Agregiranje sigurnosnih ranjivosti sustava za uređivanje sadržaja (CMS)

Omazić, Filip

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Dvopredmetni studij informatologije i informacijske tehnologije

Filip Omazić

Agregiranje sigurnosnih ranjivosti sustava za uređivanje sadržaja (CMS)

Diplomski rad

Mentor Doc. dr. sc. Tomislav Jakopec

Osijek, 2022.

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Odsjek za informacijske znanosti

Dvopredmetni studij informatologije i informacijske tehnologije

Filip Omazić

Agregiranje sigurnosnih ranjivosti sustava za uređivanje sadržaja (CMS)

Diplomski rad

Društvene znanosti, informacijske i komunikacijske znanosti,
informacijsko i programsko inženjerstvo

Mentor Doc. dr. sc. Tomislav Jakopec

Osijek, 2022.

Prilog: Izjava o akademskoj čestitosti i o suglasnosti za javno objavljivanje

Obveza je studenta da donju Izjavu vlastoručno potpiše i umetne kao treću stranicu završnoga, odnosno diplomskog rada.

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napisao/napisala te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s navođenjem izvora odakle su preneseni.

Svojim vlastoručnim potpisom potvrđujem da sam suglasan/suglasna da Filozofski fakultet u Osijeku trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta u Osijeku, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku, 30.04.2022

Filip Omazić, 0269120856

Ime i prezime studenta, JMBAG

Sadržaj

Sažetak

Razmjena podataka i informacija na internetu većinom se provodi ispod vidljivog dijela, no ljudi kao korisnici weba najčešće dohvaćaju informacije putem mrežnih stranica. Mrežne stranice su često izrađene koristeći kodiranje to jest ručno kreiranje, no s vremenom raste i popularnost sustava za upravljanje sadržajem (Web Content management system - CMS ili WCMS) za mrežna mjesta. Ovakvi sustavi revolucionarizirali su izrade mrežnih stranica jer najčešće ne zahtijevaju znanje o kodiranju već samo jednostavnu uspostavu i izmjenjivanje i premještanje segmenata stranice klikovima miša i tipkama tipkovnice, sve bez kodiranja. Jednostavnost ove vrste osvojila je srca velikog dijela mrežnih stranica na internetu. Kibernetička sigurnost po svojoj prirodi uvijek traži gdje može popraviti sigurnosne ranjivosti i pružiti zakrpe, pa se dotiče sustava upravljanjem sadržajem radi njihove poznatosti i količine ranjivosti koje posjeduju. Radi se o većoj količini ranjivosti, te je potrebno pobrinuti se da CMS-ovi generalno budu sigurniji, neovisno o vrsti i imenu CMS-a. Napadači često traže ranjivosti koje će im na lak način dati više ovlasti stoga pregledavaju repozitorije programa koji iskorištavaju ranjivosti i baze podataka koje opisuju ranjivosti te na lak način, često pokretanjem jednostavne programske skripte, dobivaju ulaz u sustav. Opasnost dodatno raste kad se radi o lančanim napadima za istu ranjivost kod sustava s istim ranjivim programskim rješenjem. Ovaj rad prikazuje trenutne probleme s kojima se sigurnost CMS sustava suočava i predstavlja platformu za detekciju i agregaciju ranjivosti na vrlo nisko nametljiv način, te se dotiče načina na koji stručnjaci za kibernetičku sigurnost ustanovljuju ranjivosti za CMS-ove.

Ključne riječi: Kibernetička sigurnost, ranjivosti web sjedišta, sustavi za uređivanje sadržaja (CMS), PHP programski jezik.

Uvod

Doba digitalnih podataka i informacija odavno obogaćuje svijet, a dolaskom interneta omogućava razmjenu komunikacija u dotad nepojmljivoj brzini. Količina rasta napretka u ovakvom omjeru je tad bila nevjerojatna, a kao glavni medij interneta (barem za prosječnog korisnika) postavljen je world wide web, te su na njemu postavljene tisuće mrežnih stranica koje sadrže razne podatke i informacije i njima korisnici pristupaju koristeći web pretraživač. Ove stranice počele su se kreirati jednostavnim HTML jezikom, te im se uskoro pridružuje CSS koji pruža ljepši izgled, te kasnije JavaScript koji daje dodatne funkcionalnosti mrežnim stranicama. Nakon nekoliko godina manualnih kodiranja mrežnih stranica u području informacijskih tehnologija dolazi se do ideje upravljačkog programskog rješenja koji će na temelju raznih predložaka i korisnikovog izmjenjivanja predloška stranice, dodijeljenog od sustava, poput teksta u Microsoft Wordu Office paketa, izmjenjivati svoju buduću stranicu. Ovdje se, dakle, dolazi se do ideje Web CMS-a (Content management system – Sustav za upravljanje sadržajem) kakvog poznajemo danas. Web CMS ili sustav za upravljanje sadržajem neke web stranice je tehnologija koja je omogućila kreaciju mrežnih stranica koje izgledaju profesionalno, s lakoćom kreacije čak i bez znanja kodiranja. Ovaj korak revolucionarizirao je svijet razvijanja stranica na webu. S obzirom na to da danas veliki broj mrežnih mjesta (usmjerenih na web) na internetu koristi razne CMS-ove potrebno je uzeti u obzir i sigurnosni pogled na njih, naravno, u području kibernetičke sigurnosti ova tema je već vrlo poznata stoga je pristup samoj potkrijepljen množinom izvora. CMS sustavi često imaju mnoge kritične ranjivosti koje nerijetko rezultiraju i lančanim napadima, a s obzirom na količinu u kojoj se koriste kibernetička sigurnost ne može stajati mirno i gledati masovnu eksploataciju ranjivosti milijuna mrežnih stranica na internetu. Potencijalno rješenje za ovakve probleme bio bi alat koji bi detektirao i prijavljivao ranjivosti za CMS kojeg mrežna stranica koristi. Samo sučelje može nuditi i opcije pregleda npr. “Exploita” (programa za iskorištavanje ranjivosti) kako bi se proširio krug potencijalnih korisnika.

Svijet interneta, weba i CMS-ova

Analiza cijelog interneta i mrežnih stranica na njemu provedena je od strane Builtwith organizacije koja je popularna za ovakve analize i provođenje istraživanja o webu. Prikazuje kako su većina stranica koje koriste sustave za uređivanje sadržaja izgrađene koristeći WordPress, a zatim Wix CMS, te nedaleko nakon toga stoje Joomla i Drupal - koji su na nekoliko ostalih istraživanja prikazani kao poznati. Iako Wix nije idealan primjer CMS-a radi svojih mogućnosti i samog Hostinga, moguće ga je ovdje i svrstati radi komponente za izradu stranica sličnoj CMS-u. Builtwith navodi kako su od skeniranja cijelog interneta saznali da oko 72.5 milijuna stranica na internetu koristi bar nekakav CMS.[1] Ovaj broj ne predstavlja postotak usko ispod 50% svih stranica na internetu, kao što neki izvori spominju, jer je brojka svih stranica na internetu otprilike u milijardi, pa čak i više.[2] S obzirom da većina članaka na internetu navodi kako postoji više od milijardu mrežnih stranica (većina kaže da su oko 80% njih aktivni), a drugi izvori poput W3tech stranice (stranica koja se bavi preciznim analitikama weba) navode da se na 43% od svih stranica na internetu pokreće WordPress, možemo zaključiti samo na WordPressu da u ovoj količini CMS-ovi (generalno) predstavljaju vrlo važan dio weba, te s obzirom da se radi o eksponencijalno rastućem broju stranica s CMS-ovima - rast je garantiran bar za blisku budućnost.[3] Prema istraživanju u Siječnju 2018. godine najveći udio tržišta kod CMS-ova drži WordPress s 60%. Iza njega je odmah Joomla! sa 6.5% i Drupal s 4.6% što prikazuje kako WordPress drži strogu većinu kod web stranica koje koriste CMS (drugi izvori dodaju i Wix u listu top 3 najpopularnijih, no ovdje nije niti u prvih 5). Što se tiče popularnosti samih web CMS-ova, 2011. godine je samo 23.6% stranica imalo neku vrstu CMS-a, a već 2018. ta brojka skače na 48.8%. Iako su ovi postotci kod svakog od izvora drukčiji, lako je zaključiti da velik dio interneta koristi CMS-ove, a da je uvijek na prvom mjestu WordPress (kao i množina drugih besplatnih rješenja, uključujući i Wix koji ima i plaćenu opciju). Problem ranjivosti ovakvih CMS-ova može uzrokovati "efekt leptira"[4], počevši od malicioznih izmjena vidljivoj strani mrežne stranice (web defacement), do potpunog

1 BuiltWith. CMS Usage Distribution on the Entire Internet: Distribution for websites using CMS technologies, 2022. URL: https://trends.builtwith.com/cms/traffic/Entire-Internet (2022-02-09)
2 Total number of Websites, 2022. URL: https://www.internetlivestats.com/total-number-of-websites/ (2022-02-03)
3 W3techs. Usage statistics of content management systems, 2022. URL: https://w3techs.com/technologies/overview/content_management (2022-02-03)
4 Nijs, Diane Elza Lea Winie. Imagineering the butterfly effect. URL: https://pure.rug.nl/ws/portalfiles/portal/6594643/Imagineering_the_Butterfly_Eff_1.pdf (2022-04-03)

preuzimanja ili gašenja stranice, infiltracije napadača u server i računalo i krađe podataka te omogućavanja lančanih napada svakim preuzimanjem kontrole. Iako većina izvora navodi da se većina najvažnijih i naj kritičnijih ranjivosti pronalazi u samoj srži CMS-ova, neki od izvora govore kako se ranjivosti CMS-ova većinski nalaze u dodacima (Plug-In-ovima) i ekstenzijama - izvori spominju da se ovdje radi čak o 80% od svih ranjivosti.[5] Ovakvim predstavljanjem različitih analiza i istraživanja, bez obzira na izvor moguće je shvatiti da su ključne komponente za pregled sigurnosti poznatost CMS-a, otvorenost koda i plug-inovi.

