Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-100241-102986

Richard Kello

Operačný systém ako web CMS (WCM) - Manažment používateľov a používateľských skupín v operačnom systéme cez webové rozhranie

Bakalárska práca

Pedagogický vedúci: Ing. Katarína Jelemenská, PhD.
Vedúci práce: Ing. Gabriel Szabó

Máj 2023

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-100241-102986

Richard Kello

Operačný systém ako web CMS (WCM) - Manažment používateľov a používateľských skupín v operačnom systéme cez webové rozhranie

Bakalárska práca

Študijný program: B-INFO4 informatika

Študijný odbor: Informatika

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky (FIIT)

Pedagogický vedúci: Ing. Katarína Jelemenská, PhD.

Vedúci práce: Ing. Gabriel Szabó

Máj 2023

Obsah

Kapitola 1

Analýza

1.1 Operačný systém

Operačný systém (OS) definujeme ako softvér, ktorý slúži na premostenie medzi používateľom počítača a jeho hardvérom. Je to softvér, ktorý riadi zdieľanie úloh medzi používateľmi a koordináciu hardvérových zdrojov. Operačný systém je súbor nástrojov, pomocných programov a systémových aplikácií, ktoré riadia počítačový hardvér a poskytujú všestranné služby pre klientsky aplikačný softvér. Akonáhle je operačný systém funkčný, spracovanie špecifík a možností zápisu sa stáva jeho hlavnou úlohou. Aby každý softvér fungoval správne, operačný systém bude pracovať v súlade s centrálnou procesorovou jednotkou (CPU), pamäťou (RAM) a úložiskom (pevný disk hdd alebo mechanika s nepohyblivým médiom teda ssd disk) každého počítača. Operačný systém spúšťa používateľské aplikačné programy a ponúka vhodné rozhranie na interakciu s hardvérom strojov. Primárnymi funkciami OS je správa počítačových zdrojov a regulácia toku údajov. Pamäť, procesory, vstupné/výstupné zariadenia a trvalé úložné zariadenia sú len niektoré z týchto zdrojov.

V dnešnej dobe existuje veľa operačných systémov, ktoré môžu byť zamerané pre server, smartfón, mikropočítač, osobné počítače a podobne. Medzi niektoré príklady patria:

1. Microsoft Windows
2. Apple macOS
3. Android OS
4. Linux
5. TempleOS

Naša bakalárka práca sa bude zameriavať na operačný systém Linux, konkrétne na linuxovú distribúciu Ubuntu.

1.2 WCMS

Používateľ môže spravovať digitálne informácie na webovej lokalite pomocou systému správy obsahu webu (WCMS), čo je typ systému správy obsahu (CMS), vývojom a správou materiálu bez predchádzajúcej znalosti webového programovania alebo markup jazykov.[10] V našom prípade to bude premostenie medzi operačným systémom Ubuntu linux a používateľom cez webové rozhranie.

1.2.1 Fungovanie WCMS

Používatelia môžu spravovať, kontrolovať, meniť a rekonštruovať obsah na webovej lokalite pomocou WCMS. Používatelia môžu zostaviť materiál pomocou flexibilného jazyka ako XML alebo .NET a uložiť ho do databázy. Používatelia môžu použiť webový prehliadač na prístup k WCMS a potom použiť rozhranie založené na prehliadači na úpravu obsahu a prispôsobenie rozloženia.[10]

Dve základné časti WCMS:

1. **Aplikácia pre správu obsahu (CMA)** - je používateľské rozhranie, ktoré umožňuje používateľom navrhovať, upravovať, meniť a odstraňovať materiál z webovej lokality bez zásahu oddelenia IT. Medzi používateľov, ktorí môžu používať toto rozhranie, patria napríklad marketéri a tvorcovia obsahu.

2. **Aplikácia pre doručovanie obsahu (CDA)** - ponúka služby typu back-end, ktoré transformujú materiál, ktorý používatelia vytvárajú v CMA, na webovú stránku, ktorú si návštevníci môžu prezerať.

