──(kali㉿kali)-[~]
└─$ curl -s GET http://192.168.1.12/wordpress/ | sed 's/href=/\n/g' | sed 's/src=/\n/g' | grep 'themes' | cut -d"'" -f2
http://localhost/wordpress/wp-content/themes/twentytwenty/style.css?ver=2.7
http://localhost/wordpress/wp-content/themes/twentytwenty/assets/css/font-inter.css?ver=2.7
http://localhost/wordpress/wp-content/themes/twentytwenty/print.css?ver=2.7
http://localhost/wordpress/wp-content/themes/twentytwenty/assets/js/index.js?ver=2.7
```

Slika 4. Enumeriranje tema [Izvor: vlastita izrada]

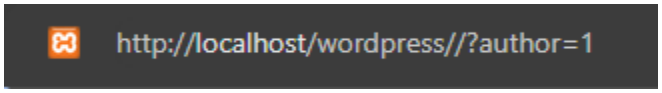
Može se primijetiti da je u upotrebi tema twentytwenty.

Zadnja stavka je enumeracija korisnika, za što postoji više metoda. Prva metoda uključuje pregled objava kako bi se otkrio ID dodijeljen korisniku i njegovo odgovarajuće korisničko ime. Kada se pokazivač postavi iznad veze autora posta pod naslovom „admin“, kao što je prikazano na slici ispod, u donjem lijevom kutu web preglednika pojavljuje se poveznica na korisnički račun.

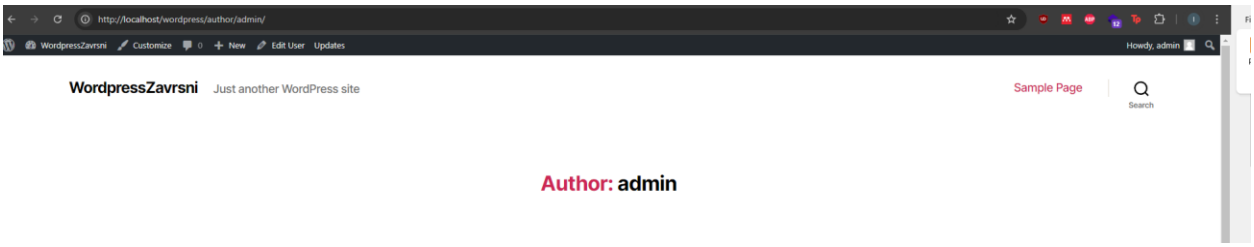


Slika 5. Poveznica na korisnički račun [Izvor: vlastita izrada]

Korisniku admin obično se dodjeljuje korisnički ID 1. To se može potvrditi navođenjem korisničkog ID-a kao parametra autora u URL-u.



Slika 6. Korisnički ID kao parametar autora [Izvor: vlastita izrada]



Slika 7. Potvrda autora s ID-em 1 [Izvor: vlastita izrada]

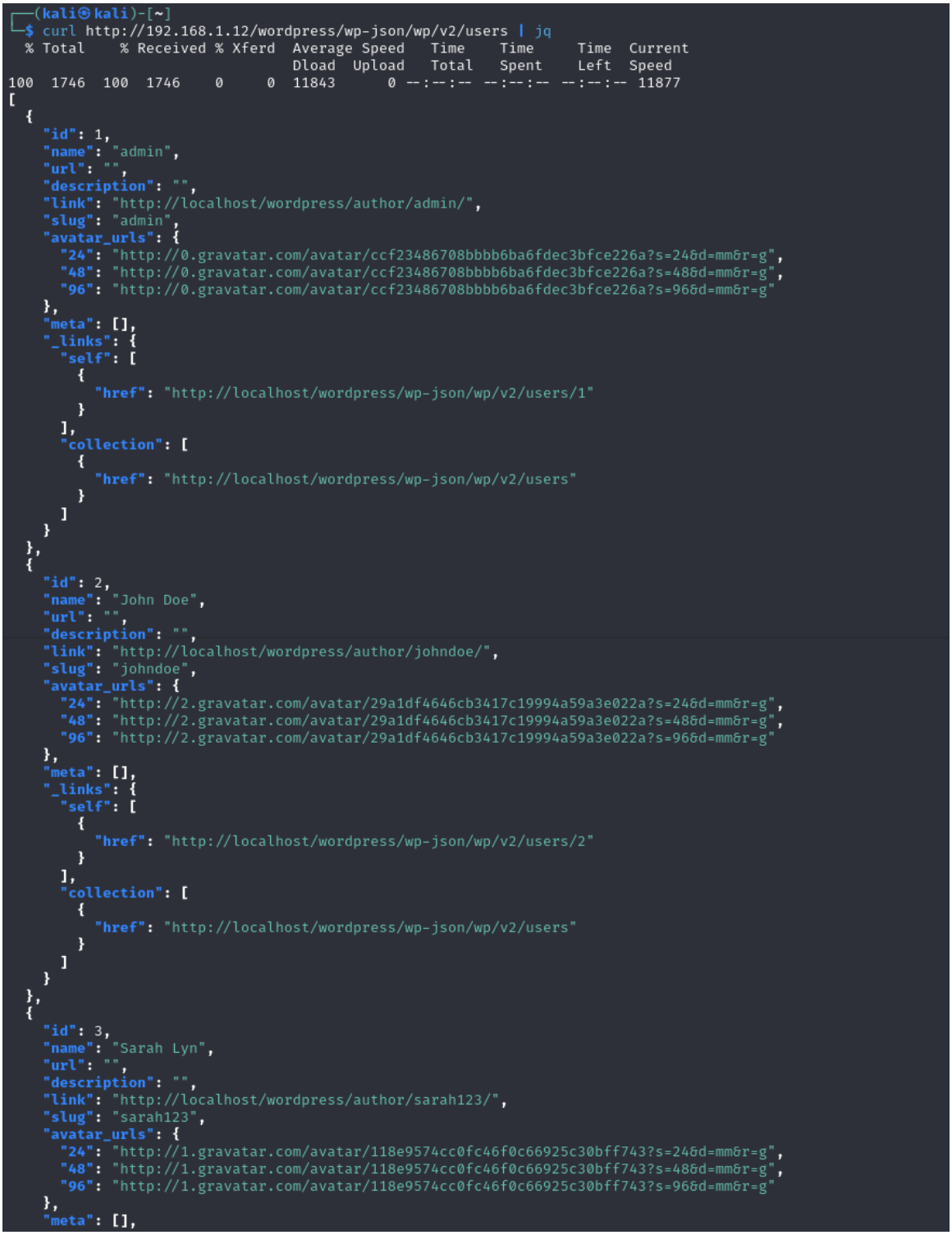
To se također može učiniti s curl komandom iz naredbenog retka. HTTP odgovor u donjem izlazu prikazuje odgovor „200 OK“ ako postoji autor s tim ID-em, dok će prikazati odgovor „400 Not Found“ ako ne postoji.

```
┌──(kali㉿kali)-[~]
└─$ curl -s -I -X GET http://192.168.1.12/wordpress/?author=1
HTTP/1.1 200 OK
Date: Fri, 16 Aug 2024 20:58:46 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/7.4.30
X-Powered-By: PHP/7.4.30
Link: <http://localhost/wordpress/wp-json/>; rel="https://api.w.org/"
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8


┌──(kali㉿kali)-[~]
└─$ curl -s -I -X GET http://192.168.1.12/wordpress/?author=2
HTTP/1.1 200 OK
Date: Fri, 16 Aug 2024 20:59:30 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/7.4.30
X-Powered-By: PHP/7.4.30
Link: <http://localhost/wordpress/wp-json/>; rel="https://api.w.org/"
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8


┌──(kali㉿kali)-[~]
└─$ curl -s -I -X GET http://192.168.1.12/wordpress/?author=3
HTTP/1.1 200 OK
Date: Fri, 16 Aug 2024 20:59:33 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/7.4.30
X-Powered-By: PHP/7.4.30
Link: <http://localhost/wordpress/wp-json/>; rel="https://api.w.org/"
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8


┌──(kali㉿kali)-[~]
└─$ curl -s -I -X GET http://192.168.1.12/wordpress/?author=4
HTTP/1.1 404 Not Found
Date: Fri, 16 Aug 2024 20:59:36 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/7.4.30
X-Powered-By: PHP/7.4.30
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Link: <http://localhost/wordpress/wp-json/>; rel="https://api.w.org/"
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

Slika 8. Prikaz HTTP odgovora o ID-u korisnika [Izvor: vlastita izrada]

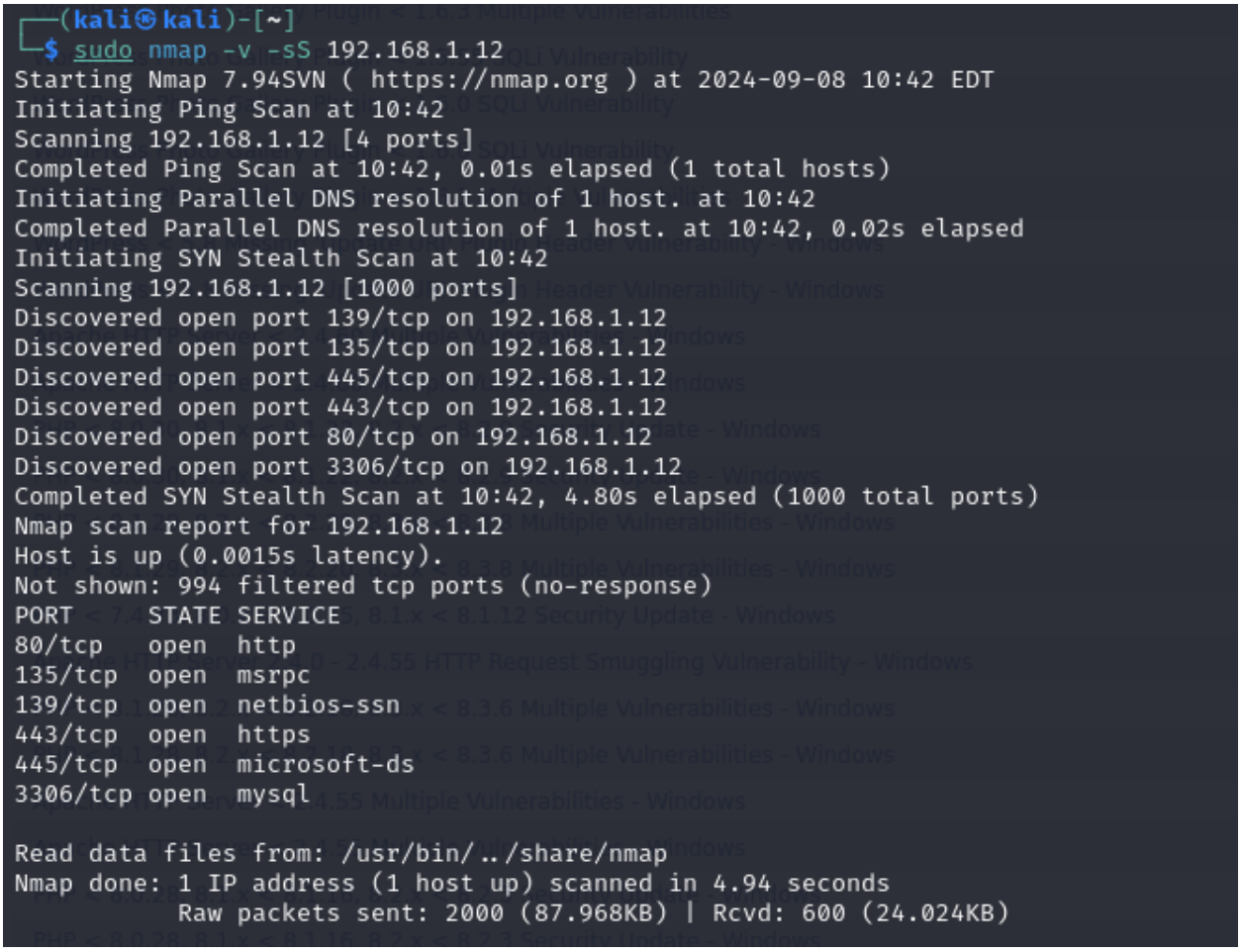
Prema gornjoj slici može se vidjeti da postoje tri korisnika, od kojih je jedan sigurno admin korisnik.