Sustavi za upravljanje sadržajem

Povijesno gledano CMS-ovi su definirani na različite načine, poput upravljanja dokumentima i datotekama do samih web CMS-ova kojih se dotiče ovaj rad. FileNet, Vignette i slični sustavi počeli su s upravljanjem datoteka i imali vodeću poziciju što se poznatosti tiče. 2000-tih godina dolaze i sustavi za upravljanje web sadržajem otvorenog koda kao što su WordPress, Drupal i Joomla - koji danas vladaju internetom radi jednostavnosti, činjenice da su besplatni i čestih ažuriranja te velike zajednice koja odgovara na pitanja na forumima ovih CMS-ova. Sustavi za upravljanje sadržajem i objavljivanjem prvi puta se pojavljuju, onakvima kakvih ih poznajemo danas, otprilike 1995. godine.[6] Radi raznolikosti potreba u današnjem poslovnom svijetu analize istraživanja često pokazuju kako većina organizacija želi prijeći na CMS-ove ako ih dosad već nisu koristili. Istovremeno, organizacije koje su već koristile neki od CMS-ova napominju kako bi voljeli prijeći na neki napredniji CMS s više mogućnosti i svojstava. 54% ispitanika odabire opciju "Da i zadovoljan sam" kod pitanja koriste li CMS i jesu li zadovoljni, a 80% ispitanika koji ne koriste CMS odgovara s "Da" na pitanje hoće li početi koristiti neki CMS u idućih dvije godine. Ovakva istraživanja dokazuju da se na tržištu weba radi o čak eksponencijalnom rastu korištenja CMS-ova.[7] Na temelju ovakvih spoznaja zaključno je reći da su CMS-ovi bar bliža budućnost, a i sadašnjost svijeta mrežnih stranica. Važno je napomenuti i da postoje razne vrste CMS-ova, te se

5 Martinez-Caro, Jose-Manuel; Aledo-Hernandez, Antonio-Jose; Guillen-Perez, Antonio; Sanchez-Iborra, Ramon; Cano, Maria-Dolores. A Comparative Study of Web Content Management Systems. // Information vol. 9, no. 27 (2018). URL: https://www.mdpi.com/2078-2489/9/2/27/pdf (2022-02-07)

6 CMSWiki. History of Content management systems, 2011. URL: https://web.archive.org/web/20110518053639/http://www.cmswiki.com/tiki-index.php?page=HistoryOfCMS (2022-02-13)

7 Ramalingam, Elanchezhian. Research Paper on Content Management Systems (CMS): Problems in the Traditional Model and Advantages of CMS in Managing Corporate Websites, 2016. URL: https://digitalcommons.harrisburgu.edu/cgi/viewcontent.cgi?article=1007&context=pmgt_dandt (2022-02-12)

one koriste za razne namjene (poslovni CMS - business CMS, CMS za učenje - learning CMS, itd.), no s obzirom na sigurnost potrebno je obuhvatiti sve CMS-ove neovisno o namjeni. Posebice, s obzirom na popularnost treba se koncentrirati na one CMS-ove za izrade mrežnih stranica, dok su na primjer CMS-ovi za učenje važni, no iz perspektive kibernetičke sigurnosti slučaj hakiranja CMS-a za učenje nije značajan kao i slučaj hakiranja CMS-ova za izrade mrežnih stranica. Bez obzira, ovakvi slučajevi moraju biti spomenuti, uključujući naravno i sigurnost poslovnih CMS-ova koji su često pregledani i osigurani od stručnjaka za kibernetičku sigurnost. Važno je napomenuti, s obzirom na sigurnost i generalni pregled vrsta CMS-ova, da postoje CMS-ovi koji su utemeljeni na kodu u otvorenom pristupu te na kodu koji je "closed-source" ili "proprietary".[8] Samim time, postoje komercijalni i besplatni CMS-ovi. Pri odabiru CMS-a korisnik odlučuje o rješenju, a većina ih se odluči za rješenje otvorenog koda radi poznatosti, široke zajednice, ali i kvalitete (primjer WordPressa). Što se sigurnosti tiče argument bi se predstaviti da su oni otvorenog koda manje sigurni, no ovo se rješava konstantnim nadogradnjama i popravcima - dok su plaćeni s druge strane nešto rjeđe nadograđeni, a imaju manje pronađenih ranjivosti. Ovdje je također važno uzeti u obzir i činjenicu da su oni otvorenog koda puno više korišteni, stoga je i napadačima u interesu baš njih hakirati. Mogući i najčešći rizici su napadi poput Brute force napada (napada pogađanja lozinki), napada izvršavanja koda na sustavu žrtve (Remote Code Execution) kroz iskorištavanje raznih ranjivosti na programskom rješenju od ranjivosti prenošenja datoteka do ranjivosti egzekucije koda kroz funkcije web stranice (radi greški ili propusta u kodiranju), SQL injection napada (napad izvršavanja koda u bazi podataka za ekstrahiranje podataka), XSS napada (Cross site scripting - izvršavanje koda na stranici kroz razne forme i mogućnosti izmjenjivanja sadržaja), Denial of service napada (napad uskraćivanja usluge) - bilo izvršenog od čestih XML (eXtensible Markup Language) ili JSON (JavaScript Object Notation) Remote Procedure Call (RPC) ranjivosti. Dodatna problematika javlja se jer sve češće sustavi pate od lančane eksploatacije (eksploatacije množine sustava, pretraženih na tražilicama koji posjeduju istu ranjivost) jer ovo dovodi do puno opasnijih mogućnosti u napadačevim rukama.[9]

8 Meike, Michael; Sametinger, Johannes; Wiesauer, Andreas. Security in Open Source Web Content Management Systems, 2008. URL: https://www.se.jku.at/wp-content/uploads/2009/07/2008.wcms_security.pdf (2022-02-12)

9 IBM (International business machines). Understanding the risks of content management systems: How open source web platforms can open your organization to attack, 2015. URL: http://hosteddocs.ittoolbox.com/undertstandingrisksofCMS.pdf (2022-02-12)

Vrste sustava za upravljanje sadržajem

Iako su spomenuti ranije, važno je dodatno pojasniti podjele CMS-ova. Stoga će biti istaknuti LCMS-ovi, BCMS-ovi te komercijalni i besplatni CMS-ovi.

CMS za učenje ili LCMS (Learning CMS), također poznat kao LMS, koncentriran je na dijeljenje znanja, mogućnosti testiranja i provođenja grupnih učenja i podijele korisnika na učitelje i učenike. CMS-ovi za učenje postaju sve poznatiji i korisniji a rapidnim rastom poznatosti dolaskom COVID-19 pandemije mnogi se upoznaju s CMS-ovima za učenje efektivnije nego ikad.[10]

Poslovni CMS ili BCMS (Business CMS) orijentiran je na poslovanje i trebao bi smanjiti cijenu objave i održavanja informacija, povećati vrijednost web sadržaja, maksimizirati efektivnost timskih vještina omogućavanjem korištenja i pregleda izvoza znanja kroz objave i komunikaciju te mnoge druge aktivnosti koje donose profit. Dokaz isplatljivosti je činjenica da su već 2003. godine poslovno-orijentirani CMS-ovi imali tržišnu vrijednost oko jedne milijarde dolara, a ova brojka je nakon toga samo rasla.[11] Korištenje BCMS-ova u korporacijama, dakle, postaje moćno oruđe u provođenju poslovanja.

CMS-ovi se također dijele na komercijalne (plaćene) i one otvorenog koda (besplatne). Negativne osobine plaćenih CMS-ova su cijena, činjenica da jedna tvrtka pruža ažuriranja i ima vlasništvo programskog rješenja u potpunosti, dok su negativne osobine CMS-ova otvorenog koda su problem garancije rada ili nastavka razvoja za određenu verziju, mogući problemi povezani s intelektualnim vlasništvom u slučaju povezivanja s drukčijim oruđima ili posjedovanja samog u slučaju izmjene otvorenosti koda, kao i slični drugi problemi poput lakšeg pronalaska ranjivosti i češće nadogradnje što je prednost, ali može biti i problem.[12] Iz perspektive sigurnosti, smisleno je pretpostaviti da će najviše korišten CMS imati najviše ranjivosti - jer je i napadačima cilj zahvatiti što više žrtava ili sustava. U današnjem slučaju to su većinom besplatni CMS-ovi, a većina njih

10 Qwaider, Walid Qassim. Information Security and Learning Content Management System (LCMS). // International Journal of Advanced Computer Science and Applications, vol. 8, no. 11 (2017), 588-593. URL: https://thesai.org/Downloads/Volume8No11/Paper_74-Information_Security_and_Learning_Content_Management.pdf (2022-02-10)

11 Crownpeak technology, Inc. The Business Case for a Web Content Management System: A guide for making the case, justifying the cost and choosing the right CMS for your organization, 2016. URL: https://www.crownpeak.com/resources/white-papers/whitepaper_the-business-case-for-a-web-cms.pdf?v=16102605173372 (2022-02-09)

12 TerpSys. Open Source vs. Proprietary CMS: Which is right for my organization, 2010. URL: https://www.terpsys.com/docs/default-source/white-papers/open-source-vs-proprietary-cms.pdf?sfvrsn=7b2f90b4_4 (2022-02-09)

ima kod u otvorenom pristupu, pa je napadaču i lakše razviti programe za iskorištavanje ranjivosti - exploite.

Iako postoje i razne druge vrste sustava za uređivanje sadržaja poput onih orijentiranih prema dokumentima, prema marketingu, mobilnim aplikacijama ili slično - nabrojani primjeri su najuži uz temu web CMS-ova. Radi raznih namjena samih doticanje svake vrste nije potrebno, a radi raširenosti web CMS-ova količina ranjivosti i same ranjivosti dovoljno su interesantne teme.

Ranjivosti programskih rješenja

U svijetu programskih rješenja i njihove sigurnosti ranjivosti se nalaze u bilo kojem dijelu i uvijek treba paziti na njih. Osigurati programsko rješenje od ranjivosti iznimno je težak zadatak i zahtijeva vremena i znanja, a čak ni tad programsko rješenje nije sto postotno sigurno. Zapravo, nijedan sustav ni programsko rješenje nisu savršeno sigurni, to je jednostavno neprirodno. Ovo su, između ostalog, riječi kojih se drže hakeri i istraživači ranjivosti u kibernetičkoj sigurnosti, te se s istom tom politikom i vode u pronalaženju novih ranjivosti. Neovisno radi li se o web CMS-ovima, sustavima u automobilima, sustavima koji upravljaju bankomatima ili video igrama – programska rješenja u bilo kojem smislu mogu imati ranjivosti. Neke od njih mogu biti opasnije, a neke raditi minimalnu štetu - u svakom slučaju postoji opcija prijavljivanja ranjivosti kako bi se nakon popravka nova zakrpa predstavila svijetu. Što se web stranica tiče najčešće ranjivosti koje su pronađene su one na XSS napad (Cross-site scripting) što podrazumijeva izvršavanje koda na web stranici (podrazumijeva JavaScript kod, pa često ne djeluje negativno na samog poslužitelja, već na stranicu), SQLi napad (SQL injection) - ubrizgavanje SQL koda kroz na primjer pretraživačko polje i izvršavanje s ciljem dohvaćanja podataka iz baze podataka (često onih podataka do kojih se ne bi smjelo doći), Brute-force napad - napad pogađanja lozinki i slične napade.[13] Nabrojani napadi nisu najopasniji, već samo neki primjeri. Najopasniji napadi bi bili oni koji napadaču daju kontrolu nad poslužitelj-računalom ili spajaju sustav u skup botova, pa sve do krađe i zlouporabe informacija. Što se CMS-ova tiče, problem raste kad se radi o napadu koji je javno opisan i dostupan za određenu verziju CMS-a koji se često koristi, na primjer zastarjela WordPress verzija - s ovakvom ranjivošću napadač često može pretražiti veliku količinu stranica s istom ranjivošću

13 Positive technologies. Security trends & vulnerabilities review: Web applications, 2016. URL: https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Web-Application-Vulnerability-2016-eng.pdf (2022-02-14)

koristeći tražilice poput Googlea i njegovih pretraživačkih parametara. Ovdje dolazi do lančanog napada i opasnost znatno raste. Ovakvi se pretraživački parametri koriste i za ostale protokole, a u cijeli plan pretraživanja često spada i Shodan tražilica koja prikazuje podatke o IP adresama (uz verzije) i frekventno osvježava svoje baze podataka.[14]

Agregiranje sigurnosnih ranjivosti sustava za uređivanje sadržaja (CMS)

Radi širine korištenja u svijetu mrežnih stranica i interneta te njihovih brojnih ranjivosti važno je spomenuti CMS-ove u pogledu informacijske sigurnosti. Ranjivosti CMS-ova, neovisno o CMS-u, raznolike su i brojne te predstavljaju različite razine opasnosti za žrtvu. Popisivanje ovakvih ranjivosti obavlja Američki nacionalni istraživački centar za kibernetičku sigurnost (NCF) u kooperaciji s CVE MITRE (Common vulnerabilities and exposures) organizacijom koja na svojoj stranici popisuje sve ranjivosti sustava koje su poznate[15] kroz "CVE" brojeve. Iako CVE popisuje i opisuje ranjivosti i citira izvore, ipak ne prikazuje način eksploatacije, te iako to pokušava - ne pokriva sve moguće ranjivosti - otkrivene, a definitivno i skrivene poput 0-day exploita - ili programa koji iskorištava ranjivost nultog dana. Stranice koje prikazuju načine eksploatacije su one poput Exploit-db stranice, 0day-today i slične poznate stranice. Razlog zašto je važno pokazati način eksploatacije je da se programeru koji popravlja grešku pojasni gdje je točno nastala kako bi ju preciznije ispravio. Agregiranje sigurnosnih ranjivosti za CMS sustave podrazumijeva skupljanje CVE brojeva za odgovarajući sustav, u idealnom slučaju s njegovom verzijom radi preciznosti, skupljanje Exploit programa za isti i prikaz dodatnih svojstava važnih kao napomene za sigurnost samog sustava. Ovakvom agregacijom vlasnik sustava lako pregledava ranjivosti i ustanovljuje stanje i potrebe za nadogradnjom ili popravcima. Nabrojane funkcije imat će i sučelje "NeonEx" kao praktični dio ovog rada. NeonEx agregira sigurnosne ranjivosti koristeći API sučelja, podatkovne skupove i programsku obradu podataka kao i detekciju CMS-a na upisanoj mrežnoj stranici od strane korisnika.