1.2.2 Základné nevýhody WCMS

1. **Požiadavky na úložný priestor** - Bežné webové stránky často obsahujú kombináciu textu, grafiky a fotografií. S rastúcim množstvom grafiky alebo obrázkov však rastie aj množstvo pamäte potrebnej na uloženie každej stránky. Výsledkom je, že ak sa nepoužije kompresia, na uchovanie celej stránky je potrebné veľa úložného priestoru. V skutočnosti z článku[20] nielen obmedzuje počet webových stránok, ktoré môžu byť prepojené so stránkou, ale tiež výrazne znižuje efektivitu triedenia a získavania údajov spojených s touto stránkou.

2. **Nízka flexibilita týkajúca sa inovácií webových stránok** - Webová stránka musí byť vždy „up“ pre firemných používateľov, ktorí chcú mať stálu online prítomnosť. Keď je však potrebné pridať nové stránky alebo zmeniť alebo odstrániť zastarané stránky, webové stránky vytvorené a udržiavané pomocou konvenčných metód zvyčajne vyžadujú uvedenie celej stránky do režimu offline.[20]

3. **Bezpečnostné riziká** - Hackeri majú stále prístup k WCMS, ak ho správca

často neopravuje kvôli bezpečnostným problémom. Správcovia musia sledovať a spravovať rôzne pohyblivé časti WCMS, vrátane MySQL, softvéru webového servera a akýchkoľvek doplnkov alebo doplnkov, aby sa znížili bezpečnostné hrozby.[10]

4. **Potreba špecializovaného tímu údržby webových stránok** - Technický tím správy databáz je často povinný spracovať údaje tak, aby dodržiavali potrebný formát, a pridať tieto údaje do databázy webovej stránky, aby mohol spravovať dátový obsah webovej stránky. Podľa zdroja[20] to výrazne zvyšuje náklady na údržbu webovej stránky a znižuje flexibilitu procesu aktualizácie, ako aj predlžuje čas potrebný na aktualizáciu webovej stránky, čím sa zvyšuje riziko, že údaje sú pri zverejnení na webových stránkach neaktuálne.

1.2.3 Základné Výhody WCMS

1. **Jednoduché na používanie** - WCMS sú zvyčajne jednoduché na používanie. Z toho dôvodu predstavujú veľkú výhodu pre ľudí, ktorí nie sú zručný v programovaní alebo s ním nemajú žiadne skúsenosti.

2. **Nízka cena** - Prevádzkové náklady na WCMS sú zvyčajne nízke v porovnaní s tým, čo ponúka používateľom alebo firmám. V niektorých prípadoch sa môže jednať aj o bezplatné predplatné.

3. **Nenáročné na spravovanie** - Väčšina WCMS je nenáročná na prevádzku z pohľadu administrátorov. Poskytujú celú radu nástrojov a možností, ako si napríklad upraviť pracovné toky či spravovať používateľov.

1.2.4 Bezpečnosť WCMS

Každý informačný systém, ktorý je pripojený na internet, musí byť bezpečný, inak môžu používatelia a operátori utrpieť vážne následky, ako napríklad odcudzenie

informácií o ich kreditnej karte alebo zákazníkovi. Open source WCMS sú pre útočníkov príťažlivým cieľom pre ich široké využitie. Používatelia so zlými úmyslami môžu spustiť útoky proti mnohým, ak nie všetkým, aplikáciám vytvoreným pomocou určitého WCMS, ak sa dozvedia o jeho zraniteľnosti.[14] WCMS sú zvyčajne podľa MDPI[13] cieľmi útokov ako: Manipulácia dát napríklad za pomoci SQL injekcie. Phishing dát ako bankové účty alebo iné používateľské dáta za pomoci aj XSS útoku. Spúšťanie kódu pomocou aj jednoduchých grafických súborov. Spam, kedy bežný webový "crawler"prechádza lokalitu a hľadá validné emailové adresy pre použitie u tohoto typu útoku. Napodobovanie WCM portálu, kedy útočníci použijú upravené formuláre na stránkach poskytovaných daným WCMS a čakajú, kým sa obete autorizujú, aby získanie ich prihlasovacích údajov. Poskytnuté informácie od MDPI[13] sa zhodovali aj s OWASP top 10, čo je list naj využívanejších cyber útokov.[15]