Druga metoda zahtijeva interakciju s JSON krajnjom točkom, što omogućuje dobivanje popisa korisnika.



Slika 9. Prikaz korisnika u JSON formatu [Izvor: vlastita izrada]

Poslije provedene enumeracije napravljeno je skeniranje otvorenih portova pomoću alata Nmap, točnije prvo smo proveli skeniranje otvorenih portova.

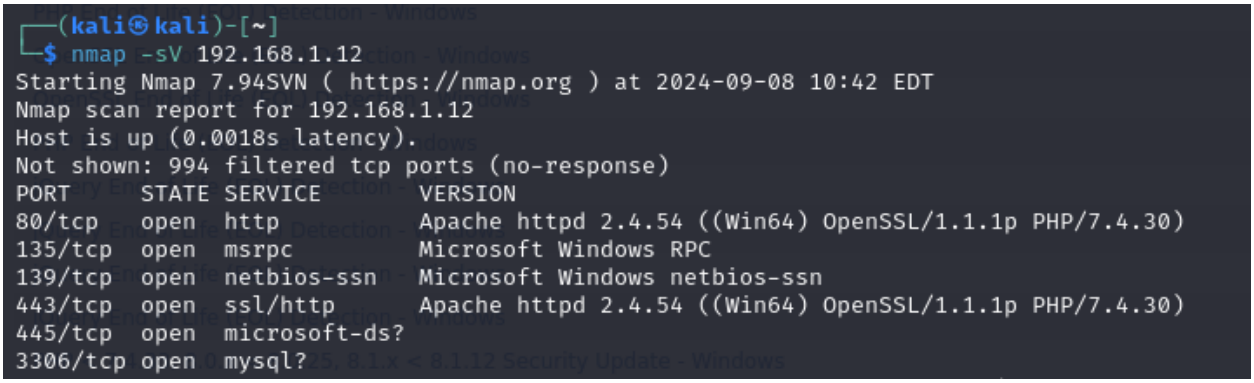


Slika 10. Rezultat Nmap-a [Izvor: vlastita izrada]

Pomoću dobivenih rezultata utvrđeno je da postoji šest otvorenih portova:

- 80/tcp (http): Ovo je standardni priključak za posluživanje web stranica putem HTTP-a Budući da se radi o WordPress stranici, ovaj priključak služi stranici bez enkripcije.
- 135/tcp (msrpc): Ovaj priključak koristi usluga Microsoft RPC (Remote Procedure Call), koja je uključena u različite mrežne funkcije, obično u Windows okruženjima.
- 139/tcp (netbios-ssn) i 445/tcp (microsoft-ds): Ovi priključci su povezani s NetBIOS i SMB (Server Message Block) protokolima. SMB se koristi za dijeljenje datoteka na Windows mrežama, što može biti ranjivo ako nije pravilno osigurano.
- 443/tcp (https): Koristi se za kriptirani HTTPS promet. Ako WordPress web stranica podržava HTTPS, ovaj priključak omogućava sigurnu komunikaciju s web mjestom.
- 3306/tcp (mysql): Ovo je zadani port za MySQL bazu podataka, koju WordPress koristi za pohranu podataka. Otvoren MySQL port može predstavljati sigurnosni rizik ako nije zaštićen odgovarajućim kontrolama pristupa, poput jakih akreditacija i vatrozida.

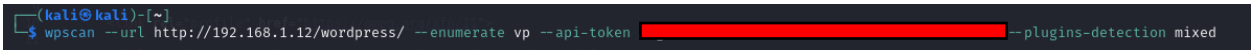
Poslije skeniranja otvorenih portova potrebno je vidjeti verzije servisa na tim portovima što se može vidjeti na slici ispod.



Slika 11. Verzije servisa [Izvor: vlastita izrada]

3.3. Identifikacija sigurnosnih prijetnji

Kako je enumeracija provedena ručno, također se može provesti uz pomoć alata kako bi se dobilo više informacija i otkrile potencijalne ranjivosti. Za tu svrhu odabran je alat WPScan, koji služi za enumeraciju WordPress stranica.



Slika 12. Komanda za enumeraciju u WPScan-u [Izvor: vlastita izrada]

Unosom komande prikazane na slici 10 dobivena je lista ranjivih dodataka, zajedno s njihovim referencama, opisom ranjivosti i informacijama o verziji u kojoj su ranjivosti popravljene.

Slika 13. Popis pronađenih ranjivosti [Izvor: vlastita izrada]

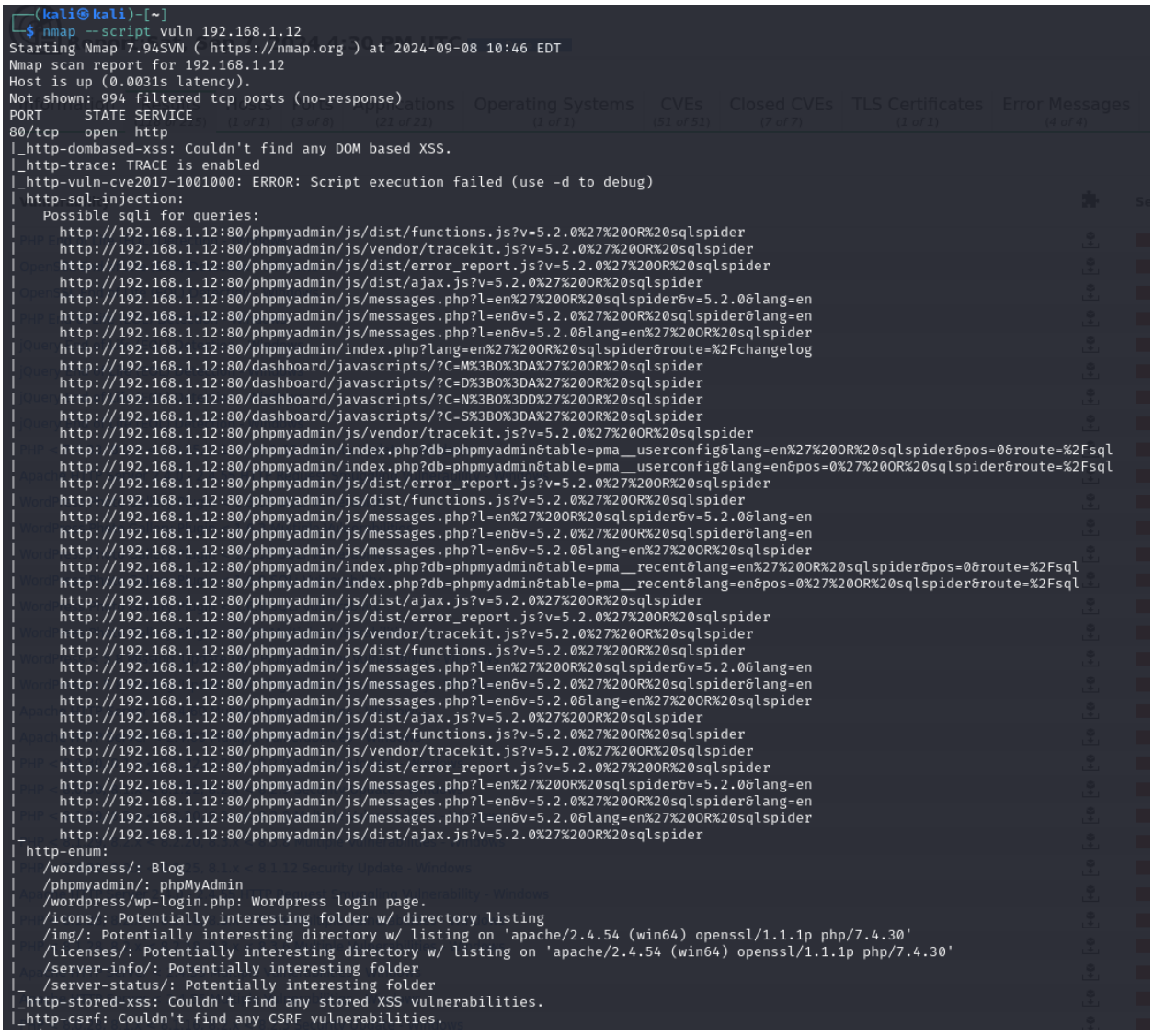
Prvu ranjivost koja je prepoznata je omogućeno indeksiranje direktorija. Indeksiranje direktorija sigurnosni je problem pri kojem web poslužitelj nenamjerno izlaže popis direktorija korisnicima [13]. Ovu ranjivost napadač može iskoristiti za pretragu datoteka unutar direktorija i eventualno manipuliranje njima.

Također, prema rezultatu skeniranja, vidljivo je da platforma ima uključen XML-RPC. Datoteka xmlrpc.php u WordPressu olakšava pozive udaljenih procedura (RPC) pomoću XML-a. U biti djeluje kao most između WordPress stranice i vanjskih aplikacija, dopuštajući međusobnu komunikaciju [14]. XML-RPC nudi 80 različitih metoda koje se mogu pozvati, a neki od napada koji se mogu izvršiti preko uključuju DDoS i Brute-force napade.