14 Shodan Search Engine. URL: https://www.shodan.io/ (2022-02-13)
15 CVE Mitre History. URL: https://www.cve.org/About/History (2022-02-12)

Pronalazak ranjivosti na CMS sustavima koristeći programska rješenja

Prije pronalaska ranjivosti potrebno je shvatiti o kojim se najčešće ranjivostima u CMS sustavima radi i samim time koje će ranjivosti biti najviše spomenute. Pronalazak ranjivosti u CMS sustavima počinje jednako kao i pronalazak većine ranjivosti u programskim rješenjima: modifikacijom podataka koji su inače pruženi sustavu. Radi ove činjenice očekuje se da su za CMS najčešće ranjivosti jednake onima za bilo koje mrežno mjesto. Iako ovo je slučaj i većina ranjivosti odnosi se na XSS, RCE,SQLi ili pak DoS, CMS sustavi su nešto kompleksniji, pa ulogu igra i sama srž, instalirane ekstenzije CMS-a i slično. Analiza ranjivosti popularnih CMS-ova 2017. godine pokazuje da su većina CMS-ova nespremni za održavanje sigurnosti na svom webu i da nemaju organizirane i isplanirane procedure u slučaju novootkrivenih ranjivosti. Primjer ovoga je da 70% WordPress instalacija na internetu ima ranjivosti na hakerske napade.[16] Ove ranjivosti se dakle odnose na razne plug-inove ali i često na samu srž/jezgru CMS-a - što često predstavlja puno veće probleme. Ranjivosti jezgre su opasniji ne samo zato što više korisnika posjeduje to programsko rješenje od korisnika koji posjeduju jezgru i plug-inove, već i jer su često i ranjivosti veće opasnosti, kako jezgra ima više ovlasti od plug-ina i više mogućnosti. Dodirujući se detekcije i agregacije ranjivosti - slična platforma projektu NeonEx platforme (koja je izrađena uz ovaj rad) je platforma izrađena od autora članka za časopis *"International Journal of Advanced Trends in Computer Science and Engineering"* u broju 1.3 (vol. 9) gdje spominju „*Sneakerz*" platformu[17]. U ovom je članku, dakle, opisana i uz njega izrađena platforma "*Sneakerz*" za detekciju ranjivosti CMS-ova. Za razliku od NeonEx projekta, koji realizira platformu koja se izrađuje uz ovaj rad, nije zapisan link na kod, te nije poznato radi li se o kodu u otvorenom pristupu. Ovo je samo jedna od nekoliko razlika između ova dva projekta, iako dijele srž i ideju, izvršenje i svojstva su vrlo različiti. Platforma "Sneakerz" izrađena je koristeći Python programski jezik, posjeduje grafičko sučelje izrađeno za korištenje kao Desktop aplikaciju, dok je NeonEx izrađen kao web aplikacija koristeći PHP programski jezik. "*Sneakerz*" detektira samo određenih nekoliko CMS-ova s

16 Jerkovic, Hrvoje; Sinkovic, Branko. Vulnerability analysis of most popular open source Content Management Systems with focus on WordPress and proposed integration of artificial intelligence cyber security features. // International Journal of Economics and Management Systems, vol. 2 (2017). URL: https://www.iaras.org/iaras/filedownloads/ijems/2017/007-0010(2017).pdf (2022-02-02)

17 Web Vulnerability Assessment Tool for Content Management System. // International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 1.3 (2020). URL: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse69913sl2020.pdf (2022-03-03)

mrežnih stranica (Joomla, WordPress i Drupal), dok je NeonEx modularan i otvorenog koda te može detektirati više poznatih CMS-ova (lista navedena u prilozima). Također je uz NeonEx dodana i opisana procedura dodavanja novih sustava među postojeće sustave na temelju kojih se detektira CMS. Još jedna razlika je činjenica da "*Sneakerz*" radi "agresivno" skeniranje korisnikove web stranice, gdje pokušava pronaći ranjivosti na ubrizgavanje koda - specifično SQL ubrizgavanje, XSS napade i još nekoliko sličnih napada. Ovakav sken nazivao bi se agresivne prirode jer može oštetiti samu stranicu ili se detektirati kao maliciozan promet. NeonEx rješenje detektiranje ranjivosti radi pasivno, tako da mrežna stranica ostaje sigurna i netaknuta, sve što radi je šalje nekoliko zahtjeva kako bi NeonEx obradio podatke. Ovi zahtjevi dohvaćaju jednako stranice kao što bi dohvatio i pretraživač, stoga ni "*load*" nije zahtjevan za poslužitelja. Pošto većina korisnika želi sigurno pristupiti skeniranju svoje mrežne stranice i ne želi upozoriti vatrozide ili IPS sustave (sustave za sprječavanje upada - *Intrusion prevention*) preferiraju pokrenuti skeniranja koja će blago dotaknuti samu stranicu i server. Iako su izrađena mnoga rješenja s agresivnijim pristupom detekcije, gdje neka čak koriste skeniranje portova[18] što bi se na korisnikovom poslužitelju prikazalo kao sumnjivo ponašanje, postoji manji broj sustava za detekciju ranjivosti CMS-ova koji rade pasivno kao i NeonEx.

Od svih CMS-ova na webu WordPress zauzima veliki dio. Njegova popularnost radi količina ranjivosti podrazumijevaju i da će biti privlačan napadačima, pa se napadač prije svega priprema iskoristiti ranjivost jezgre WordPressa, a ako nema ranjivosti u njoj za određenu verziju koja ga zanima - pretražit će Plug-Inove ili Add-on-ove (dodatke) za koje će pronaći verziju i pokušati iskoristiti ranjivost. Česta ranjivost kod WordPressa je njegova XML-RPC (eXtensible Markup Language - Remote Procedude Call) autentikacija gdje napadači znaju pronaći sve postojeće korisnike, pa sve do iskoristiti ranjivosti i dobiti pristup panelu CMS-a. WordPress se bori s ovakvim ranjivostima tako što često nadograđuje verzije, pa korisniku Wordpressa pošalje obavijest kako je vrijeme za nadogradnju. Omogućava i nadogradnje često te je vrlo lako nadograditi verziju (većinom se svodi na klik gumba), no istraživanja pokazuju da velika količina stranica na internetu ima zastarjeli CMS. Neovisno radi li se o WordPressu, ovo je zabrinjavajuće i ovakve problematike se dotiču platforme poput NeonEx stranice. Problematika kod nadogradnji je moguća nekompatibilnost s dodacima, moguć gubitak podataka ili dijela podataka i slični rizici,

18 Asaduzzaman, Md.; Rawshan, Proteeti Prova; Liya, Nurun Nahar; Islam, Muhmmad Nazrul; Dutta, Nishith Kumar. A Vulnerability Detection Framework for CMS Using Port Scanning Technique, 2020. URL: https://easychair.org/publications/preprint_download/DpgN (2022-03-04)

pa skepticizam vlasnika stranice nije neočekivan. Bez obzira, velika većina kreatora CMS-ova potiče korisnike da uvijek nadograđuju svoj CMS kako se ne bi našli žrtvom na primjer masovne eksploatacije, napada ubrizgavanja malicioznog SQL koda ili čak RCE (izvršavanje koda na poslužitelju s udaljenog računala) napada koji omogućuje lako preuzimanje kontrole nad cijelim serverom.[19]

Osiguravanje CMS-ova

Provedeno je istraživanje na području Hrvatske gdje su ispitani zaposlenici velikih IT poslovnih subjekata od kojih 33.33% odbija ili ne zna odgovoriti koriste li SSL protokol (za šifriranje mrežnih konekcija), 36.36% koristi SSL i preostalih 30.30% nikad nije koristilo za njihovu stranicu ili trenutno ne koristi SSL. SSL je važan dio internet sigurnosti ako se radi o bilo kakvom prijavljivanju, registriranju ili razmjeni osjetljivih podataka na stranici. Nažalost, niti jedan CMS ne garantira korištenje SSL-a iako bi ovo bio izvrstan način za osiguranje paketa koji putuju u posluživanju i generalno razmjeni podataka. Također, većina poslovnih subjekata (60.60%) provjeravaju jeli verzija programskog rješenja, komponente ili ekstenzije (dodatka) kojeg koriste zastarjela, ali isto tako 36.36% ovih organizacija ne prati nove ranjivosti za programsko rješenje s kojim rade.[20] Neki od izvora tvrde kako su većina ranjivosti pronađena u Plug-inovima i ekstenzijama CMS-ova, a ne u samoj srži ili čak nepažnji administratora stranice.[21] Ranjivosti se plug-inova mogu popraviti također nadogradnjom plug-ina na noviju verziju, no ponekad se postavlja pitanje kompatibilnosti te verzije s CMSom.

Jedan od online resursa koji služi za legalne svrhe i istraživače u području kibernetičke sigurnosti da se zaštite od eksploatacija ili da provedu RedTeam napade (primjer penetracijskog testiranja) je poznati Exploit Database. Exploit Database ima i stranicu “Google hacking database” koja

19 Acunetix. Web Application Vulnerability Report, 2016. URL: https://www.it-sa.de/EDB/EDB3/LOADPRESSINFO/2721 (2022-03-05)

20 Kaluža, Marin; Vukelić, Bernard; Rojko, Tamara. Content management system security. // Zbornik Veleučilišta u Rijeci, Vol. 4 (2016). URL: https://hrcak.srce.hr/file/236346 (2022-03-05)

21 Martinez-Caro, Jose-Manuel; Aledo-Hernandez, Antonio-Jose; Guillen-Perez, Antonio; Sanchez-Iborra, Ramon; Cano, Maria-Dolores. A Comparative Study of Web Content Management Systems. // Information vol. 9, no. 27 (2018). URL: https://www.mdpi.com/2078-2489/9/2/27/pdf (2022-02-07)

ispisuje sve moguće parametre za pretraživanje Google tražilice koji prikazuju ranjive sustave.[22] Napadač će dakle preuzeti odgovarajući upit i unijeti ga u Google tražilicu.