Podľa analýzy údajov na obrázku 1.1[1], 49.8% týchto útokov na webové prehliadače sú útoky XSS, ktoré teda tvoria skoro polovicu napadnutí. Pri XSS útoku používateľ objaví spôsob, ako zadať časť škodlivého kódu na webovú stránku [14]. Inak povedané, útok XSS vloží do renomovanej webovej stránky zákernú sériu pokynov, ktoré sa vykonávajú vo webovom prehliadači návštevníka (bez vedomia návštevníka), čím útočníkovi poskytne prístup k citlivým údajom používateľa vrátane tokenov relácie a uložených súborov cookie, v prehliadači [22].

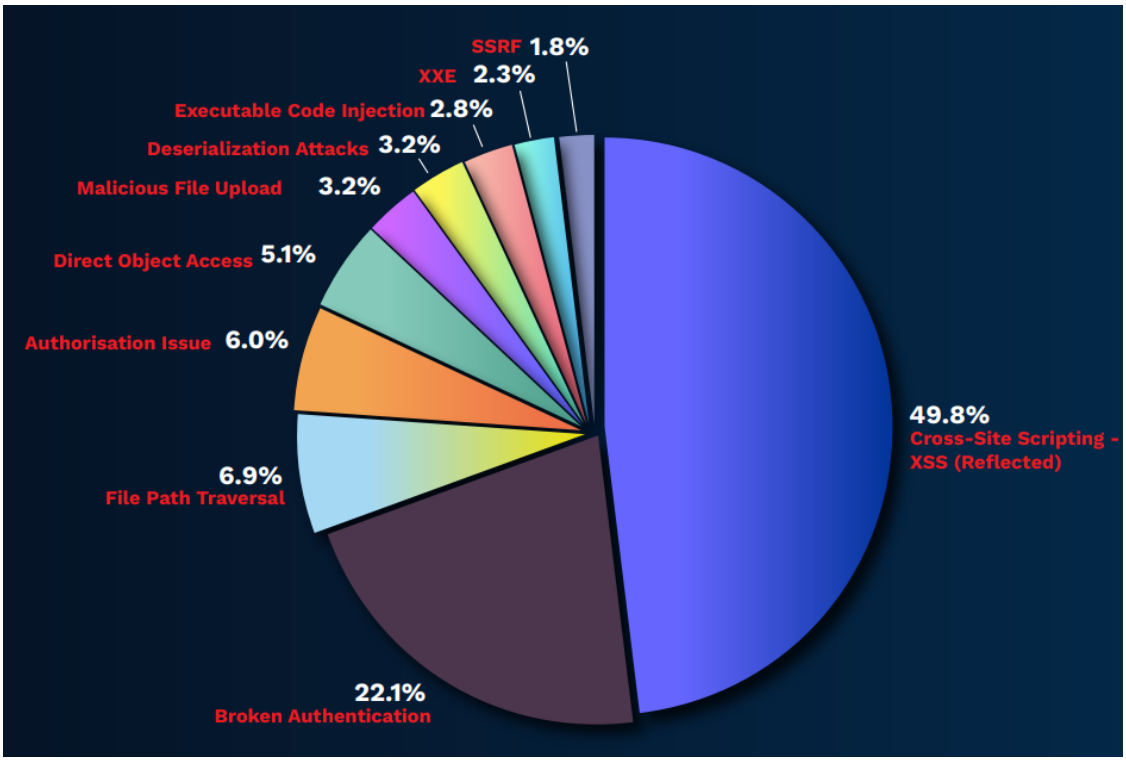
Niektoré varianty útoku XSS:

1. **Reflected XSS útok** - Tento útok používa iné komunikačné prostriedky, aby sa dostal k svojim cieľom, ako sú falošné odkazy v e-mailoch alebo iných webových stránkach, ktoré hlásia útok do webového prehliadača používateľa. Keďže skript pochádza z „dôveryhodného servera“, webový prehliadač ho môže spustiť. Netrvalé alebo XSS útoky typu I sú iným názvom pre tento

druh útoku.[9]