Nadalje, prepoznat je veći broj ranjivosti u dodacima koji se koriste na stranici. Dodatak „Mail Masta“ ima dvije prepoznate ranjivosti, dok dodatak „Photo-gallery“ ima dvadeset pet prepoznatih ranjivosti.

Neke od prepoznatih ranjivosti uključuju Unauthenticated Local File Inclusion (LFI), Remote Code Execution (RCE), SQL Injection, Cross-Site Scripting (XSS). Posljedice ovih napada već su ranije objašnjene.

Pomoću Nmap-a provedeno je potpuno skeniranje ranjivosti.



Slika 14. Ranjivosti dobivene Nmap-om [Izvor: vlastita izrada]

```
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
| ssl-dh-params:
|   VULNERABLE:
|   Diffie-Hellman Key Exchange Insufficient Group Strength
|     State: VULNERABLE
|       Transport Layer Security (TLS) services that use Diffie-Hellman groups
|       of insufficient strength, especially those using one of a few commonly
|       shared groups, may be susceptible to passive eavesdropping attacks.
|     Check results:
|       WEAK DH GROUP 1
|             Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA
|             Modulus Type: Safe prime
|             Modulus Source: RFC2409/Oakley Group 2
|             Modulus Length: 1024
|             Generator Length: 8
|             Public Key Length: 1024
|     References:
|_      https://weakdh.org
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-trace: TRACE is enabled
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|   /wordpress/: Blog
|   /phpmyadmin/: phpMyAdmin
|   /wordpress/wp-login.php: Wordpress login page.
|   /icons/: Potentially interesting folder w/ directory listing
|   /img/: Potentially interesting directory w/ listing on 'apache/2.4.54 (win64) openssl/1.1.1p php/7.4.30'
|   /licenses/: Potentially interesting directory w/ listing on 'apache/2.4.54 (win64) openssl/1.1.1p php/7.4.30'
|   /server-info/: Potentially interesting folder
|_  /server-status/: Potentially interesting folder
445/tcp  open  microsoft-ds
3306/tcp open  mysql
|_ssl-poodle: ERROR: Script execution failed (use -d to debug)
|_ssl-dh-params: ERROR: Script execution failed (use -d to debug)
|_mysql-vuln-cve2012-2122: ERROR: Script execution failed (use -d to debug)

Host script results:
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
```

Slika 15. Ranjivosti dobivene Nmap-om [Izvor: vlastita izrada]

Na temelju skeniranja Nmap-a i otkrivenih ranjivosti, evo analize glavnih nalaza:

1. HTTP (Port 80)

- phpMyAdmin datoteke: Skeniranje je pokazalo nekoliko potencijalnih datoteka i URL-ova povezanih s phpMyAdmin. Oni mogu predstavljati sigurnosni rizik ako phpMyAdmin nije ispravno osiguran.
- Metoda TRACE omogućena: Metoda HTTP TRACE je omogućena, što može dovesti do napada Cross-Site Tracing (XST). Ovo se smatra ranjivošću u modernim web poslužiteljima.
- Popisi direktorija: Nekoliko direktorija je javno dostupno (/img/, /icons/, /server-info/, /server-status/). Popis direktorija izlaže sadržaj tih direktorija napadačima i može otkriti osjetljive informacije o sustavu, što napadačima može pomoći da iskoriste ranjivosti.

2. SSL/TLS Ranjivosti (Port 443 – HTTPS)

- Diffie-Hellman slaba razmjena ključeva: Skeniranje prijavljuje ranjivost u SSL/TLS konfiguraciji koja se odnosi na slabe parametre Diffie-Hellmanove razmjene ključeva (slaba DH grupa 1). Korištenje slabih parametara razmjene ključeva može učiniti SSL/TLS vezu ranjivom na napade poput napada Logjam.

3. SQL injection

- Skeniranje je označilo potencijalne točke ubacivanja SQL-a u phpMyAdmin URL-ove. SQL injekcija omogućuje napadačima manipuliranje SQL upitima, što im može omogućiti pristup bazi podataka.

4. MSRPC (Port 135)

- Ovaj port koristi Microsoftova usluga Remote Procedure Call (RPC). Iako nije spomenuta nikakva specifična ranjivost, izlaganje ovog priključka može povećati površinu napada, posebno za Windows hostove.

5. NetBIOS (Port 139)

- NetBIOS preko TCP-a je stariji mrežni protokol koji može izložiti usluge dijeljenja datoteka, što ih čini ranjivima na različite napade, uključujući napade temeljene na SMB-u.

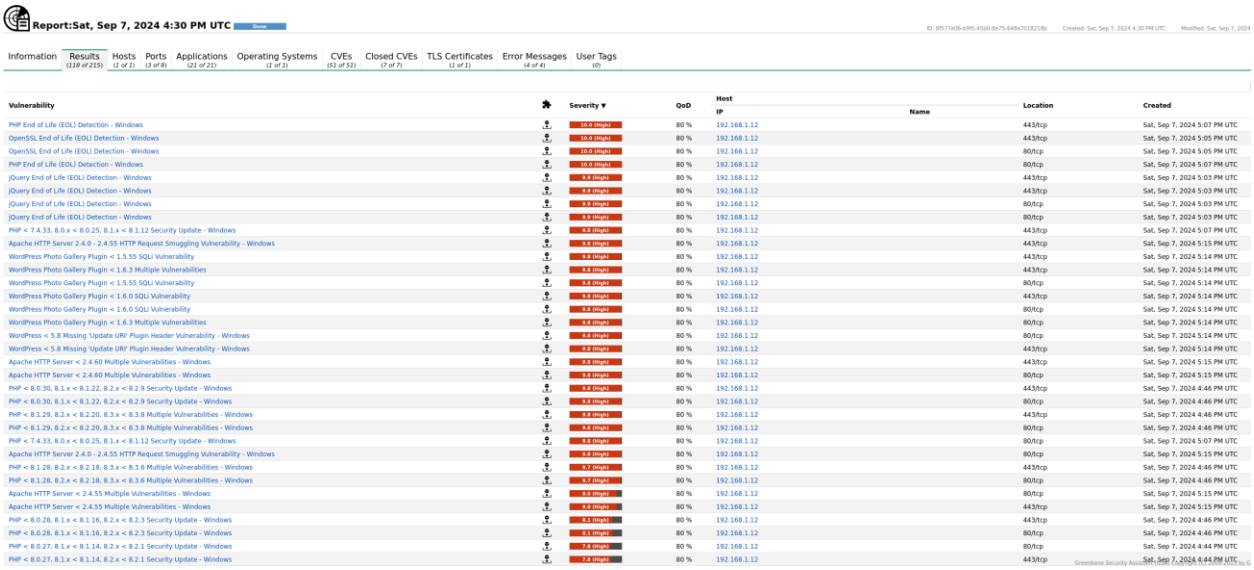
6. MySQL (Port 3306)

- mysql-vuln-cve2012-2122: Skeniranje je pokušalo provjeriti ovu specifičnu ranjivost, ali nije uspjelo u potpunosti izvršiti skriptu. Međutim, CVE-2012-2122 predstavlja ranjivost zaobilaženja autentifikacije MySQL-a koja može omogućiti napadaču da zaobiđe autentifikaciju korištenjem slabih akreditacija.

7. Samba (Port 445)

- Skeniranje je pokušalo testirati SMB ranjivosti (poput CVE-2012-1182 i MS16-061), ali nije uspjelo uspostaviti vezu.

Poslije provedenog mrežnog skeniranja provedeno je sveobuhvatno skeniranje ranjivosti sustava pomoću alata OpenVAS.



Slika 16. Izvješće OpenVAS-a [Izvor: vlastita izrada]

Izvješće o skeniranju OpenVAS-a identificiralo je više ranjivosti visoke i srednje težine na sustavu, uključujući zastarjele verzije OpenSSL-a, PHP-a, jQueryja, WordPress dodataka i Apache HTTP poslužitelja.

Ranjivosti visoke razine:

- Zastarjeli OpenSSL (verzija 1.1.1p): Instalirana verzija OpenSSL-a dosegla je kraj života (EOL) i više ne prima sigurnosna ažuriranja.
- Zastarjeli PHP (verzija 7.4.30): Ova verzija dosegla je kraj života i ima višestruke sigurnosne propuste, uključujući SQL injection.
- Zastarjeli jQuery (verzija 1.10.2): Verzija jQueryja koja se koristi dosegla je kraj života i mogla bi imati nezakrpane sigurnosne propuste.
- Ranjivosti dodataka za WordPress (dodatak za galeriju fotografija): U zastarjeloj verziji dodatka Photo Gallery pronađeno je više ranjivosti SQL injection.
- Apache HTTP poslužitelj (verzija 2.4.54): Ova verzija Apachea ima nekoliko ranjivosti uključujući uskraćivanje usluge (DoS) i probleme s krijumčarenjem zahtjeva.
- WordPress XSS ranjivost: Ranjivost cross-site scripting (XSS) u WordPressu omogućuje napadačima da ubace proizvoljne web skripte.

Ranjivosti srednje razine:

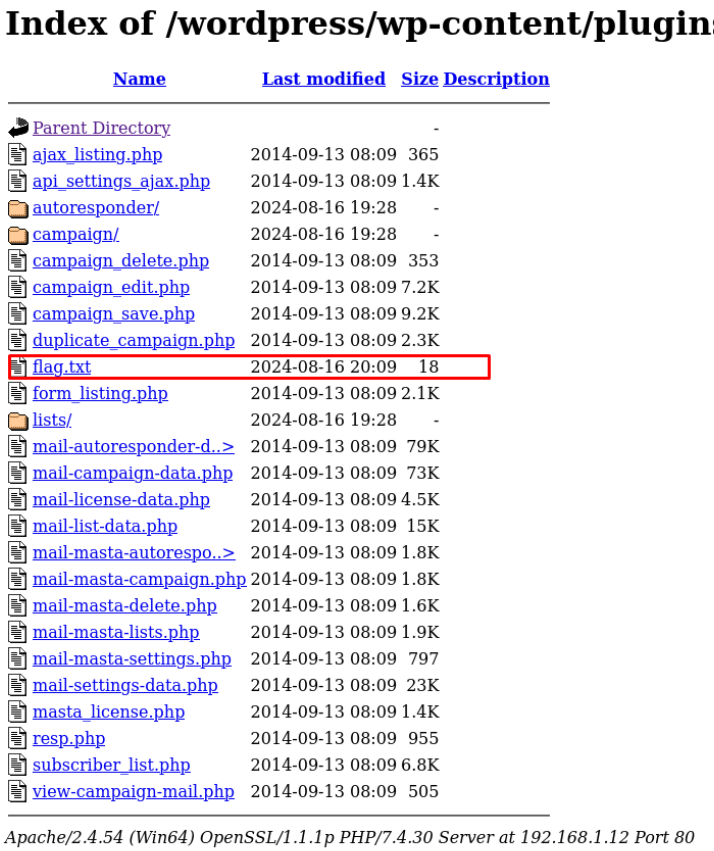
- Krijumčarenje zahtjeva Apache HTTP poslužitelja i HTTP/2 DoS: Poslužitelj je ranjiv na krijumčarenje HTTP zahtjeva i napade uskraćivanjem usluge (DoS) putem HTTP/2 protokola.
- Slaba SSL/TLS enkripcija (slaba Diffie-Hellman): Koriste se slabe Diffie-Hellman grupe koje su podložne napadima.
- Enumeracija DCE/RPC i MSRPC usluga: DCE/RPC ili MSRPC usluge na glavnom računalu mogu se enumerirati.
- Dostupan Apache HTTP poslužitelj /server-info: Dostupna je stranica s informacijama o poslužitelju, koja pruža detaljne informacije o konfiguraciji koje napadači mogu koristiti.