Ponekad će i dodati odgovarajuću državu, na primjer da pronađemo sve rezultate za zastarjelu verzije WordPress CMS-a 4.0 s Hrvatskom domenom upit ćemo formirati ovako:

```
"WordPress 4.0" site:.hr -github
```

Ovakav upit vratit će sve stranice koje završavaju u nazivu s ".hr", eliminirati sve rezultate s koji uključuju riječ "Github" (kako bi se eliminirali prikazi programskog koda, te samo prikazali sustavi koji su otvoreni internetu i pokreću WordPress 4) i najvažnije pretražiti stranice koje u sebi sadrže tekst "WordPress 4.0". Vidljivo je i da će ovo prikazati i web stranice koje samo spominju WordPress 4.0 - no moguće je radi toga pružiti i dodatne parametre poput inurl:wp-content iako će ovo suziti pretraživanje na rezultate koji prikazuju direktorij /wp-content, ali to ne predstavlja problem jer većina WordPress stranica ostavlja Wp-content vidljivim (čak i kad im se ne može pristupiti) kako bi se preciznije odredili ranjivi sustavi. Zaključno je dakle da se lančani napadi mogu ostvarivati lako kad se koristi zastarjela verzija CMS-a, te da isti lančani napadi mogu biti bilo kakve vrste i iskorištavati bilo kakvu ranjivost da bi svu kontrolu nad ranjivim sustavima preuzeli i umrežili.[23] Ideja autora ovog rada je da se isti parametri koriste za detekciju ranjivih stranica kako bi se na vrijeme javilo vlasniku stranice da je potrebna nadogradnja - ovaj dio posla izvršavao bi nacionalni CERT (Computer Emergency Response Team - https://www.cert.hr/). Australski nacionalni centar za kibernetičku sigurnost predlaže korištenje "Managed CMS hosting" infrastrukture za razliku od običnog posluživanja na poslužitelju i instaliranja CMS-a "out of the box" jer upravljani CMS hosting nudi bolju potporu i precizan "Patching" (zakrpavanje). Također predlažu upravljanje samim zakrpama i svim plug-in-ovima koje CMS koristi, a ovo se proteže i sve do razine operacijskog sustava i njegovih nadogradnji kako bi se postigla potpunija sigurnost. Što se uspostave web sigurnosti tiče preporučena je česta i procjena ranjivosti CMS-

22 Exploit database Google hacking database. URL: https://www.exploit-db.com/google-hacking-database (2022-02-20)

23 Kumar, Vinoth R.; Kumar, Kishore K. Exploitation of content management system vulnerabilities to launch large scale cyber attacks. // International Journal of civil engineering and technology, vol. 8, no. 10 (2017). URL: https://iaeme.com/MasterAdmin/Journal_uploads/IJCIET/VOLUME_8_ISSUE_10/IJCIET_08_10_141.pdf (2022-03-02)

ova, upravljanje računima i uspostava sigurnih računa za upravitelje CMS-a kao i nadgledanje samih CMS instalacija za svim promjenama.[24]

24 Australian cyber security centre. Securing Content Management Systems, 2015. URL: https://www.cyber.gov.au/sites/default/files/2021-10/PROTECT%20-%20Securing%20Content%20Management%20Systems%20%28October%202021%29.pdf (2022-03-08)

Platforma za agregiranje ranjivosti CMS-ova

Agregiranje ranjivosti na bazi URL-a i prikaz svih mogućih ranjivosti kroz sučelje, svih exploita i više informacija o ranjivostima zahtijevat će korištenje mnogih API sučelja i podatkovnih skupova. Korisnik će, dakle, pružiti platformi URL svoje stranice koju želi skenirati, te će zatim dobiti klikom ispis svih potencijalnih ranjivosti, CVE brojeva, potencijalnih exploita kao i verziju i vrstu CMS-a koji je pokrenut na web stranici (na temelju ovih podataka će se inicijalizirati detekcija ranjivosti). S obzirom na zakonska ograničenja skeniranje stranice za ranjivosti obavlja se na vrlo pasivan način. Najzahtjevnija procedura za poslužitelja u cijelom procesu detekcije je slanje više zahtjeva na stranicu, stoga se pretpostavlja da su većine mogućih poteškoća eliminirane. Ovi zahtjevi dohvaćaju dokumente jednako kao što ih dohvaća obični pretraživač, no u ovom slučaju se programski čitaju kroz kod te se pretražuje verzija za odgovarajuće CMS-ove, kako bi se na temelju toga pretražili CVE brojevi i ranjivosti. Platforma koja je izrađena za ovaj rad ima naziv NeonEx. Osim detektiranja CMS-a i na temelju njega ranjivosti - platforma nudi i pregled dodatnih opcija ranjivosti u odnosu na programsko rješenje, plug-inove (dodatke) i predlaže exploite (programe za eksploataciju ranjivosti) za odabrano programsko rješenje. Sve ispunjavanjem jedne forme i jednim klikom. Ovakva platforma potrebna je istraživačima na području kibernetičke sigurnosti, posebno ako je kod u otvorenom pristupu, jer ju istraživači mogu koristiti svakodnevno radi svoje jednostavnosti i minimalne intruzije i agresivnosti. S obzirom na modularnost i izmjenjivost obogaćenu kroz primjere, NeonEx dozvoljava razne nadogradnje od detekcije novih CMS-ova do detekcije ranjivosti, verzije ili prikaza exploita (kao i nadogradnje podatkovnih skupova koje koristi). Naravno, ovu platformu ne moraju koristiti samo istraživači na području crvenog ili plavog tima (crveni tim napada sustav, plavi ga brani) kibernetičke sigurnosti, ona može poslužiti i za Bug Bounty programe (lov na greške u programskim rješenjima), analizu vlastitih mrežnih mjesta na ne-agresivan način, za CTF (Capture the flag) natjecanja, a možda i za pronaći 0-day ranjivosti (nove, neistražene ranjivosti).

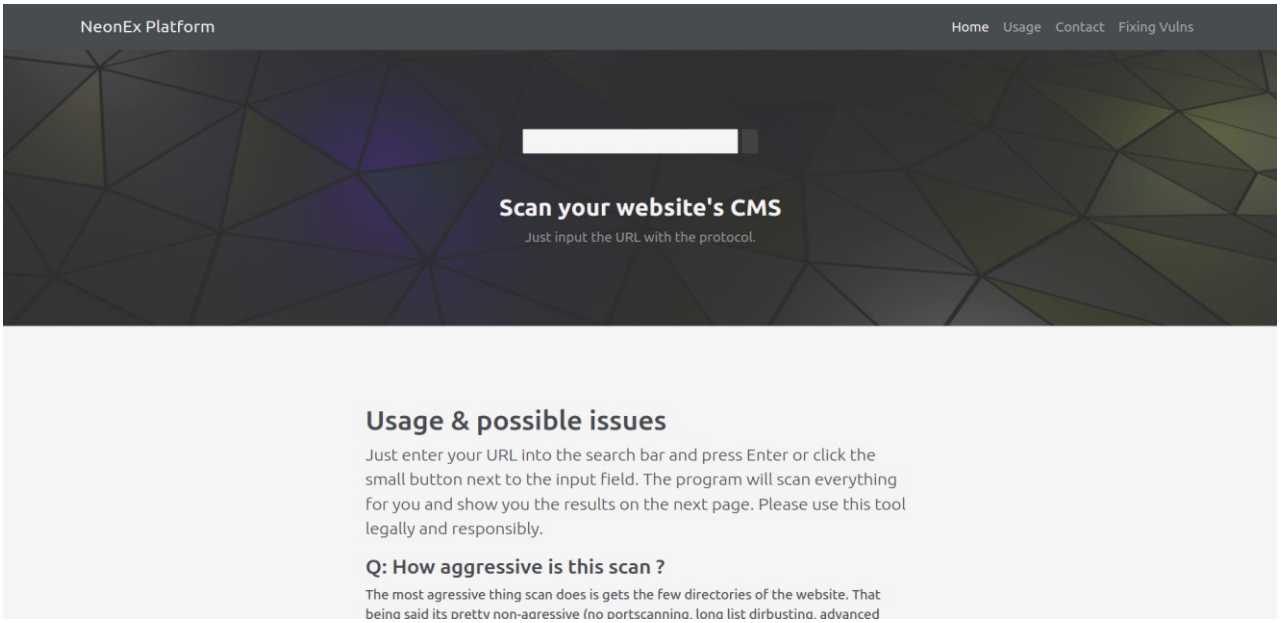
Dizajn sučelja i izrada

NeonEx poslužuje jednostavnu pozadinsku sliku na kojoj stoji polje za tekstualni unos i gumb koji se može aktivirati klikom, ili pritiskom na Enter tipku tipkovnice. Sučelje za NeonEx platformu oslanja se na vrlo jednostavni bootstrap CSS i HTML5 kod. Na ovaj način štedi se na performansama i pruža korisniku jednostavan pristup svemu bez kompliciranih menu-a i pop-

upova, no performanse ovise i o podatkovnim skupovima koji se konstantno šire, pa je ponekad potrebno sačekati par sekundi. Najvažnije dvije stranice su početna stranica i ona koja prikazuje rezultate na temelju upisa kojeg pruža početna. Početna stranica sadrži jednostavan dizajn sa zaglavljem koji izgleda kao alatna traka, ali sadrži samo naslov projekta i mogućnost povratka na početnu stranicu kad se ode na bilo koju drugu. Ispod toga, kao što je spomenuto ranije, nalazi se slika koja na sredini ima mogućnost unosa URL-a za korisnika te mali gumb. Ispod slike nalaze se dodatne instrukcije i pitanja koje bi korisnik mogao imati te odgovori na njih. Stranica koja prikazuje rezultate u segmentima prikazuje razne rezultate kako bi bili korisniku pregledni. Početni izgled podijeljen je na dva dijela te je jedina stvar koja je vidljiva od prošle stranice zapravo naslov unutar stila navigacijske trake. Moguća je i implementacija konstantno vidljivog polja za unos, no ovo bi spadalo u noviju verziju programa, a i preglednost bi bila manja. Što se estetike tiče sučelje je izrađeno da je jednostavno, ali ne i pre monotono. Odabrane boje i slike daju dojam elegantnijeg sučelja orijentiranog "cyberpunk" (zamišljanje moderne, tehnološki orijentirane budućnosti)[25] potkulturi slikama, a minimalističkom smjeru ostatkom dizajna grafičkog sučelja.

Podjela sučelja na segmente ne omogućuje samo preglednost i jednostavnost, već na suptilan način pokazuje korisniku na kakve je segmente podijeljen projekt. Na ovaj način korisnik može pregledati kod i odmah znati koji mu dio treba te pronaći dio koji želi izmijeniti. Ovo je vrlo važno jer je cijeli projekt NeonEx koncentriran na dozvoljavanje izmjena i dodavanje mogućnosti od strane korisnika pa bi sama čitljivost igrala važnu ulogu. Razlog za ovo je, stoga, želja za razvijanjem i napredovanjem u području kibernetičke sigurnosti i informacijske tehnologije, ali i interes za kreativna rješenja za unaprjeđenje koja korisnici možda imaju. Ovako će se osigurati konstantan napredak sučelja, ne samo u smislu opcija koje nudi i svojstava koja posjeduje, već će i kod, dohvaćanje podataka i obrada postati boljima.