2. **Stored XSS útok** - Keď obeť odošle dotaz, škodlivý skript sa uloží niekde na webový server (napríklad do databázy, správy vo fóre, denníkov, komentárov atď.). Trvalé alebo XSS typu II sú ďalšie názvy pre tento typ útoku.[9]

3. **DOM-Based (Document Object Model) útok** - Na rozdiel od predchádzajúcich typov, kde skript servera spracováva údaje používateľa a vkladá ich späť na webovú stránku, tento druh injekcie vykonáva používateľ.[8]



Obr. 1.1: Distribúcia rôznych techník útoku na prehliadač. Obrázok prevzatý z [1].

Ďalším útokom s vysokým zastúpením bol podľa [1] práve útok porušenia autentifikácie. Jedná sa o útok súvisiacimi s autentifikáciou a potvrdením identity používateľa. Z analýzy údajov na obrázku 1.1[1], celkový počet týchto útokov tvoril až 22.1% celkových útokov.

Z informácií z dostupných článkov[10, 1, 22, 13] môžeme dedukovať že pred typmi útokov na Obr. 1.1, sa vieme chrániť napríklad nasledovne:

1. Vytváranie rutinnej zálohy WCMS (súbory a databázy).
2. Vedenie služieb v skúsenej hostingovej spoločnosti, aby sme sa vyhli útokom ako SQL injekcia.
3. Používanie najnovšej verzie WCMS a doplnkov.
4. Používanie špecializovaných bezpečnostných doplnkov ako JHackGuard.
5. Obmedziť prístup k súborom a priečinkom k administrácii.
6. Odstránenie inštalačných skriptov ako napríklad install.php
7. Vytvorenie bezpečných používateľských rolí a povinná zmena predvoleného hesla.
8. Povolenie captcha pre anonymných používateľov, aby sa zabránilo spamu.
9. E-mailové adresy by mali byť skryté, aby sa zabránilo nežiaducemu spamu.
10. Úprava nastavení globálnych parametrov webových stránok.
11. Pokiaľ je to možné, počas procedúry inštalácie je vhodná zmena predvolenej predpony databázy.
12. Nezobrazovať súkromné údaje WCMS v klientskom rozhraní.

1.2.5 Existujúce riešenia

Medzi existujúce riešenia, ktoré sú dostupné na internete patria napríklad:

1. HubSpot
2. WooCommerce

3. WordPress
4. Joomla
5. Wix
6. Drupal
7. BigCommerce
8. Ghost
9. Magento
10. Textpattern
11. TYPO3

1.2.6 WordPress

Najznámejším WCM systémom v dobe písania tejto práce je WordPress. Zdroj WordPress[6] uvádza, že až 42% web content management systémov používa práve túto platformu. Výhody pri používaní wordpress zahŕňajú: Blockový editor, ktorý zabezpečuje jednoduchosť pri implementácii web portálov. Používatelia, teda nemusia mať žiadne znalosti v oblasti programovania. Medzi výhody rovnako patria aj stovky pluginov a tém[6], ktoré sú platené alebo aj zdarma. Obrovskou výhodou WordPress je fakt, že to je opensource platforma. Používatelia, teda môžu vyhľadať rôzne komunitné skupiny v prípade, že narazia na nejaký problém a nemusia sa spoliehať na support team produktu. Aj napriek množstvu výhod, ktoré WordPress poskytuje, nie je to bez nevýhod. Bezpečnosť na webovom portáli si používatelia musia zabezpečovať sami. Je to spôsobené tým, že WordPress je práve opensource platforma. Z rovnakého dôvodu si vlastník WCMS bude musieť hradiť aj vedenie doménového mena alebo, aj robenia záloh.