3.4. Penetracijsko testiranje CMS platforme

Prema provedenoj enumeraciji dobiveno je mnogo informacija o ranjivostima na platformi koje se mogu iskoristiti, što je i učinjeno. Prvo je provedeno indeksiranje direktorija.

Kako su prilikom postavljanje WordPress-a u direktorij postavljene određene zastave, zadatak je bio pronaći ih. Unosom sljedeće adrese http://192.168.1.12/wordpress/wp-content/plugins/mail-masta/ došlo se do situacije gdje je moguće prolaziti kroz direktoriju dodatka „mail masta“ i tražiti zastavu.

Slika 17. Izgled direktorija dodatka „Mail Masta“ [Izvor: vlastita izrada]

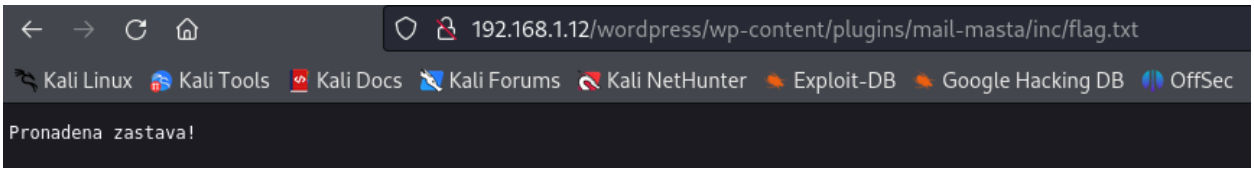


Index of /wordpress/wp-content/plugins/mail-masta/inc

Name	Last modified	Size	Description
Parent Directory		-	
ajax_listing.php	2014-09-13 08:09	365	
api_settings_ajax.php	2014-09-13 08:09	1.4K	
autoresponder/	2024-08-16 19:28	-	
campaign/	2024-08-16 19:28	-	
campaign_delete.php	2014-09-13 08:09	353	
campaign_edit.php	2014-09-13 08:09	7.2K	
campaign_save.php	2014-09-13 08:09	9.2K	
duplicate_campaign.php	2014-09-13 08:09	2.3K	
flag.txt	2024-08-16 20:09	18	
form_listing.php	2014-09-13 08:09	2.1K	
lists/	2024-08-16 19:28	-	
mail-autoresponder-d..>	2014-09-13 08:09	79K	
mail-campaign-data.php	2014-09-13 08:09	73K	
mail-license-data.php	2014-09-13 08:09	4.5K	
mail-list-data.php	2014-09-13 08:09	15K	
mail-masta-autorespo..>	2014-09-13 08:09	1.8K	
mail-masta-campaign.php	2014-09-13 08:09	1.8K	
mail-masta-delete.php	2014-09-13 08:09	1.6K	
mail-masta-lists.php	2014-09-13 08:09	1.9K	
mail-masta-settings.php	2014-09-13 08:09	797	
mail-settings-data.php	2014-09-13 08:09	23K	
masta_license.php	2014-09-13 08:09	1.4K	
resp.php	2014-09-13 08:09	955	
subscriber_list.php	2014-09-13 08:09	6.8K	
view-campaign-mail.php	2014-09-13 08:09	505	

Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/7.4.30 Server at 192.168.1.12 Port 80

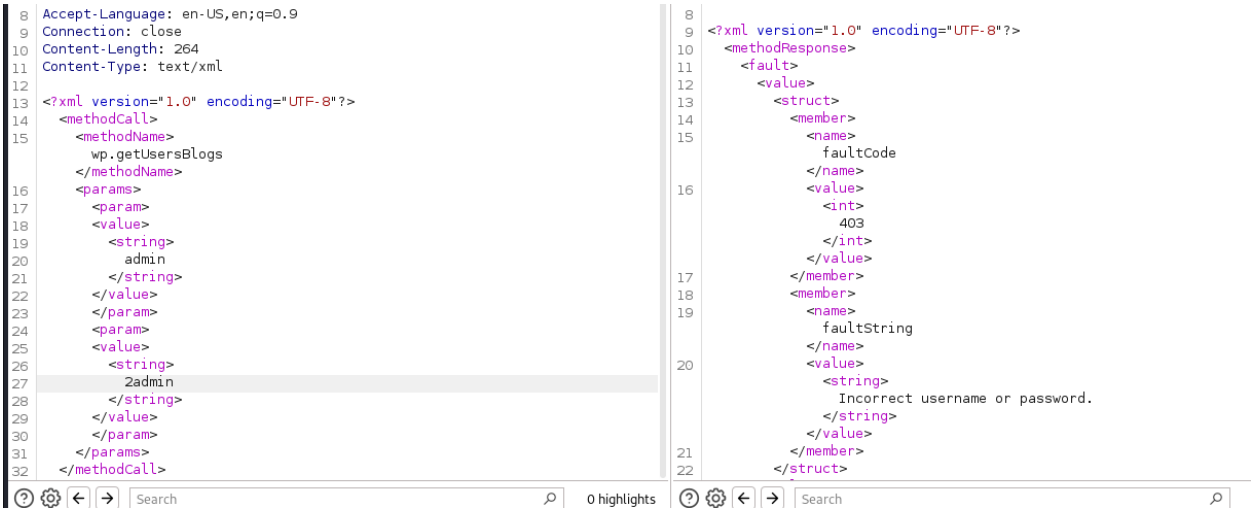
Slika 18. Pronađena zastava u direktoriju [Izvor: vlastita izrada]



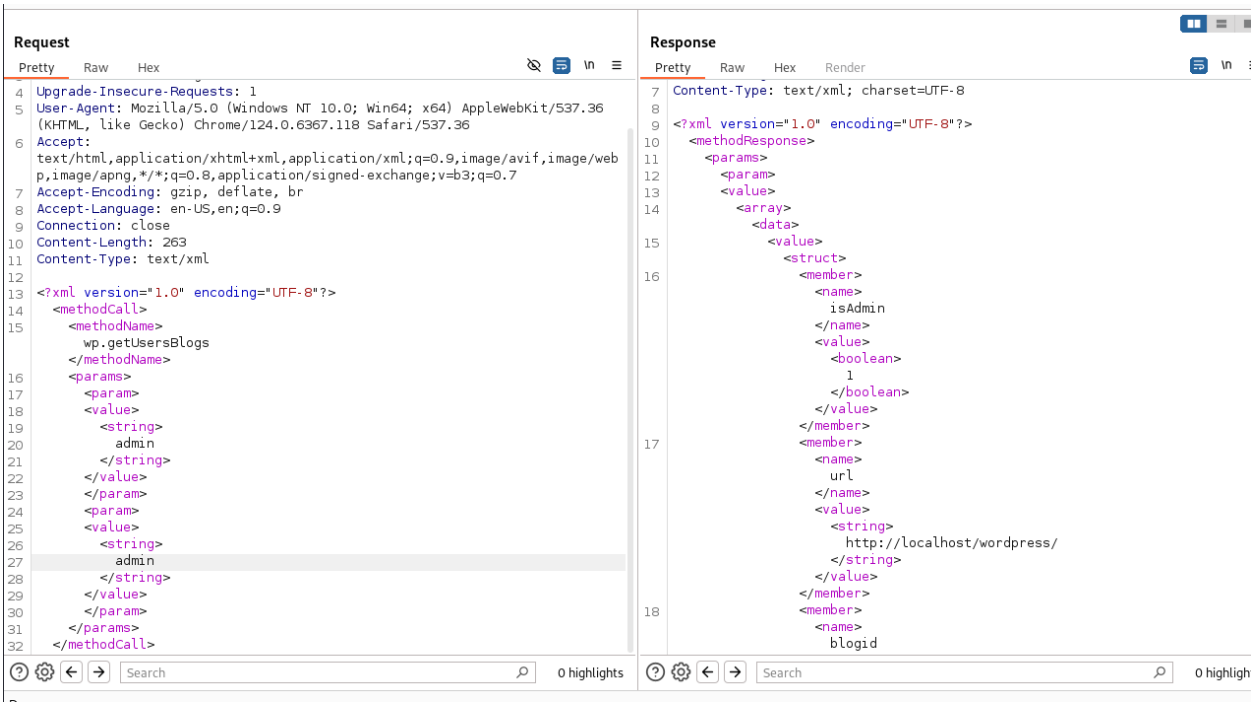
Slika 19. Sadržaj flag.txt datoteke [Izvor: vlastita izrada]

Za indeksiranje direktorija potrebno je poznavanje moguće strukture direktorija kako bi se lakše i brže snalazilo u pretraživanju direktorija.

Nakon indeksiranja direktorija, odlučeno je izvršiti DDoS i Brute-force napad preko XML-RPC-a koristeći alat Burp. Pokušaj DDoS napada izvršen je putem metode pingback.ping, no napad nije bio uspješan jer je stranica podignuta lokalno, pa nije bilo moguće preplaviti je zahtjevima s druge stranice. Metoda pingback.ping prima dva parametra: prvi je ciljana stranica, a drugi je stranica s koje se šalju zahtjevi. Brute-force napad proveden je korištenjem metode wp.GetUsersBlogs, koja zahtjeva korisničko ime i lozinku. Kada je kombinacija točna metoda vraća sve objave korisnika na blogu, a u suprotnom vraća odgovor „403“.