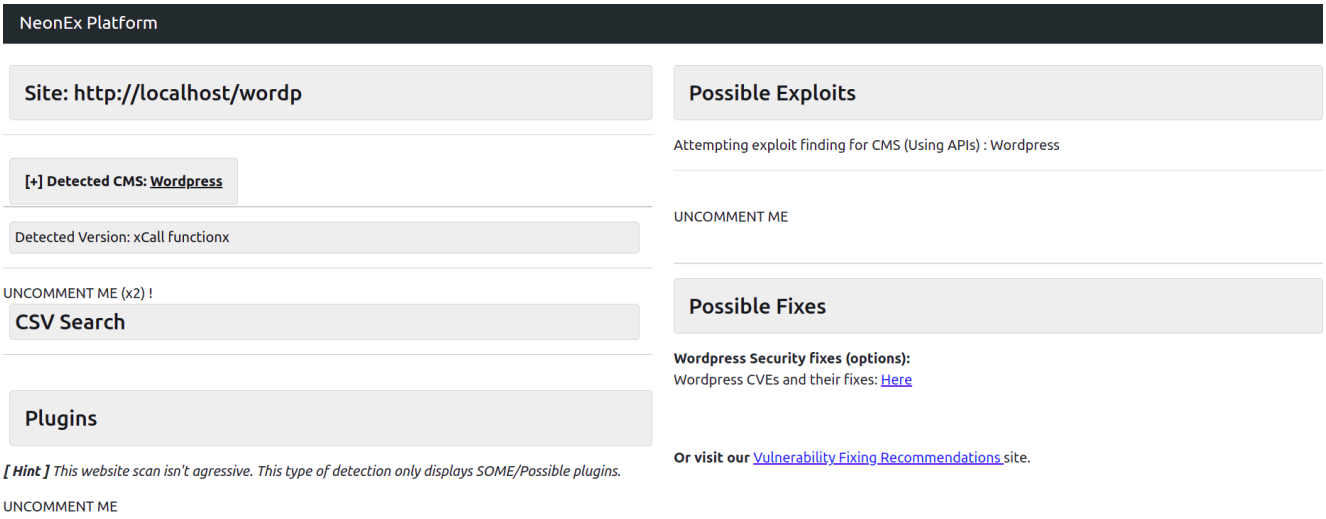
25 Stanić, Ivan. Visions of the Future in Contemporary Cyberpunk, 2021. URL: https://zir.nsk.hr/islandora/object/ffzg:4432/datastream/PDF/view (2022-02-10)



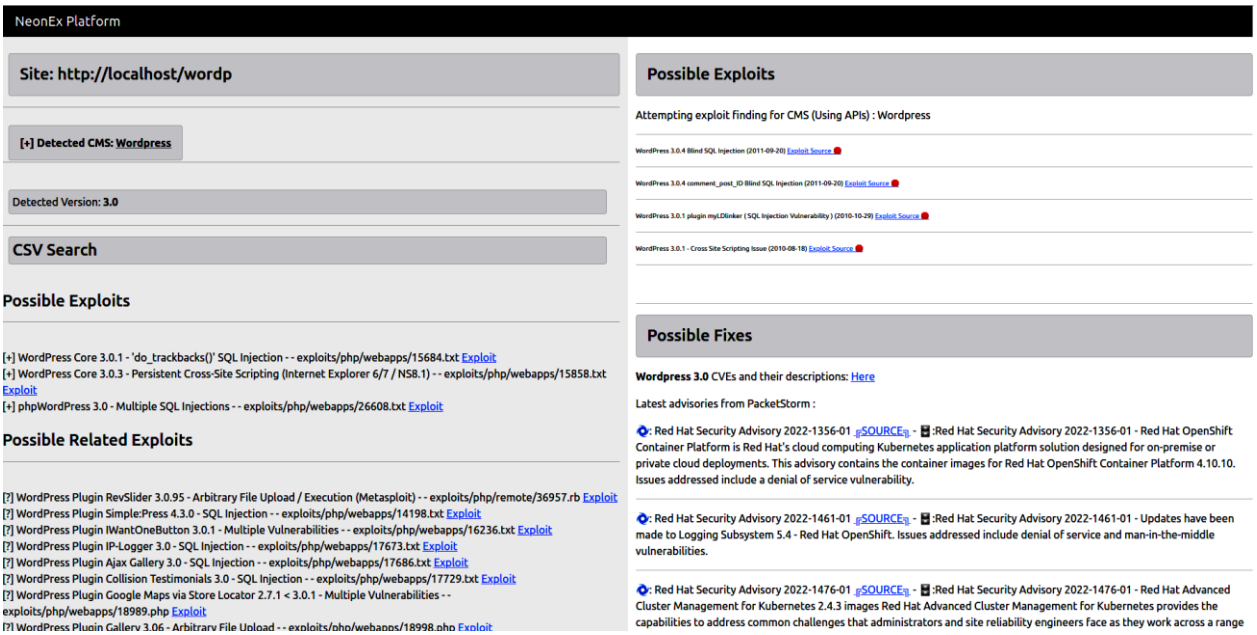
Slika 1. Naslovna stranica NeonEx platforme

Prikaz rezultata na sučelju

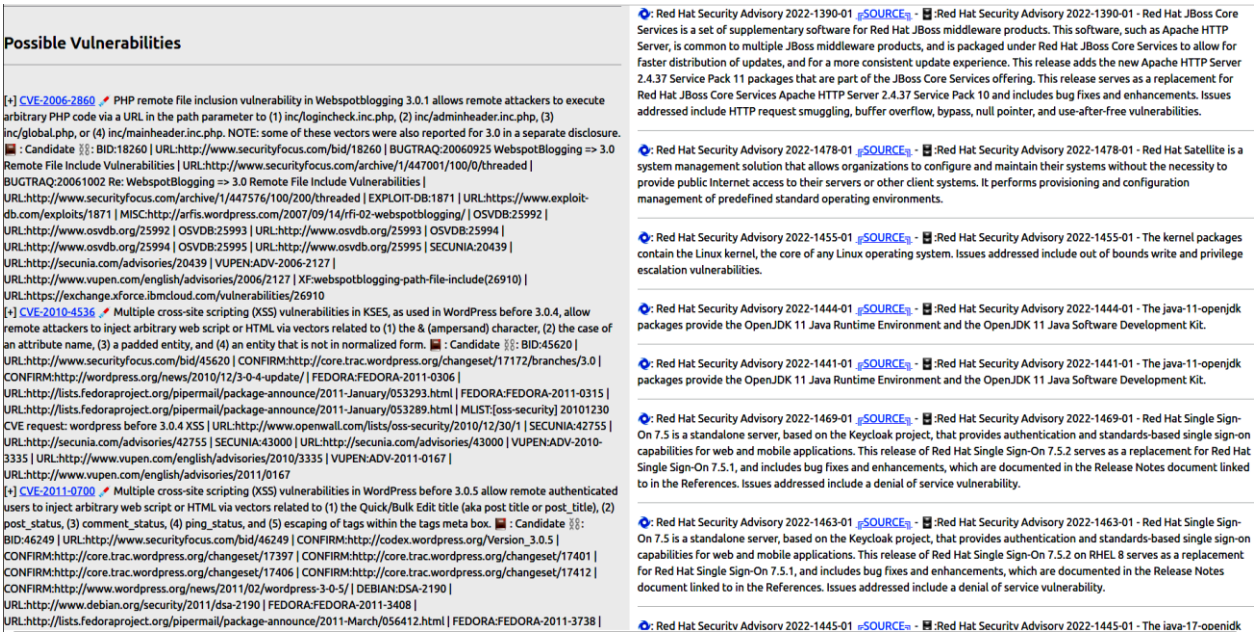
NeonEx nakon unosa vodi korisnika na novu stranicu gdje pomoću pruženog URL-a obrađuje podatke i na ekran korisniku vraća, podijeljene u kategorije, strukturirane rezultate. Ovi rezultati podijeljeni su po kategorijama kako bi bili uočljiviji korisniku i prikazali sve potrebno u organiziranom formatu na kojeg se korisnik može uvijek osloniti. Ovisno o tom koji su mu potrebni podaci korisnik može gledati u zasebne dijelove ekrana i s brzinom dohvatiti odgovarajuću informaciju poput ranjivosti i URL-a do nje. Naslovi kategorija prikazani su većim slovima te sam tekst manjim, a URL-ovi su skraćeni većinom jednom riječju kako bi stranica bila više pregledljiva. Postoji i problematika (pri neuspjehu detekcije CMS verzije) gdje se ispisuje previše rezultata jer se šalje širi upit (npr. umjesto “WordPress 3” upit će biti “WordPress”) no svi rezultati u svakom slučaju moraju biti prikazani, stoga je potrebno da sučelje ispisuje sve - tako da su svi rezultati uvijek prikazani bez obzira o upitu. Što se tiče rezultata koji su izvan stranice postoje nekoliko linkova na eksternalne stranice na formirani na temelju podataka.



Slika 2. Prikaz sučelja bez većine modula

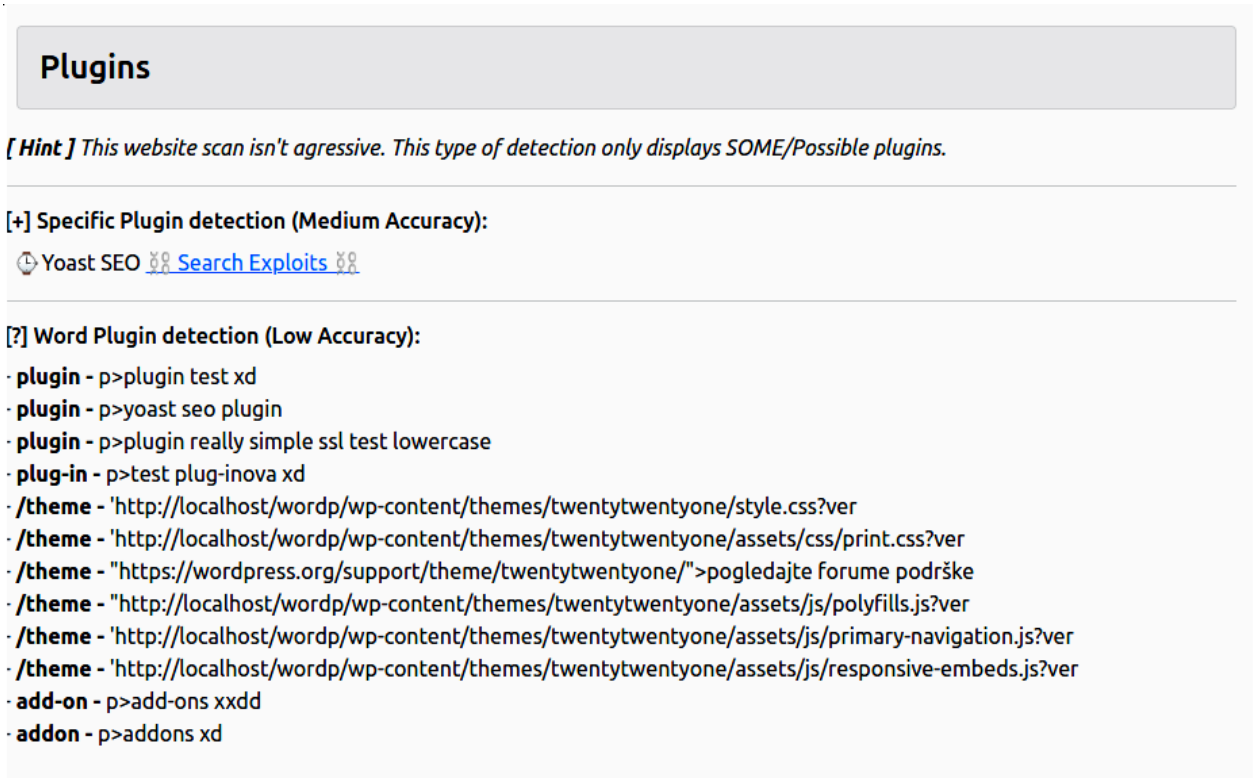


Slika 3. Prikaz sučelja s rezultatima na upit WordPress stranicu



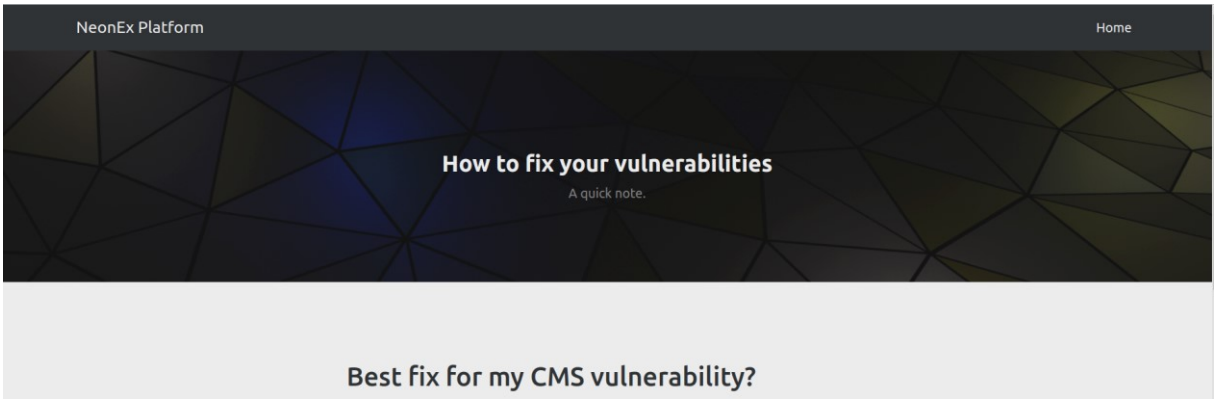
Slika 4. Prikaz sučelja s rezultatima na upit WordPress stranice (ostali podaci)