User Management - Je jedným z pluginov pre správu používateľov voľne dostupných v portáli WrodPress. Dáva možnosť spravovať používateľov a ich údaje z jedného dashboardu. Import, export a aktualizácia používateľských údajov pomocou rolí a filtrov. Okrem toho ponúka správu používateľov pre WordPress, ktorá umožňuje správcom webových stránok importovať alebo exportovať informácie o používateľoch cez CSV súbor.[19]

1.2.7 HubSpot

HubSpot je ďalším známym web content management systémom. Rovnako, ako WordPress ani pri používaní tohoto web content management systému používatelia nepotrebujú žiadne programátorské znalosti vďaka ich drag-and-drop editoru.[12]. Pre developerov ponúka HubSpot príkazový riadok, ktorý z vlastných skúseností s podobnou featurou značne uľahčuje a urýchľuje prácu. Nakoľko sa jedná o platený produkt, obsahuje aj vbudované bezpečnostné features ako: Content delivery network (CDN), teda doručovanie obsahu, ktorý obsahuje citlivé údaje, chráni heslom. Zašifruje určité súbory na doručovanie obsahu a podobne. No rovnako produkt poskytuje aj web application firewall (WAF), a tak isto, aj dedikovaný bezpečnostný tím, ktorý zabezpečuje stránky pred DDoS útokmi, hakermi a inými bezpečnostnými porušeniami.[12]

Manažment používateľov pre HubSpot je zabudovaný a nie je potreba inštalácie žiadnych ďalších pluginov.[18] Pridávanie je riešené priamo cez nastavenia používateľov a teamov. HubSpot poskytuje rôzne templaty[18] pre nastavenie používateľských práv ako: Super admin, bežný používateľ, vedúci služby a podobne. Tieto práva sú aplikovateľné aj pre používateľské teamy, čo značne uľahčuje prácu nakoľko stačí raz nastaviť práva pre team a ďalej už len pridávať ľudí do daného tímu.

1.3 Linux

Naša implementácia WCMS bude zahŕnať Linux. Jedná sa o open-source operačný systém navrhnutý pánom Linus Torvalds. Pre C, C++, Pascal, Modula-2 a 3, Oberon, Smalltalk a Fortran poskytuje špičkové kompilátory.[5] Existujú rôzne verzie editorov ako vi a Emacs. Virtuálna pamäť, multitasking, viacnásobné prihlásenia, zabezpečenie heslom a ochrana súborov sú plne podporované. Veľké siete teraz umožňujú vzdialené prihlásenie, vzdialené shelly a e-mail vďaka pokrokom v sieťovaní Linuxu.[5] Pre Linux bol vytvorený variant Network File System (NFS). To umožňuje zdieľanie súborového systému medzi niekoľkými počítačmi, takže spotrebúva menej miesta na pevnom disku a vyžaduje menej práce so správou systému.[16] Používatelia Linuxu majú teraz prístup k systému na spracovanie textu TeX/LaTeX, ako aj kresliacim programom (ghostview a xdvi) a nástrojom na náhľad (xfig a idraw).[5] Neplatia sa žiadne licenčné poplatky a všetky tieto funkcie sú bezplatné čo predstavuje obrovskú výhodu pre developmente napríklad WCM systému.

Medzi distribúcie Linuxu patria:

1. Kali Linux - Zameraný na digitálnu forenziu a penetračné testovanie.

2. Ubuntu - Pôvodne vydaný napríklad pre servery a osobné počítače.

3. Fedora Linux - Prispôsobený pre osobné počítače, cloud computing, servery a iné.

4. Arch Linux - Pre používateľov osobných počítačov ktorý chcú voľnosť vo svojo operačnom systéme.

5. Gentoo - Distribúcia zameraná pre power user-ov.

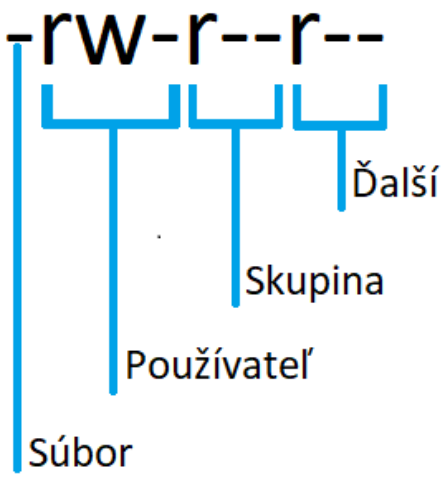
1.3.1 Manažment používateľov v operačnom systéme Linux