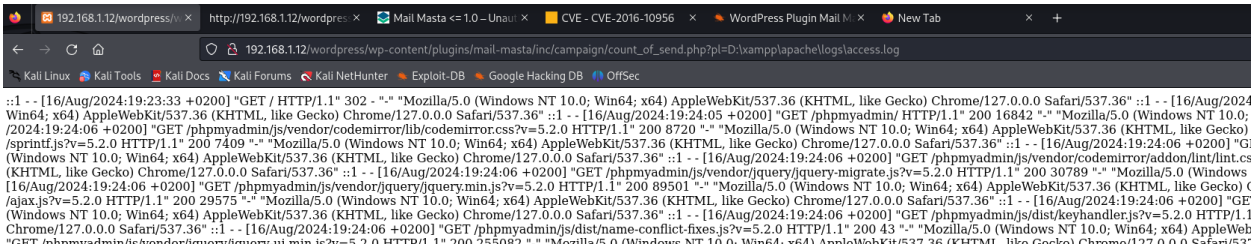
Slika 20. Netočna kombinacija korisničkog imena i lozinke [Izvor: vlastita izrada]



Slika 21. Točna kombinacija korisničkog imena i lozinke [Izvor: vlastita izrada]

Kako je prepoznata mogućnost iskorištavanja Unauthenticated Local File Inclusion (LFI), kliknuto je na reference kako bi se saznalo više informacija o tome. Prema referencama, ranjivost se nalazi u datotekama unutar dodataka koje sadrže komandu „include($_GET['pl']);“. Ako se server nalazi na Linux operativnom sustavu, moguće je pristupiti putanji „/etc/passwd“, na primjer putem „http://server/wp-content/plugins/mail-masta/inc/campaign/count_of_send.php?pl=/etc/passwd“.

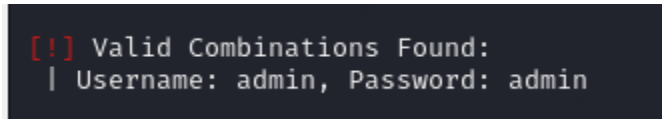
Korištenjem ovih podataka i LFI ranjivosti, uspjelo se pristupiti Apache logovima od WordPress-a, odnosno mjestu gdje je server podignut lokalno, jer je server bio postavljen lokalno na Windows operativnom sustavu.



Slika 22. Apache log [Izvor: vlastita izrada]

Prethodnom enumeracijom korisnika dobivena su njihova korisnička imena, što omogućava alatu WPScan da automatizirano izvrši Brute-force napad na korisnika admin. Unosom sljedeće komande dobivena je kombinacija korisničkog imena i lozinke korisnika admin:

„wpscan --password-attack xmlrpc -t 20 -U admin -P ~/Desktop/rockyou.txt –url http://192.168.1.12/wordpress/“

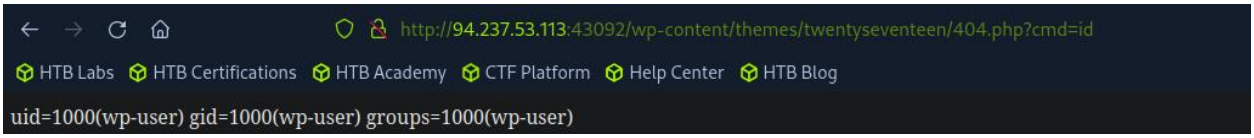


Slika 23. Rezultat Brute-force napada [Izvor: vlastita izrada]

Za pronalazak lozinke korištena je poznata rockyou.txt datoteka koja sadrži mnoge poznate lozinke. Jedini problem s Brute-force napadom je taj što traje jako dugo, a dužina procesa ovisi o broju jezgri procesora koji se koriste.

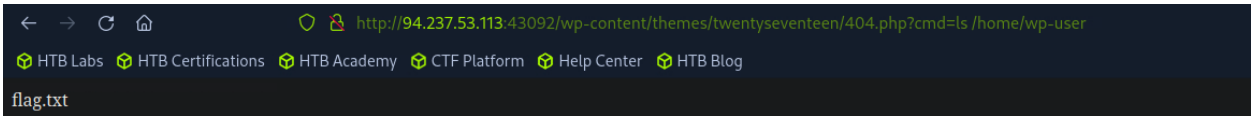
Zadnji napad koji je izveden bio je Remote Code Execution (RCE) napad putem „Theme Editor-a“ na WordPress-u. Nakon što je prethodno dobivena kombinacija korisničkog imena i lozinke admin/admin, osiguran je administrativni pristup stranici. Uz administrativni pristup WordPressu omogućeno je modificiranje PHP izvornog koda za izvršavanje sistemskih naredbi. Za izvođenje ovog napada, prijavljeno je na WordPress s administratorskim akreditacijama, a zatim je na bočnoj ploči odabran „Apperance“ i potom „Theme Editor“. Odabrana je neaktivna tema kako bi se izbjeglo oštećenje glavne teme, te je pod „Theme files“ odabrana datoteka „404 Template“ i u nju dodan kod „system($_GET['cmd']);“. Ovaj kod omogućio je izvršavanje naredbi putem GET parametra „cmd“.

RCE je potvrđen unosom URL-a u web preglednik ili izdavanjem curl zahtjeva.

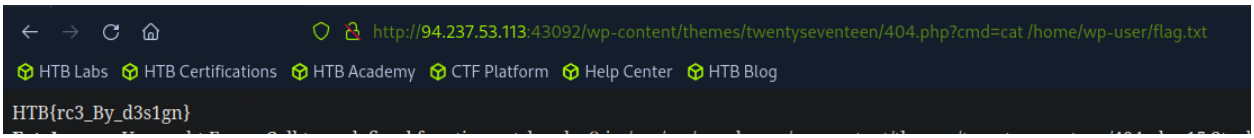


Slika 24. Dokaz RCE-a [Izvor: vlastita izrada]

Kako su prethodno prilikom postavljanja WordPress platforme postavljene određene zastave, sada je zadatak pronaći i drugu zastavu koja je postavljena.



Slika 25. Pronađena zastavica preko RCE-a [Izvor: vlastita izrada]



Slika 26. Sadržaj zastavice [Izvor: vlastita izrada]

4. Sigurnosni okvir

Nakon provedene analize i penetracijskog testiranja, preostalo je razviti sigurnosni okvir. Sigurnosni okvir predstavlja skup dobro dokumentiranih standarda, politika, postupaka i najboljih praksi, koji su namijenjeni jačanju sigurnosnog položaja organizacije i smanjenju rizika [15].

4.1. Postavljanje CMS platforme

- **Odabir prave hosting tvrtke:** Hosting tvrtka je poznata kao temelj za bilo koje web mjesto, stoga je potrebno pronaći pouzdanu hosting tvrtku, mnoge tvrtke pružaju dodatne slojeve sigurnosti. Dakle, odabir pravog hosta može spasiti web mjesto od različitih vrsta napada [16, str. 39].
- **Korištenje SSL certifikata:** SSL certifikati također su namijenjeni šifriranju svih osjetljivih podataka poput brojeva kreditnih kartica ili lozinki. Korištenjem SSL certifikata, URL web stranice će se promijeniti iz HTTP na HTTPS; S označava da je sigurno („secure“). Provjerava autentičnost i kriptira podatke koji se putem web stranice prenose na poslužitelj. Stoga, ako se korisnik prijavi na stranicu ili pošalje osjetljive podatke, podaci se ne mogu ukrasti. Značajke SLL-a štite stranicu od prijevare, stječu povjerenje

posjetitelja prikazivanjem sigurnosnog lokota, onemogućuju promjenu i uređivanje podataka i štite osobne podatke posjetitelja od zlouporabe [16, str. 40].

- **Instalacija najnovijih verzija**: Kako bi platforma imala najnovije sigurnosne zakrpe, potrebno je instalirati najnoviju verziju platforme te koristiti najnovije verzije dodataka i tema.
- **Jedinstveni prefiksi baze podataka:** Prema zadanim postavkama, mnoge CMS instalacije koriste predvidljive prefikse tablica, što ih može učiniti ranjivima na napade SQL injekcijom. Promjena zadanog prefiksa (npr. wp_ za WordPress) u nešto jedinstveno može ublažiti ovaj rizik. [17]

4.2. Zaštita web poslužitelja

- **Onemogućavanje nepotrebnih usluga:** Treba provjeriti rade li samo bitne usluge na web poslužitelju, što uključuje onemogućavanje svih zadanih usluga, modula ili priključaka koji se ne koriste. Na primjer, ako FTP ili SSH nisu potrebni, trebali bi biti onemogućeni. Alati poput netstata i ps mogu se koristiti za provjeru servisa koji rade.
- **Onemogućiti nepotrebne HTTP metode:** Potrebno je provjeriti jesu li onemogućene HTTP metode poput TRACE, PUT i DELETE, osim ako nisu izričito potrebne, jer se mogu iskoristiti za napade poput skriptiranja na više stranica (XSS) i drugih napada. Na primjer, onemogućavanje TRACE metode u Apacheu ili Nginxu sprječava napadače da koriste napade praćenja između web-mjesta. To se može učiniti putem konfiguracijskih datoteka poput httpd.conf ili nginx.conf. [18]
- **Ograničiti pristup putem IP adrese:** Potrebno je ograničiti pristup osjetljivim direktorijima, poput administracijske nadzorne ploče ili osjetljivih resursa, na temelju IP adresa. Bez obzira na to koristi li se Apache, Nginx ili IIS, pravila kontrole pristupa mogu se konfigurirati na temelju IP adrese (npr. na Nginx: location /admin { allow 192.168.1.100; deny all; }). [18]
- **Koristiti balansere opterećenja i obrnute proxyje:** Potrebno je koristiti balansere opterećenja ili obrnute proxyje, poput Nginx-a, HAProxy-a ili Cloudflare-a, kako bi se sakrili pozadinski poslužitelji, riješili skokovi prometa i filtrirali dolazni zahtjevi. Obrnuti proxyji također mogu implementirati predmemoriju za ublažavanje DDoS napada i poboljšanje performansi.
- **Ograničiti učitavanje datoteka:** Ako web aplikacija dopušta učitavanje datoteka, potrebno je osigurati da se datoteke temeljito provjeravaju i da su dopuštene samo određene vrste datoteka, poput slika. Također, potrebno je osigurati da se učitane

datoteke pohranjuju izvan web direktorija kako bi se izbjegao izravan pristup potencijalno zlonamjernim datotekama (npr na Nginx: location /uploads { deny all; }). [18]

- **Ograničiti broj zahtjeva:** Ograničenje broja zahtjeva štiti poslužitelj od Brute-force napada i DDoS pokušaja ograničavanjem broja zahtjeva koje korisnik ili IP adresa mogu uputiti tijekom određenog razdoblja. Potrebno je implementirati ograničenje brzine na razini web poslužitelja (npr. za Nginx: limit_req_zone $binary_remote_addr zone=mylimit:10m rate=5r/s; server { location /login { limit_req zone=mylimit burst=10 nodelay; } }). [19]
- **Bilježenje i praćenje:** Potrebno je omogućiti detaljno bilježenje na web poslužitelju kako bi se pratili obrasci pristupa i prepoznale zlonamjerne aktivnosti. Zapisci pristupa (Access logs) trebaju bilježiti svaki zahtjev upućen poslužitelju, uključujući vrijeme zahtjeva, IP adresu klijenta i traženi URL. Zapisi pogrešaka (Error logs) trebaju pratiti neuspjele zahtjeve, uključujući pokušaje neovlaštenog pristupa i pogreške poslužitelja. Zapise je potrebno integrirati s centraliziranim rješenjima za upravljanje zapisima, poput Splunk-a, ELK Stack-a ili Graylog-a, radi praćenja u stvarnom vremenu.