Slika 5. Prikaz sučelja s rezultatima na upit WordPress stranice (ostali podaci)



Slika 6. Prikaz sučelja s rezultatima na upit WordPress stranice (ostali podaci)

U priloženim slikama prikazano je prije svega sučelje koje detektira CMS, ali ne obrađuje nikakve druge podatke te samo prikazuje izgled segmenata za podatke. Nakon toga prikazan je primjer upita “WordPress 3.0” gdje se detektira CMS i verzija, obrađuju podaci i prikazuju se ranjivosti za odgovarajuću verziju uključujući i exploite te opcije popravaka ranjivosti i preporuke za iste. Radi opširnosti rezultata na iduće tri slike prikazani su dodatni rezultati kao i poglavlja koja nije bilo moguće prikazati na prijašnjoj slici. Posljednja od navedene tri slike prikazuje dvije vrste detekcije dodataka i nadodatke koji su pronađeni kao i URL na skripte za iskorištavanje ranjivosti istih.



Slika 7. „Fixing vulns" stranica

Nakon pristupa rezultatima skeniranja, ako korisnik ne pronađe popravke koji mu odgovaraju niti preporuke može posjetiti klikom gumba „Vulnerability fixing recommendations" u segmentu „Possible Fixes" gdje će dobiti opće napomene za popravke ranjivosti CMS-ova.

Detekcija CMS-a

Detekcija CMS-a vršit će se koristeći skriptu preuzetu kao račvanje s Github repozitorija; korisnika Krisseck. Ova skripta bit će modificirana za odgovarajuću PHP verziju i XAMPP program za pokretanje poslužitelja i samog PHP-a. (Modificirana verzija: https://github.com/ffos-user-57/Detect-CMS). Skripta detekcije prima URL kao parametar i prema njemu vrši detekciju slanjem zahtjeva i pregledavanjem sadržaja. Kad skripta vrati pronađeni CMS, NeonEx sučelje će zahtjevati podatke od raznih izvora kako bi se korisniku pružile bogate informacije poput na primjer ranjivosti za određeni CMS. Sama detekcija CMS-a kreirana je za nekoliko sustava: Bohemiasoft, Joomla, Moodle, Typo3, Concrete5, Laravel, Shopsys, vBulletin, Drupal, Liferay, Shoptet, Webgarden, EshopRychle, Magento2, Sitecore, Webnode, ExpressionEngine, Magento, Squarespace, WordPress, Fastcentrik i Modx. Kao primjer dodavanja nove skripte koja opisuje sustav, za detekciju CMS-a, dodana je skripta za Textpattern CMS i u kod skripte za detekciju CMS-a dodana je linija da se i sama Textpattern skripta uzme u obzir..

Detekcija verzija

S obzirom da je datoteka za detekciju CMS-ova preuzeta s Github repozitorija ograničena je u mogućnostima - ne posjeduje detekciju verzija. Potrebno je izraditi programske skripte koje

će to obaviti, te prikazati primjere dodavanja dodatnih sustava u programsko rješenje za detekciju. Skripta za detekciju verzija mora biti modularna i programirana kako bi odgovarala za svaki od CMS sustava, stoga će morati biti djelomično neutralna i pretraživati ključne riječi. Iz ovog razloga točnost opada, no rezultat će i dalje u većini slučajeva biti točan. Neke od stranica sakrivaju podatke o verziji radi sigurnosti, a ponekad i podatke o CMS-u, no ovo je puno teže sakriti pa će se sam CMS detektirati u većini slučajeva, dok detekcija verzije ponekad ovisi o stranici. Broj ili ime verzije će se zatim pružiti ostalim skriptama koje će dohvatiti dodatne podatke na temelju CMS-a i njegove verzije. Primjer ovakvih podataka su dohvaćanje ranjivosti za "WordPress 3.0".

Kreiranje novih modula za postojeće programsko rješenje

Postojeće programsko rješenje s Github repozitorija (otvorenog izvornog koda) sadrži skripte nekoliko naj popularnijih sustava za uređivanje sadržaja (CMS) poput WordPress, Drupal, Joomla, Moodle, Squarespace i slične CMSove (Lista svih CMS-ova koje pokriva detekcija u prilozima). Ove skripte su izrađene da se na temelju URL-a pruženog klasi pregledavaju sadržaji stranice raznim zahtjevima neovisno o vrsti stranice to jest njenom sustavu za uređivanje sadržaja. Cilj skripte je da na temelju URL-a provjeri CMS i vrati točan rezultat o kojem se CMS-u radi (ali ne uključuje podatke o verziji). Iako je pokrenuto testiranje na lokalnim instancama dva naj poznatija CMS-a (WordPress i Joomla), ponekad je potrebno unijeti neki novi CMS u listu sustava kako bi se implementacijom u NeonEx platformu korisniku pružila šira mogućnost pogađanja i dalje analize. Prvi korak je dodavanje nove skripte, koja je najčešće kopija stare s izmijenjenim imenom i nazivima u kodu, te dodavanje imena nove skripte u listu glavne datoteke (lista u DetectCMS.php koja sadrži sve izlistane sustave u mapi „Systems"). Nakon ovih dodavanja, te praćenja normi imenovanja datoteka prema sustavu kojeg opisujemo u datoteci, potrebno je izmijeniti kod tako da se zaista pronađe točan sustav. Potrebno je provjeriti sam CMS koji želimo dodati te pronaći mjesta gdje će se neovisno o verziji nalaziti njegov naziv, ali imati na umu da bi se isti taj naziv mogao spomenuti na bilo kojoj mrežnoj stranici u tekstu, a da ta stranica ne pokreće CMS koji želimo detektirati (primjer članka koji opisuje najbolje CMS-ove za e-commerce). Jedan pristup je oslanjati se na potpune rečenice s URL-ovima koji napominju da je stranica izrađena koristeći određeni CMS, no ovaj pristup nije dovoljan niti savršen. U izvornom kodu naslovne stranice može se također pronaći svašta korisno. Na primjer meta oznake (tagovi) koji često definiraju točno o kojem se CMS-u radi, npr. *<meta name="generator" content="Textpattern CMS">*. Ova oznaka znači da se najvjerojatnije i pokreće TextPattern CMS na web stranici koju

pregledavamo. Da bi to dokazali možemo posjetiti razne pod-direktorije ove stranice poput „*/README.txt*“. U ovoj datoteci ponekad možemo u samom naslovu potvrditi da se zapravo radi o CMS-u koji tražimo. U obzir treba i uzeti scenarij gdje ova datoteka ne postoji, pa se smanjuje točnost detekcije ili se jednostavno koristi drugi način detekcije koji je također napisan u skripti. Na kraju je potrebno pokrenuti test za novo dodani CMS, te za sve stare kako bi se ustanovilo da nije došlo do greške pri dodavanju novog koda. Kad sve radi ispravno, postoji mogućnost odabira postavljanja koda na Github kako bi originalni kreator znao da su u ovoj verziji dodana svojstva s naše strane. U svakom slučaju, sada je detekcija Textpattern CMS-a spremna.

Spajanje na API sučelja

Ovisno o vrsti API (aplikacijsko programskog) sučelja i tipu zahtjeva kojeg traži, kroz PHP programski kod moguće je na više načina doći do podataka. Dva osnovna pristupa su standardni PHPov zahtjev za dohvaćanje, te zahtjevi koristeći CURL modul koji omogućuju specifičnije, ali i šire zahtjeve. Ovakvi pristupi omogućuju dohvaćanje podataka preko mnogo API sučelja, što je u ovom projektu i potrebno. Dohvaćeni podaci najčešće su oblikovani u JSON (JavaScript Object Notation)[26] formatu kojeg je potrebno učitati u kod, te kroz funkcije pojasniti kodu kako da upravlja s vrijednostima koje JSON daje kroz oblik “key” i “value”. Pozivom na svaki “key” dohvaća se svaki “value” ili vrijednost koju tražimo. Primjer “key:value” odnosa bio bi “ime:Ivan”. Nakon spajanja na API sučelje podaci se obrađuju i samo potrebni za sučelje prikazuju. Obrađeni podaci ponekad su proslijeđeni drugom API sučelju kako bi se na lakši način došlo do drugih podataka. Primjer ovog je prosljeđivanje ID broja s jednog API sučelja, jer ID broj označava kraj URL-a na drugom API sučelju, što dozvoljava formiranje linka na koji korisnik sučelja može kliknuti da dođe do šireg opisa ranjivosti. Bez API sučelja ili internet konekcije platforma i dalje pretražuje svoje podatkovne skupove i prikazuje ranjivosti. Obe opcije su dodane radi raznolikosti rezultata iz raznih izvora i radi pouzdanosti rada sučelja.

26 Introducing JSON (Javascript object notation). URL: https://www.json.org/json-en.html (2022-02-17)

Čitanje CSV datoteka (ExploitDB)

U procesu implementacije pretrage exploita (programa za iskorištavanje ranjivosti programskih rješenja) za naše CMS-ove potrebno je imati naziv CMS-a i njegovu verziju kako bi se isti pružili PHP skripti koja će pretražiti .csv (datoteke gdje su podaci razdvojeni zarezom, s imenima podataka na prvoj liniji i samim podacima na svim ostalim linijama)[27] datoteke. CSV datoteke koje će biti korištene za exploite bit će preuzete od exploit-db stranice, a kao alternativni način dohvaćanja Exploita bit će korištenje API sučelja za exploite. Ranije spomenuta baza podataka koja sadrži exploite "Exploit DB" sadrži razne exploite koji su pretraživi upisivanjem imena programskog rješenja, ili u ovom slučaju imena i verzije. Dakle pomoću PHP skripte koja je izrađena, a kojoj je dodana funkcija čitanja .csv datoteka, pretraženi će biti rezultati koji odgovaraju na upit koji izgleda ovako: "wordpress 3.0" - te će se zasebno pretražiti riječ "wordpress" u svim linijama .csv datoteke, a zasebno "3.0" kao verzija. Razlog ovog je činjenica da kao odgovor moraju biti dostupne samo one verzije WordPressa za koje je poslan upit, kako korisnik ne bi dobivao previše nepotrebnih informacija, ali je i poželjno priložiti exploite za ostale verzije kako bi se korisnik bolje informirao ukoliko traži nove ranjivosti. No ovdje se javlja problem. Novije verzije koje su ranjive, a označene "<4.0" podrazumijevaju da su sve ranije verzije također ranjive, što uključuje 3.0 verziju koju je korisnik zatražio, a to se neće uvijek prikazati. Iako postoje elegantnija rješenja, s obzirom na nasumičnost upita od strane korisnika, te znatiželjnost s obzirom na ostale verzije (a i činjenicu da se nekad verzije ne definiraju brojkama već riječima) prikazani su svi rezultati za upit "wordpress", a istaknuti oni koji imaju odgovarajuću verziju. Na ovaj način korisnik između ostalog ima opciju pregledati povijest ranjivosti, što dovodi dodatne korisnike za NeonEx platformu, jer Bug Bounty istraživači (tražitelji greški za koje budu plaćeni od strane vlasnika programskog rješenja) uvijek gledaju prijašnje ranjivosti kako bi pronašli nove. PHP skripta, dakle, čita .csv datoteku i vraća odgovarajuće rezultate koji se ukomponiraju u sučelju.