4.3. Opće sigurnosne mjere za baze podataka

- **Onemogućiti daljinski pristup:** Kako bi se smanjila izloženost, potrebno je onemogućiti daljinski pristup poslužiteljima baze podataka, osim ako to nije apsolutno neophodno. To se može konfigurirati ograničavanjem IP adrese na localhost ili korištenjem VPN-a za udaljene veze (npr. za PostgreSQL u postgresql.conf datoteci: listen_addresses = 'localhost'). [20]
- **Omogućiti SSL/TLS šifriranje:** Potrebno je osigurati da su veze s bazom podataka šifrirane pomoću SSL/TLS-a kako bi se spriječili napadi čovjeka u sredini (npr. za PostgreSQL u postgresql.conf datoteci: ssl = on, ssl_cert_file = '/etc/ssl/certs/ssl-cert-snakeoil.pem', ssl_key_file = '/etc/ssl/private/ssl-cert-snakeoil.key').
- **Koristiti pravila jake lozinke:** To podrazumijeva da lozinka ima minimum 12 znakova, zahtjeve složenosti (velika i mala slova, brojevi i posebni znakovi) i pravila isteka lozinke kako bi se osiguralo redovito ažuriranje lozinki.
- **Ograničiti pristup osjetljivim tablicama:** Potrebno je osigurati da osjetljive tablice, poput onih koje sadrže financijske ili osobne podatke, imaju stroža dopuštenja. Također, potrebno je primijeniti sigurnost na razini retka ili stupca, gdje je primjenjivo, kako bi se kontrolirao pristup na temelju osjetljivosti podataka (npr. za PostgreSQL: CREATE

POLICY policy_name ON table_name FOR SELECT TO role_name USING (condition);). [20]

- **Zaštita od SQL injection napada:** Pripremljene izjave pomažu u ublažavanju napada SQL injekcije odvajanjem SQL koda od unosa podataka. Potrebno je osigurati da su svi upiti parametrizirani umjesto korištenja dinamičkog SQL-a (npr. za PostgreSQL: PREPARE stmt_name (datatype) AS, INSERT INTO table_name (column) VALUES ($1);). Također, preporučuje se postaviti vatrozid baze podataka kako bi se analizirali upiti za SQL ubacivanje ili druge nepravilne aktivnosti. DBShield ili GreenSQL mogu se koristiti za presretanje i analizu upita u stvarnom vremenu. Oracle Database Vault ili SQL Server Threat Detection pružaju ugrađene SQL sigurnosne alate za naprednu zaštitu. [21]
- **Šifrirati osjetljive podatke:** Potrebno je šifrirati osjetljive podatke i u prijenosu i u mirovanju kako bi se zaštitili od krađe podataka. Mnoge baze podataka nude ugrađenu podršku za Transparent Data Encryption (TDE), ali ako TDE nije dostupna, može se koristiti enkripcija na razini stupca (npr. za PostgreSQL može se koristiti pgcrypto za enkripciju na razini stupca: INSERT INTO table_name (column_name) VALUES (pgp_sym_encrypt('data', 'encryption_key'));). [21]
- **Automatizirano testiranje sigurnosne kopije i vraćanja:** Potrebno je redovito izrađivati sigurnosne kopije baze podataka, osiguravajući da su sigurnosne kopije šifrirane i pohranjene izvan lokacije direktorija. Također, potrebno je implementirati strategiju za provjeru valjanosti sigurnosnih kopija povremenim vraćanjem kako bi se testirao njihov integritet (npr. za PostgreSQL: pg_dump -U username -F c -f backup_file.sqlc database_name). [21]

4.4. Konfiguracija datoteka i dopuštenja CMS platforme

- **Zaštita .htaccess datoteke:** .htaccess je kratica za Hypertext Access, to je konfiguracijska datoteka koju čita poslužitelj. Nalazi se u korijenskom direktoriju svih CMS-ova kada se preuzme, vrlo je važna za CMS jer se datoteka koristi za sigurnost web stranice, optimizaciju web stranice, preusmjeravanje posjetitelja na različite stranice te omogućavanje i onemogućavanje funkcionalnosti na web stranici. Bilo koja sigurnosna praksa može se unaprijediti tako da se upiše u htacesses datoteku. To je moćna datoteka za korištenje na web stranici, stoga je treba zaštititi. U Joomla datoteku treba aktivirati promjenom datoteke htaccess.txt u .htaccess. Ovaj se korak izvodi preimenovanjem datoteke u upravitelju datoteka web-mjesta. Ovo je vrlo važan korak budući da je

.htaccess konfiguracijska datoteka za korištenje na web poslužiteljima koji pokreću softver web poslužitelja Apache. Što se tiče Drupala i WrodPress već je aktivirana, ali je treba osigurati ponovnim kopiranjem, ograničavanjem pristupa datoteci i sprječavanjem pregledavanja direktorija [16, str. 41].

- **Dozvole direktorija:** Kako bi web stranica bila sigurna, potrebno je postaviti odgovarajuće dozvole za datoteke. Jedan od najčešćih sigurnosnih problema u CMS-u je taj što neki korisnici posjeduju dopuštenja za mape i datoteke kojima ne bi smjeli imati pristup, što može ugroziti sigurnost web stranice. Na primjer, ako više osoba koristi web mjesto, poput dizajnera, administrator bi trebao odrediti odgovarajuće dozvole. Za svaku web stranicu postoje tri vrste pristupa: čitanje, pisanje i izvršavanje. Datoteke s pristupom za čitanje mogu se prikazati korisnicima, dok se datoteke za pisanje mogu mijenjati, a korisnici s pravom izvršavanja mogu pristupati i izvršavati te datoteke. Web poslužitelj mora imati pristup čitanju web stranica kako bi ih mogao prikazati u pregledniku. Stoga bi dopuštenja trebala slijediti preporuke hosting tvrtke. Mnoge hosting tvrtke daju zadane preporuke, ali korisnik ih može mijenjati. Međutim, svaka promjena koju korisnik napravi može utjecati na sigurnost datoteka i mapa [16, str. 42].
- **Uređivanje datoteka putem CMS-a:** Onemogućavanjem mogućnosti uređivanja datoteka izravno iz administratorske ploče CMS-a smanjuje se rizik da kompromitirani administratorski račun bude iskorišten za ubacivanje zlonamjernog koda.
- **Promjena početnog mjesta za prijavu:** Stranica za prijavu mora biti zaštićena jer joj se može lako pristupiti unosom „naziv web stranice/Administrator“, čime postaje dostupna. Ova sigurnosna mjera može se postići korištenjem dodatka koji omogućuje preimenovanje stranica za prijavu na drugu adresu.
- **Onemogućiti izvršavanje PHP-a u ranjivim direktorijima:** Jedna od ključnih sigurnosnih mjera za CMS platforme je sprječavanje izvršavanja PHP skripti u direktorijima gdje se ne bi trebale izvoditi, poput direktorija za učitavanje medija (/wp-content/uploads/ u WordPressu). Time se smanjuje rizik od pokretanja zlonamjernih datoteka koje bi napadač mogao učitati. Ova mjera može se konfigurirati dodavanjem pravila u datoteku .htaccess.
    - WordPress i Joomla dodati sljedeći kod: „<Files *.php> deny from all </Files>“
    - Drupal: Koristiti Paranoia modul za otkrivanje i blokiranje područja u kojima bi se moglo koristiti izvršavanje PHP koda. [22]

- **Ispravne dozvole za datoteke:** Direktoriji trebaju imati 755 dopuštenja, a datoteke 644. Za osjetljive konfiguracijske datoteke (wp-config.php, configuration.php, settings.php), dozvole treba postaviti na 400 ili 440 kako bi osigurali da su čitljive, ali ne i pisane. [22]
- **Zaštititi kolačiće s HTTPOnly i sigurnim zastavicama:** Kako bi se zaštitili kolačići sesije od krađe ili neovlaštenog mijenjanja, potrebno je konfigurirati kolačiće s oznakama „HTTPOnly“ i „Secure“. Ovo pomaže u sprječavanju napada poput skriptiranja između web-mjesta (XSS) da pristupe kolačićima sesije.
    - U WordPress-u treba konfigurirati wp-config.php datoteku na sljedeći način: @ini_set('session.cookie_httponly', true); @ini_set('session.cookie_secure', true);
    - Drupal i Joomla: Slične modifikacije se mogu napraviti u njihovim konfiguracijskim datotekama ili kroz postavke servera. []

4.5. Upravljanje korisnicima

- **Onemogućiti standardnog admin korisnika:** Potrebno je onemogućiti standardnog administrativnog korisnika sa svim ovlastima koji ima korisničko ime „admin“ te postaviti korisničko ime koje je teže za pogoditi.
- **Korištenje jakih lozinki:** Najbolja sigurnosna praksa bi bila da korisnici koriste jake lozinke koje su teške za pogoditi i koje već nisu kompromitirane. Kako bi se izbjegli Brute-force napadi, preporučuje se korištenje što duže lozinke. Također, korisnici bi trebali redovito mijenjati i ažurirati svoje lozinke, primjerice svakih mjesec dana.
- **Omogućiti dvofaktorsku autentifikaciju:** Potrebno je omogućiti dvofaktorsku autentifikaciju preko aplikacija koje generiraju jedinstveni ključ za korisnika u određenom vremenskom periodu, jer pristup tom ključu ima isključivo vlasnik uređaja na kojem se aplikacija nalazi. Preporučuje se korištenje „Google Authenticator-a“ za dvofaktorsku autentifikaciju.
- **Ograničavanje pristupa:** Različiti korisnici imaju različite privilegije, stoga je potrebno ograničiti koje privilegije svaki korisnik ima, u skladu s njihovim ulogama.
- **Kontrola korisnika:** Potrebno je periodično provoditi kontrolu korisnika kako bi se provjerilo koja prava imaju te ukloniti korisničke račune koji se ne koriste ili oduzeti prava korisnicima koji ih više ne trebaju.

4.6. Dodatci u CMS platformama

- **Vatrozid web aplikacije:** Vatrozid je sustav koji se koristi za zaštitu CMS-a od cjelokupnog dolaznog prometa. Blokirat će mnoge sigurnosne prijetnje prije nego što uspiju doći do web stranice. Dodatak vatrozida neophodan je za bilo koje CMS web mjesto [16, str. 46].
- **Zaustavljanje enumeracije:** Enumeracija služi napadačima kako bi saznali što više informacija o platformi te sukladno tome postoje dodatci koji onemogućuju enumeraciju CMS platformi.
- **Ograničavanje pokušaja prijave:** Mnogi dodatci za sigurnost imaju u sebi značajku za ograničavanje pokušaja prijave na 3-5 puta kako bi se spriječio Brute-force napad.
- **Blokiranje IP adresa:** Kako sigurnosni dodaci nadziru aktivnosti na platformi, oni mogu prepoznati neobične postupke s različitih IP adresa i omogućiti blokiranje tih IP adresa.
- **Ažuriranja:** CMS platforme sadrže dodatke koje provode automatska ažuriranja verzije CMS platforme, dodataka i tema kako ne bi došlo do sigurnosnih propusta.
- **Automatska skeniranja ranjivosti:** Pojedini sigurnosni dodatci sadrže u sebi funkcionalnosti skeniranja dodataka i tema kako bi pronašli ranjivosti u njima.
- **Ograničavanje administratorskog pristupa određenim IP adresama:** Takvo se ograničenje može temeljiti na IP adresama, rasponima IP adresa ili geolokaciji IP adrese.

4.7. Alati za automatizirano otkrivanje ranjivosti

- **WPScan (za WordPress)**: Ovaj alat omogućuje automatsko skeniranje WordPress CMS platformi za poznate ranjivosti, uključujući one u temama i dodacima. WPScan može biti konfiguriran da se redovito izvršava i šalje izvješća administratorima o bilo kakvim problemima.
- **JoomScan (za Joomla)**: Slično kao WPScan, JoomScan automatski skenira Joomla instalacije za poznate ranjivosti u komponentama, modulima i dodatcima.
- **Droopescan (za Drupal)**: Droopescan omogućuje skeniranje Drupal CMS platformi te identificira ranjivosti u jezgru, modulima i temama.
- **ZAP (Zed Attack Proxy)**: ZAP se može koristiti za skeniranje CMS platformi automatskim indeksiranjem web stranica i aplikacija, testiranjem problema kao što su SQL injection, Cross-Site Scripting (XSS) i neispravna autentifikacija. Također uključuje alate za ručno sigurnosno testiranje i daje detaljna izvješća o otkrivenim ranjivostima.

- **Skipfish:** Djeluje tako da generira interaktivnu kartu web-mjesta ciljne aplikacije, a zatim agresivno ispituje sigurnosne probleme kao što su skriptiranje na više stranica, krivotvorenje zahtjeva između stranica (CSRF), ubacivanje SQL-a i drugi. Skipfish je učinkovit i pruža mogućnosti skeniranja velike brzine, što ga čini prikladnim za korištenje u procjenama ranjivosti CMS-a.
- **Burp Suite:** Alat uključuje niz značajki kao što su proxy poslužitelj, alat za indeksiranje weba, skener za prepoznavanje ranjivosti kao što su SQL injection i XSS te alate za testiranje upravljanja sesijom, autentifikaciju i kontrole pristupa. Burp Suite često se koristi za dubinsko testiranje CMS platformi i drugih web aplikacija, pružajući detaljan uvid u potencijalne sigurnosne slabosti.
- **Nmap:** Nmap je moćan alat koji se koristi za skeniranje priključaka, što je temeljni korak u procjeni sigurnosti mreže. Nmap identificira otvorene portove na poslužitelju ili mreži, omogućujući administratorima da razumiju koje su usluge izložene javnosti i koje bi mogle biti potencijalne ulazne točke za napadače.
    - Na primjer naredba „nmap -sS -p- your-server-ip“ izvodi SYN skeniranje, koje brzo identificira otvorene portove. To isto skeniranje može se zakazati pomoću cron poslova u Linuxu: „crontab -e“, „0 1 * * 7 nmap -sS -p- your-server-ip >> /path/to/logfile“. Prikazana naredba zakazuje skeniranje svake nedjelje u 1 ujutro; 0 1 određuje kada će se skeniranje pokrenuti (0 određuje minute, a 1 sate), * * * ove tri zvjezdice predstavljaju dan u mjesecu, mjesec i dan u tjednu, 7 predstavlja nedjelju što znači da će se skeniranje odvijati svakih tjedan dana.
- **OpenVAS:** Sveobuhvatan je alat koji se koristi za prepoznavanje ranjivosti u mreži ili CMS platformi. Skenira sigurnosne probleme u konfiguracijama poslužitelja, aplikacijama i mrežnim uslugama, pomažući administratorima da otkriju slabosti prije nego što ih napadači mogu iskoristiti.
    - Automatiziranje OpenVAS skeniranja može se postići korištenjem ugrađene značajke planiranja. OpenVAS omogućuje konfiguriranje zadataka skeniranja i njihovo raspoređivanje u redovitim intervalima (dnevno, tjedno, itd.). Time se osigurava kontinuirano praćenje sustava zbog novih ranjivosti. Kako bi se konfiguriralo automatizirano skeniranje potrebno je otići na karticu „Configuration > Schedules“, zatim kreirati novi raspored koji se pokreće npr. svaku nedjelju u 1 ujutro. Na kraju slijedi povezivanje rasporeda s određenim zadatkom skeniranja, čim je to postavljeno OpenVAS će automatski provoditi skeniranje i generirati izvještaje, šaljući ih administratoru ako se ta opcija podesi.

4.8. Ažuriranje i održavanje CMS platforme

- **Dodatci:** Potrebno je provoditi redovita ažuriranja dodataka kako bi se primijenile najnovije sigurnosne zakrpe.
- **Teme:** Potrebno je provoditi redovita ažuriranja tema kako bi se primijenile najnovije sigurnosne zakrpe.
- **Verzija CMS platforme:** Potrebno je provoditi redovita ažuriranja verzije platforme kako bi se primijenile najnovije sigurnosne zakrpe.
- **Korištenje sigurnosnih dodataka:** Važno je koristiti sigurnosne dodatke koji pružaju mnoge mogućnosti različite zaštite.
- **Provođenje redovitih skeniranja:** Kako bi se otkrile pojedinačne ranjivosti potrebno je periodično provoditi skeniranje CMS platforme kako bi se utvrdilo postoje li ranjivosti na samoj platformi.

4.8. Primjena sigurnosnog okvira

Kako bi se procijenila učinkovitost razvijenog sigurnosnog okvira, potrebno ga je primijeniti na postojeću CMS platformu.

4.8.1. Ažuriranje platforme

Kako je verzija postojeće CMS platforme bila zastarjela, bilo je potrebno ažurirati je, što je i učinjeno, te je verzija nadograđena na 6.6.1.

Nakon ažuriranja WordPress-a, provedeno je ažuriranje dodataka „Photo Gallery“, „WP Google Review Slider“ i „Aksimet Anti-Spam“, dok je dodatak „Mail Masta“ uklonjen jer nije dobio ažuriranja. što znači da je zastario i ima mnogo sigurnosnih propusta.

Zadnje što je preostalo bilo je ažuriranje tema, pri čemu su uklonjene sve neaktivne teme, a aktivna tema je ažurirana.

4.8.2. Instaliranje sigurnosnih dodataka

Dodani su sigurnosni dodaci Sucuri Security, Solid Security, Wordfence Security, WPS Hide Login, Stop User Enumeration i Easy Updates Manager.

Sucuri Security ima sljedeće značajke: reviziju sigurnosnih aktivnosti, nadzor integriteta datoteke, daljinsko skeniranje zlonamjernog softvera i praćenje crne liste. Također, Sucuri Security provodi redovna skeniranja stranice kako bi pronašao moguće ranjivosti.

Solid Security ima sljedeće značajke: dvofaktorsku autentifikaciju (2FA) i zaštitu protiv Brute-force napada.

Wordfence Security sastoji se od vatrozida (WAF) i skenera zlonamjernog softvera; WAF identificira i blokira zlonamjerni promet, a premium verzija pruža pravila vatrozida i ažuriranja potpisa zlonamjernog softvera u stvarnom vremenu. Premium verzija također omogućuje popisa crnih IP adresa u stvarnom vremenu kako bi se blokirali svi zahtjevi poznatih zlonamjernih IP adresa.

WPS Hide Login služi za promjenu stranice za prijavu s zadane web adrese na proizvoljnu web adresu.

Stop User Enumeration pomaže u zaustavljenju enumeracije CMS platforme.

Na kraju je dodan dodatak Easy Updates Manager kako bi se omogućila automatska ažuriranja verzije WordPressa, tema i dodataka.

4.8.3. Konfiguracija datoteka i dopuštenja CMS platforme

Izmijenjena je wp-config.php datoteka kako bi se omogućila automatska ažuriranja korištenjem sljedećih komandi.