Pretraživanje exploita

Pretraživanje exploita koristeći upite poput "wordpress 3" omogućilo je više izvora. Jedan od izvora je "exploitAlert" API sučelje koje omogućuje ovakve upite i vraća vrijednosti u JSON

27 Shafranovich, Y. Common Format and MIME Type for Comma-Separated Values (CSV) Files, 2005. URL: https://datatracker.ietf.org/doc/html/rfc4180#section-2 (2022-03-16)

formatu, gdje su ime exploita, datum objave i ID broj. Dobiveni ID broj koristi se za dohvaćanje dodatnih podataka o exploitima sa stranice Exploit-Db gdje najčešće stoji i sam programski kod ili program za iskorištavanje te ranjivosti. Radi ovog programer koristi API sučelje da dohvati ID i uključuje ID u URL Exploit-Dba gdje korisnik klikom dohvaća odabrani exploit. Druga opcija je pretraživanje preuzete .CSV datoteke koja sadrži sve exploite. Problem kod ovog je jednak kao i pretraživanje bilo kakvog podatkovnog skupa s lokalnog računala: ubrzo će ga zamijeniti najnovija verzija, pa je važno omogućiti i program koji sam nadograđuje verziju podatkovnog skupa ili slične opcije kako bi pretraživanje uvijek donosilo i nove rezultate. Ovaj program, to jest skripta, izrađen je za Linux sustave, a vrlo slična skripta može raditi na Windowsu (informacije za Windows pružene u komentarima za Linux skriptu). Oba primjera dohvaćanja exploita korištena su u NeonEx platformi, jedan je, dakle, prikazan gore u poglavlju spajanja na API sučelja, gdje se GET zahtjevom dohvaćaju podaci i obrađuju, a drugi je pretraživanje CSV datoteke, poput onog u poglavlju “Čitanje CSV datoteka”.

Dohvaćanje CVE broja na temelju imena sustava

Dohvaćanje CVE broja (broja oznake ranjivosti) potrebno je jer se na temelju CVE broja dohvaćaju mnogi različiti opisi ranjivosti i podaci na temelju kojih će korisnik uspostaviti mišljenje o kritičnosti i shvatiti gdje se nalazi slabost koja bi se mogla popraviti. S obzirom da kroz PHP kod ime sustava za uređivanje sadržaja (CMS-a) dolazi detekcijom kroz uvjete u dio koda koji mora vratiti CVE broj za dalji prikaz, potrebno je pobrinuti se da se koristi CSV datoteku ili API sučelje gdje se na temelju unosa npr. “WordPress 3” vraćaju svi CVE brojevi koje odgovaraju za WordPress 3. Vrlo je lako dohvatiti putem sučelja za pretraživanje CVE brojeva ove rezultate, no s obzirom na legalnost i upitnost djelovanja “scrape-anja” (preuzimanja podataka sa stranice programskim pristupom bez da je za to stranica namijenjena) na samu stranicu bolje je koristiti odgovarajuće API sučelje ili preuzetu CSV datoteku. Odabrano rješenje je preuzimanje datoteke popisa svih CVE brojeva (CSV datoteka) koja će se čitati kroz programski kod i po njoj će se tražiti CVE brojevi. Problem kod ove datoteke je činjenica da se često mijenja, pa ju je potrebno nadograđivati. Za nadogradnju je izrađena posebna skripta koja preuzima najnoviju verziju ove liste i preimenuje ju da odgovara kodu.

Kao alternativa uspostavljena je i opcija dohvaćanja putem API sučelja za CVE brojeve. Što se alternative za pretraživanje exploita tiče postoji opcija vrlo nježnog scrape-anja stranice.

Nadogradnja podatkovnih skupova

Skripta za nadogradnju podatkovnih skupova NeonEx platforme dostupna je za Linux distribucije i nadograđuje datoteke kad je pokrenuta u odgovarajućem direktoriju. Izmjene skripti (kao dodavanje "try" i "except" opcija) i izrada Windows ili MacOS verzije skripte ostaje u rukama korisnika ili budućih izmjena i verzija koda. U komentarima skripte napisane su i napomene kako bi komanda izgledala u Windows okružju kako bi se korisniku olakšala instalacija ili pisanje Windows skripte. Održavanje NeonEx platforme i nove verzije u planu su nakon objave prve verzije.

Detekcija dodataka (Plug-Inova)

Uzimajući u obzir veliku količinu ranjivosti koje donose plug-inovi (dodaci) za CMS-ove, potrebno je i dodati u NeonEx bar neki oblik detekcije plug-inova, add-onova i ekstenzija. Ovakva detekcija dat će specifičniji opis sigurnosnog stanja mrežnog mjesta i pružiti generalno bolju uslugu korisniku. Način na koji je izvršena detekcija dodataka radi raznolikosti njihove evidencije je jednostavno pretraživanje izvornog koda stranice za razne upite. Ukoliko ovi upiti vrate rezultate postoji šansa da se radi o instaliranim dodacima koje je potrebno provjeriti. Ova detekcija nije idealna, te će nerijetko predložiti krivu riječ kao plug-in, no ovdje korisnik sam raspoznaje što mu je potrebno. S obzirom da postoji potreba za pretraživanje ranjivosti plug-inova dodana je i opcija pretraživanja na temelju pronađenog imena plug-ina. Postoji i problematika detekcije verzije za plug-inove, no tog će se dotaknuti iduća verzija koda ili će korisnici doprinijeti na Githubu.

Zaključak

Prikazom znanstvenih radova, obradom podataka i izradom sučelja koje je dokazalo da se na temelju CMS-ova mogu pronaći mnoštva ranjivosti koja web čine nesigurnijim, predočena je potreba za sigurnosnim popravcima i zakrpama CMS-ova - posebice najpoznatijih. Isto tako pruženi su dokazi da je za vlasnike mrežnih stranica s CMS-ovima potrebna provjera ranjivosti CMS-ova u što češćim vremenskim intervalima. S obzirom na eksponencijalni rast poznatosti CMS-ova i samog weba, kibernetička sigurnost mora se orijentirati prema CMS-ovima sada više nego ikada. S obzirom da područje kibernetičke sigurnosti ne može pokriti cijeli svijet CMS-ova i njihovih nadogradnji na sigurnije verzije, potrebno je podizanje svijesti i mrežnim administratorima. Ovdje može pomoći platforma koja detektira ranjivosti stranice, ali na pasivan način kako bi administrator bio siguran da mu stranica neće biti oštećena. NeonEx sažima i agregira ranjivosti, obrađuje podatke i neinvazivno dohvaća sve potrebno sa stranice kako bi se mogla izvršiti detekcija i kako bi se na temelju nje pripremili ostali rezultati. Ideja je korisniku dati široki pregled stanja i pružiti opširne podatke na temelju kojih će se uspostaviti sigurniji sustav. Uz to, radi opširnosti prikazanih podataka na samom sučelju, potencijalni korisnici su i oni koji traže greške u najnovijem ili starijem sustavu, koji ga pokušavaju hakirati kako bi se ranjivost prijavila i sustav ojačao. Uz ovaj rad i prikaz izvornog koda u otvorenom pristupu, zajednica istraživača u području kibernetičke sigurnosti s nišanom na sigurnosti CMS-ova ima platformu i izvrstan temelj za korištenje ili nadogradnju. Cilj NeonEx sučelja je učiniti stranice s CMS-ovima sigurnijima radi opasnosti napada koji su često i lančani te radi količine stranica koje koriste CMS-ove (najčešće zastarjele). Prikazana platforma za agregiranje ranjivosti i detekciju CMS-ova poboljšava sigurnost u web prostoru i podiže svijest o sigurnosti CMS-ova.

Literatura

1. BuiltWith. CMS Usage Distribution on the Entire Internet: Distribution for websites using CMS technologies, 2022. URL: https://trends.builtwith.com/cms/traffic/Entire-Internet (2022-02-09)
2. Total number of Websites, 2022. URL: https://www.internetlivestats.com/total-number-of-websites/ (2022-02-03)
3. W3techs. Usage statistics of content management systems, 2022. URL: https://w3techs.com/technologies/overview/content_management (2022-02-03)
4. Nijs, Diane Elza Lea Winie. Imagineering the butterfly effect. URL: https://pure.rug.nl/ws/portalfiles/portal/6594643/Imagineering_the_Butterfly_Eff_1.pdf (2022-04-03)
5. Martinez-Caro, Jose-Manuel; Aledo-Hernandez, Antonio-Jose; Guillen-Perez, Antonio; Sanchez-Iborra, Ramon; Cano, Maria-Dolores. A Comparative Study of Web Content Management Systems. // Information vol. 9, no. 27 (2018). URL: https://www.mdpi.com/2078-2489/9/2/27/pdf (2022-02-07)
6. CMSWiki. History of Content management systems, 2011. URL: https://web.archive.org/web/20110518053639/http://www.cmswiki.com/tiki-index.php?page=HistoryOfCMS (2022-02-13)
7. Ramalingam, Elanchezhian. Research Paper on Content Management Systems (CMS): Problems in the Traditional Model and Advantages of CMS in Managing Corporate Websites, 2016. URL: https://digitalcommons.harrisburgu.edu/cgi/viewcontent.cgi?article=1007&context=pmgt_dandt (2022-02-12)
8. Meike, Michael; Sametinger, Johannes; Wiesauer, Andreas. Security in Open Source Web Content Management Systems, 2008. URL: https://www.se.jku.at/wp-content/uploads/2009/07/2008.wcms_security.pdf (2022-02-12)
9. IBM (International business machines). Understanding the risks of content management systems: How open source web platforms can open your organization to attack, 2015. URL: http://hosteddocs.ittoolbox.com/undertstandingrisksofCMS.pdf (2022-02-12)
10. Qwaider, Walid Qassim. Information Security and Learning Content Management System (LCMS). // International Journal of Advanced Computer Science and Applications, vol. 8, no. 11 (2017), 588-593. URL: https://thesai.org/Downloads/Volume8No11/Paper_74-Information_Security_and_Learning_Content_Management.pdf (2022-02-10)
11. Crownpeak technology, Inc. The Business Case for a Web Content Management System: A guide for making the case, justifying the cost and choosing the right CMS for your organization, 2016. URL: https://www.crownpeak.com/resources/white-papers/whitepaper_the-business-case-for-a-web-cms.pdf?v=16102605173372 (2022-02-09)
12. TerpSys. Open Source vs. Proprietary CMS: Which is right for my organization, 2010. URL: https://www.terpsys.com/docs/default-source/white-papers/open-source-vs-proprietary-cms.pdf?sfvrsn=7b2f90b4_4 (2022-02-09)
13. Positive technologies. Security trends & vulnerabilities review: Web applications, 2016. URL: https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Web-Application-Vulnerability-2016-eng.pdf (2022-02-14)
14. Shodan Search Engine. URL: https://www.shodan.io/ (2022-02-13)
15. CVE Mitre History. URL: https://www.cve.org/About/History (2022-02-12)
16. Jerkovic, Hrvoje; Sinkovic, Branko. Vulnerability analysis of most popular open source Content Management Systems with focus on WordPress and proposed integration of artificial intelligence cyber security features. // International Journal of Economics and Management Systems, vol. 2 (2017). URL: https://www.iaras.org/iaras/filedownloads/ijems/2017/007-0010(2017).pdf (2022-02-02)
17. Web Vulnerability Assessment Tool for Content Management System. // International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 1.3 (2020). URL: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse69913sl2020.pdf (2022-03-03)