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

add_filter( 'auto_update_plugin', '__return_true' );

add_filter( 'auto_update_theme', '__return_true' );
```

Zatim je stranica za prijavu /wp-admin promijenjena u /my-custom-login korištenjem dodatka WPS Hide Login.

Korištenjem sigurnosnog dodatka „Sucuri Security“ onemogućen je pristup uređivanju datoteka administratoru iz bočnog izbornika, kao i pretraživanje direktorija.

4.8.4. Upravljanje korisnicima

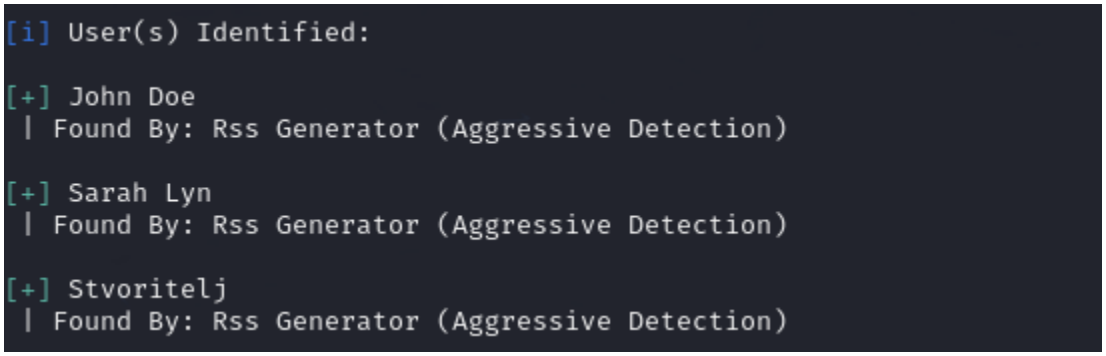
Onemogućen je standardni korisnik „admin“ promjenom korisničkog imena i lozinke u teško pogodivu kombinaciju. Također, sva preostala korisnička imena su izmijenjena i postavljene su jake lozinke za sve korisnike.

Uključena je dvofaktorska autentifikacija za sve korisnike koristeći sigurnosne dodatke i mobilnu aplikaciju „Google Authenticator“.

4.8.5. Korištenje alata za analizu uspješnosti sigurnosnog okvira

Prije uključivanja vatrozida (WAF), provedena je enumeracija kako bi se vidjelo koji su rezultati, odnosno koje bi ranjivosti, korisnici, dodaci, teme i verzija WordPress-a mogli biti otkriveni. Nakon svih provedenih mjera zaštite, primijećeno je da je enumeracija trajala znatno dulje nego inače, oko 45 minuta. WPScan je uspio prepoznati neke dodatke i njihove prijašnje ranjivosti, ali nije mogao utvrditi verziju dodataka, kao ni verziju samog WordPress-a.

Nakon enumeracije dodataka, pokušana je enumeracija korisnika, ali WPScan je uspio pronaći samo njihovo ime i prezime, a ne korisničko ime kao u prethodnim slučajevima.



Slika 27. Rezultat enumeracije korisnika [Izvor: vlastita izrada]

Nakon provedene enumeracije, odlučeno je uključiti vatrozid unutar sigurnosnog dodatka „Wordfence“ i pokušati provesti Brute-force napad na račun korisnika „Sarah Lyn“, iako korisničko ime nije bilo otkriveno, a ono je zapravo sarah123. Primijećeno je da Brute-force napad nije mogao započeti zbog aktivnog vatrozida.

Slika 28. Pokušaj Brute-force napada [Izvor: vlastita izrada]

Nakon uključivanja vatrozida, ponovno je pokušana cjelokupna enumeracija WordPress platforme, no ona nije mogla započeti zbog aktivnog vatrozida.

Slika 29. Pokušaj enumeracije [Izvor: vlastita izrada]

5. Zaključak

U ovom završnom radu istražena je sigurnost CMS platformi kroz razvoj robusnog sigurnosnog okvira, koji se temelji na detaljnoj analizi postojećih ranjivosti, penetracijskom testiranju i implementaciji najboljih sigurnosnih praksi. CMS platforme, kao što su WordPress, Joomla i Drupal, iako izuzetno korisne i fleksibilne, također predstavljaju značajan sigurnosni rizik zbog svoje popularnosti i otvorenog koda.

Razvijeni sigurnosni okvir obuhvaća niz preporuka i alata koji omogućuju automatizaciju sigurnosnih provjera i ažuriranja, čime se smanjuje rizik od iskorištavanja poznatih ranjivosti. Kroz primjenu ovog okvira, pokazalo se da se može značajno unaprijediti sigurnost CMS platformi, smanjujući broj uspješnih napada i osiguravajući da su sustavi uvijek ažurirani s najnovijim sigurnosnim zakrpama.

Penetracijsko testiranje provedeno na ranjivoj verziji WordPress platforme pokazalo je kako su zastarjele verzije CMS-a i dodataka ključna točka napada te je istaknulo važnost redovitog održavanja i ažuriranja. Implementacijom sigurnosnog okvira, uključujući upotrebu alata kao što su WPScan i Burp Suite, te uvođenjem sigurnosnih dodataka poput Wordfencea i Sucuri Securityja, značajno su smanjene mogućnosti napada na CMS platformu.

Ovaj rad naglašava potrebu za stalnim poboljšanjem i prilagođavanjem sigurnosnih mjera u skladu s novim prijetnjama. Iako je razvijeni okvir uspješno smanjio sigurnosne rizike, potrebno je nastaviti s istraživanjem i razvijanjem dodatnih sigurnosnih mjera kako bi se osigurao dugoročni uspjeh i sigurnost CMS platformi.

Popis literature

[1] „What is a content management system? (CMS)“ (n.d.), 2018. [Na internetu]. Dostupno: https://kinsta.com/knowledgebase/content-management-system/ [pristupano 20.08.2024 s]

[2] D. Barker, *Web Content Management*, 1. izdanje, Sebastopol, CA: O'Reilly Media, Inc., 2016. https://books.google.hr/books?hl=hr&lr=&id=x6_NCwAAQBAJ&oi=fnd&pg=PR2&dq=cms+platforms&ots=VLPHCpoVHQ&sig=I6X7Al8N7bSf5hxbyxJ36KKpgbE&redir_esc=y#v=onepage&q&f=false

[3] M. Martin, „5 Types of CMS“, 2023. [Na internetu]. Dostupno: https://www.designrush.com/agency/web-development-companies/trends/types-of-cms [pristupano 26.08.2024.]

[4] M. Gopinath, „What are the Different Types of Content Management Systems?“, 2024. [Na internetu]. Dostupno: https://www.vplayed.com/blog/types-of-cms-content-management-system/ [pristupano 26.08.2024.]

[5] J. Varty, „List of Top CMS Security Vulnerabilities“, 2021. [Na internetu]. Dostupno: https://agilitycms.com/resources/posts/cms-security-vulnerabilities [pristupano 26.08.2024.]

[6] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, C. Meinel, „Large Language Models in Cybersecurity: State-of-the-Art“ 2024. [Na internetu]. Dostupno: https://arxiv.org/html/2402.00891v1 [pristupano 04.09.2024.]

[7] T. O. Abrahams, S. K. Ewuga, S. O. Dawodu, A. O. Adegbite, A. O. Hassan, „A REVIEW OF CYBERSECURITY STRATEGIES IN MODERN ORGANIZATIONS: EXAMINING THE EVOLUTION AND EFFECTIVENESS OF CYBERSECURITY MEASUERS FOR DANA PROTECTION“, 2024. [Na internetu]. Dostupno: https://www.researchgate.net/publication/377346019_A_REVIEW_OF_CYBERSECURITY_STRATEGIES_IN_MODERN_ORGANIZATIONS_EXAMINING_THE_EVOLUTION_AND_EFFECTIVENESS_OF_CYBERSECURITY_MEASURES_FOR_DATA_PROTECTION [pristupano 04.09.2024.]

[8] National Cyber Security Centre NCSC, „Measures to secure content management systems (CMS)“, 2024. [Na internetu]. Dostupno: https://www.ncsc.admin.ch/ncsc/en/home/infos-fuer/infos-unternehmen/aktuelle-themen/massnahmen-schutz-cms.html [pristupano 26.08.2024.]

[9] B. Ramos-Cruz, J. Andreu-Perez, L. Martinez, „The cybersecurity mesh: A comprehensive survey of involved artificial intelligence methods, cryptographic protocols and challenges for future research“, 2024. [Na internetu]. Dostupno: https://www.sciencedirect.com/science/article/pii/S092523122400198X [pristupano 05.09.2024.]

[10] S. A. Mohsin, A. S. Alfoudi, „Traffic dana classification in SDN network based on machine learning algorithms“, 2024. [Na internetu]. Dostupno: https://www.iasj.net/iasj/download/e675363dd1a8a474 [pristupano 05.09.2024.]

[11] „XAMPP“, (bez dat.) u Wikipedia, the Free Encyclopedia. Dostupno: https://en.wikipedia.org/wiki/XAMPP [pristupano 26.08.2024]

[12] S. Atram, „What is Website enumeration?“, 2022. [Na internetu]. Dostupno: https://www.linkedin.com/pulse/what-website-enumeration-sushant-a/ [pristupano 26.08.2024.]

[13] M. Zakee, „Directory Indexing: Security threat“, 2024. [Na internetu]. Dostupno: https://medium.com/@zakeeandroid/directory-indexing-security-thread-9ad3d88fc96e [pristupano 26.08.2024.]

[14] „What is xmlrpc.php In WordPress & Why Should You Disable It?“ (n.d.), 2024 [Na internetu]. Dostupno: https://www.sitelock.com/blog/wordpress-xmlrpc/ [pristupano 28.08.2024]

[15] „What is Security Framework?“ (n.d.), Dostupno: https://www.cyberark.com/what-is/security-framework/ [pristupano 28.08.2024.]

[16] R. A. Alghofaili, Security Analysis of Open Source Content Management Systems Wordpress, Joomla, and Drupal [Diplomski rad]. California State Polytechnic University, Pomona, CA, SAD, 2018, Dostupno: https://scholarworks.calstate.edu/downloads/z029p7034 [pristupano 28.08.2024.]

[17] „A Complete Security Guide for WordPress, Joomla, and Drupal Users“ (n.d.), 2024. [Na internetu]. Dostupno: https://responsible-cyber.com/blogs/guides-tutorials/a-complete-security-guide-for-wordpress-joomla-and-drupal-users [pristupano 07.09.2024]

[18] C. Kumar, „Apache Web Server Hardening and Security Guide“, 2024. [Na internetu]. Dostupno: https://geekflare.com/apache-web-server-hardening-security/ [pristupano 07.09.2024.]

[19] StackHawk, „Web API Security: Essential Strategies and Best Practices“, 2024. [Na internetu]. Dostupno: https://www.stackhawk.com/blog/web-api-security-essential-strategies-and-best-practices/ [pristupano 07.09.2024]

[20] Microsoft, „Recommendations for hardening resources“, 2023. [Na internetu]. Dostupno: https://learn.microsoft.com/en-us/azure/well-architected/security/harden-resources [pristupano 08.09.2024.]

[21] C. Kime, „7 Database Security Best Practices: Database Security Guide“, 2023. [Na internetu]. Dostupno: https://www.esecurityplanet.com/networks/database-security-best-practices/ [pristupano 08.09.2024.]

[22] A. Moussa, „Best Practices to Secure your Joomla Website“, 2021. [Na internetu]. Dostupno: https://magazine.joomla.org/all-issues/april-2021/best-practices-to-secure-your-joomla-website [pristupano 08.09.2024.]

[23] „How to secure Drupal – tips & best practices“ (n.d.), 2016. [Na internetu]. Dostupno: https://www.fdgweb.com/2016/04/15/how-to-secure-drupal-tips-best-practices/ [pristupano 09.09.2024.]

Popis slika