18. Asaduzzaman, Md.; Rawshan, Proteeti Prova; Liya, Nurun Nahar; Islam, Muhmmad Nazrul; Dutta, Nishith Kumar. A Vulnerability Detection Framework for CMS Using Port Scanning Technique, 2020. URL: https://easychair.org/publications/preprint_download/DpgN (2022-03-04)
19. Acunetix. Web Application Vulnerability Report, 2016. URL: https://www.it-sa.de/EDB/EDB3/LOADPRESSINFO/2721 (2022-03-05)
20. Kaluža, Marin; Vukelić, Bernard; Rojko, Tamara. Content management system security. // Zbornik Veleučilišta u Rijeci, Vol. 4 (2016). URL: https://hrcak.srce.hr/file/236346 (2022-03-05)
21. Exploit database Google hacking database. URL: https://www.exploit-db.com/google-hacking-database (2022-02-20)
22. Kumar, Vinoth R.; Kumar, Kishore K. Exploitation of content management system vulnerabilities to launch large scale cyber attacks. // International Journal of civil engineering and technology, vol. 8, no. 10 (2017). URL: https://iaeme.com/MasterAdmin/Journal_uploads/IJCIET/VOLUME_8_ISSUE_10/IJCIET_08_10_141.pdf (2022-03-02)
23. Australian cyber security centre. Securing Content Management Systems, 2015. URL: https://www.cyber.gov.au/sites/default/files/2021-10/PROTECT%20-%20Securing%20Content%20Management%20Systems%20%28October%202021%29.pdf (2022-03-08)
24. Stanić, Ivan. Visions of the Future in Contemporary Cyberpunk, 2021. URL: https://zir.nsk.hr/islandora/object/ffzg:4432/datastream/PDF/view (2022-02-10)
25. Introducing JSON (Javascript object notation). URL: https://www.json.org/json-en.html (2022-02-17)
26. Shafranovich, Y. Common Format and MIME Type for Comma-Separated Values (CSV) Files, 2005. URL: https://datatracker.ietf.org/doc/html/rfc4180#section-2 (2022-03-16)

Prilozi

Github URLovi :

Platforma (Izrađena od strane autora rada) : https://github.com/HawkstoNGriM/NeonEx/

Skripta za detekciju CMS-a: https://github.com/Krisseck/Detect-CMS

Modificirana skripta za detekciju CMS-ova (Od strane autora rada): https://github.com/ffos-user-57/Detect-CMS

Podatkovni skupovi: https://github.com/MrSentex/0day.today-API te https://github.com/offensive-security/exploitdb

Programske skripte i opisi

Uspostava na Linux

Temelj koda za detekciju CMS-a kroz Windows operacijski sustav oslanja se na definiranje direktorija i putanja u skripti koristeći obrnutu kosu crtu (\), dok se za linux u skripti ovo treba izmijeniti. Srećom, u ovom slučaju nije potrebno izmjenjivati kose crte, jer PHP programski jezik percipira direktorije neovisno o operacijskom sustavu - pa će na taj način moći shvatiti da se "\\DetectCMS\\Systems\\Drupal.php" odnosi na direktorij počevši s onim gdje je .php skripta koja pokreće ovu komandu, te će sva struktura odgovarati kodu i potrebno je napraviti samo minimalne izmjene.

Primjer modifikacije skripte da uključuje biblioteke na Linuxu:

```
$system_class = '\\DetectCMS\\Systems\\' . $system_name;
include __DIR__ . "/" . "Systems/" . $system_name . ".php";
```

umjesto Windowsovog:

```
$system_class = 'DetectCMS\\Systems\\' . $system_name;
include __DIR__ . "\\Systems\\" . $system_name . ".php";
```

Uspostava koda na više platformi potrebna je kako bi potencijalni korisnici mogli koristiti alat NeonEx na različitim sustavima i u različitim okružjima te kako bi se omogućilo pokretanje na poslužiteljskim sustavima bilo na Windows ili Linux sustavima. Bar svojevrsna interoperabilnost potrebna je kod izgradnje bilo kakvog programskog rješenja, pogotovo ako se radi o programskom rješenju povezanom uz kibernetičku sigurnost koja je često povezana uz Linux. I na Linuxu i na Windowsu se koristi XAMPP aplikaciju za učitavanje ovakve platforme, no s obzirom da završni proizvod neće imati bazu podataka, bit će dovoljno samo učitati server koji izvršava PHP (te ima odgovarajuću verziju PHPa) kako bi se NeonEx koristio uspješno. Modificiranje koda za druge verzije podrazumijeva minimalne izmjene u kodu, no to ostaje za buduće verzije. Kod je testiran na Windows 10 operacijskom sustavu, kao i Ubuntu (20.04.4 LTS) Linux operacijskom sustavu, no očekuje se da će raditi na bilo kojem operacijskom sustavu (uključujući naravno MacOS i FreeBSD/OpenBSD) dok se pokreće server koji čita odgovarajuću verziju PHPa.

Nadogradnja podatkovnih skupova

Primjer skripte za nadogradnju podatkovnih skupova za Linux distribucije:

```
#!/bin/bash

rm Resources/allCVEs2022.csv

rm Resources/files_exploits.csv

echo "[-] Removed the old files"

wget https://cve.mitre.org/data/downloads/allitems.csv -O allCVEs2022.csv

wget https://raw.githubusercontent.com/offensive-
security/exploitdb/master/files_exploits.csv -O files_exploits.csv

echo "[+] Downloaded the new files"

mv allCVEs2022.csv Resources/

mv files_exploits.csv Resources/

echo "[!] New files replaced to the folder. Done. You may exit"
```

Kao što je vidljivo u priloženom kodu skripta uklanja iz Resources foldera .csv datoteke, te dohvaća nove i stavlja ih gdje su stare datoteke bile, a preimenuje ih kako bi PHP skripte prepoznale ime.

Detekcija dodataka (Plug-Inova)

Dva primjer detekcije Plug-inova:

```
#specific keyword detecion - Medium accuracy
$site = file_get_contents($site);
$systems = ["Wordpress"];
if(strtolower($cms) == strtolower($systems[0])){
      $dataset = "wp_popular_extensions_list.txt";
      if($dataset !== ""){
                  $data = file_get_contents($dataset);
                  $data = explode("\n",$data);
                  foreach($data as $d){
                        if(str_contains($site,$d) && $d !== " " && $d !==
"" ){
                                    echo " Plugin :" . $d . " ; " ; }}}
```

Te detekcija riječi :

```
#word detection - Low accuracy

$possiblePluginNames = ["plugin", "plug-in", "extension", "/theme","add-
on","add-in","addon"];

$arrayOfResults = array();

foreach($possiblePluginNames as $pluginName){

      $pluginName = strtolower($pluginName);

      $site = strval($site);

      $site = strtolower($site);

      $siteTextified = explode("/>",$site);

      foreach($siteTextified as $part){

            if(str_contains($part, $pluginName)){

                  $part = explode("<",$part);

                  foreach($part as $miniPart){

                        if(str_contains($miniPart, $pluginName)){

                              $miniPart = explode("=", $miniPart);

                              foreach($miniPart as $reallyTinyPart){

                                if(str_contains($reallyTinyPart,$pluginName)){

                                  if(str_contains($reallyTinyPart,"<") &&
                                    str_contains($reallyTinyPart,">")){

                                      $reallyTinyPart = str_replace("/>"," ");

                                      $reallyTinyPart= str_replace("<", " ");

                                    }

                                                echo $reallyTinyPart;}

                                    }}}}}}
```

Čitanje CSV datoteka

Primjer čitanja CSV datoteke i ispisa vrijednosti:

```
if (($handle = fopen($datasetfile, "r")) !== FALSE) {
      while (($data = fgetcsv($handle, 0, ",")) !== FALSE) {
          $num = count($data); $row++;
          for ($c=0; $c < $num; $c++) {echo "ID ranjivosti :" . $data[c];}}}
```

Spajanje na API sučelja

U idućem primjeru prikazano je dohvaćanje i učitavanje podataka s API sučelja kroz programski kod:

```
      $query = urlencode(“Wordpress 3”);
      $url = "https://www.exploitalert.com/api/search-exploit?name=$query";
      $request = file_get_contents($url);
      $data = json_decode($request);
      foreach($data as $key => $value){
        echo $value->name . " (" . $value->date . ") " . "<a
      href='https://www.exploit-db.com/exploits/" . $value->id . "'> Exploit
      Source </a>"; }
```

Varijabla query kodirana je prema pravilima URLova, a u njoj se nalazi upit CMS-a za kojeg tražimo Exploite i njegova verzija. Zatim se na URL API sučelja dodaje u “name” parametar (jednostavan GET zahtjev definiran od strane APIja) naš upit te se šalje zahtjev koristeći file_get_contents. Nakon što su dohvaćeni podaci potrebno je objasniti PHPu kako ih iščitati - stoga je ubačena linija json_decode gdje vrijednosti varijable “data” postaju čitljive kao JSON podaci jer ih takvim i sam API definira. Naposljetku potrebno je proći kroz sve rezultate i prikazati datum, ID (jer se putem njega pretražuju detalji na Exploit-DB stranici dodavanjem u URL) i urediti ispis podataka na stranicu.

Kreiranje novih modula za postojeće programsko rješenje

Primjer dodavanja TextPattern CMS-a u sustave na temelju kojih se vrši detekcija s primjerom jedne funkcije za detekciju :

```
namespace DetectCMS\Systems;

class Textpattern extends \DetectCMS\DetectCMS {

      public $methods = array( "readme",
"generator_header","generator_meta",);

      public $home_headers = array();

      public $url = "http://localhost/TextPattern";

      function __construct(“”, $home_headers, $url) {

                  $this->home_html = $home_html;

                  $this->home_headers = $home_headers;

                  $this->url = $url;}

      public function readme() {

        if($data = $this->fetch($this->url."/README.txt")) {

         require_once(dirname(__FILE__)."/../Thirdparty/simple_html_dom.php");

         if($html = str_get_html($data)) {

            if (str_contains($html,"Textpattern")){

               return strpos($title->plaintext, "Textpattern CMS") !== FALSE;

                  }}}

                  return FALSE; }}
```

Lista svih CMS-ova koje pokriva detekcija

U skripti kreatora za detekciju su dodani CMS-ovi:

1. Bohemiasoft
2. Joomla
3. Moodle
4. Typo3
5. Concrete5
6. Laravel
7. Shopsys
8. vBulletin
9. Drupal
10. Liferay
11. Shoptet
12. Webgarden
13. EshopRychle
14. Magento2
15. Sitecore
16. Webnode
17. ExpressionEngine
18. Magento
19. Squarespace
20. Wordpress
21. Fastcentrik
22. Modx
23. Textpattern (skripta izrađena od autora rada)

Napomena: Skripta ne uključuje detekciju verzija, ranjivosti, exploita niti slično (dodaci, teme, ...). Jedino služi za detekciju CMS-ova na osnovne načine.