Pre začiatok v tejto časti budeme rozoberať povolenia súborov. Existujú 3 základné zaradenia používateľov na základe, ktorých sa v Linuxe rozdeľuje vlastníctvo súboru. A to síce používateľ, skupina a ďalší. Používateľ bude tá osoba, ktorá vytvorila daný súbor, teda vlastník súboru. Pod skupinou rozumieme začlenenie jedného alebo viacerých používateľov, ktorí budú mať rovnaké povolenia k súboru pokiaľ to nenastavíme inak pre jednotlivca. Skupiny zjednodušujú manažment používateľov nakoľko nám stačí nastaviť povolenia pre spúšťanie, čítanie alebo zapisovanie raz pre určitú skupinu a následne už len pridávať používateľov do danej skupiny. V neposlednej rade existuje zaradenie ďalší. Tu zaraďujeme používateľov, ktorí buď nevytvorili daný súbor alebo nie sú v žiadnej skupine, ktorá má jedno z troch spomínaných povolení k súboru. Linuxové povolenia možno rozdeliť na r(read teda čítanie), w(write, teda zapisovanie), x(execute, teda spúšťanie, napríklad skriptov a podobne) a nakoniec pomlčku, ktorá hovorí o tom, že dané zaradenie(používateľ, skupina, ďalší) nemá žiadne povolenie k súboru. Na obrázku 1.2 sme vy zobrazili, ako môžu vyzerať povolenia k súboru. Čiara súbor nám hovorí o druhu súboru. V prípade, že by sa jednalo o adresár, miesto pomlčky by sme tam videli písmeno d(directory). Ďalej na obrázku môžeme vidieť, že používateľ, teda tvorca súboru má povolenia pre čítanie(r) a zápis(w). Skupina a ďalší majú povolenia len na čítanie súboru. Z obrázku môžeme vidieť, že povolenia na spúšťanie nemá žiadne z troch zaradení. Povolenia môžeme v konzole zobraziť napríklad príkazom "ls -l". Povolenia k súboru by sme vedeli editovať príkazom chmod, ktorý by mohol vyzerať nasledovne: chmod u+x <názov súboru>. Príkaz by nám nahradil pomlčku u používateľa za x(execute). Povolenia by, teda vyzerali nasledovne: -rwxr–r–. Pre odstránenie povolenia by sme v pôvodnom príkaze nahradili + za -: chmod u-x <názov súboru>. Výsledné povolenie by, teda opäť vyzeralo ako na obrázku 1.2.

V príkaze chmod vieme ďalej nahrádzať zaradenia a písmená(rwx) aj číslami, ktoré predstavujú nasledujúce povolenia:

1. **Číslo** **Povolenie**
2. 0 Žiadne povolenie
3. 1 Spúšťanie
4. 2 Zapisovanie
5. 3 Spúšťanie a zapisovanie
6. 4 Čítanie
7. 5 Čítanie a spúšťanie
8. 6 Čítanie a zapisovanie
9. 7 Čítanie, zapisovanie a spúšťanie

Príkladom upraveného príkazu chmod by bolo: chmod 777 <názov súboru>. Čísla v poradí predstavujú rovnakú hierarchiu ako na obrázku 1.2. Teda prvé číslo je pre používateľa, druhé pre skupinu a posledné pre ďalší. Predchádzajúci príkaz by nám nastavil povolenia nasledovne pri napísaní príkazu "ls -l": -rwxrwxrwx. Teda všetci zo zaradení by mali povolenie vykonávať každú operáciu. Takýto prístup sa všeobecne v praxi nevyužíva nakoľko to predstavuje bezpečnostnú hrozbu, ako to je opísané v pod sekcii 1.3.2, šiestej odrážke "Povolenia na prístup k súborom".



Obr. 1.2: Povolenia súboru v OS Linux.

Pre manažment používateľov v operačnom systéme Linux najskôr potrebujeme používateľov a skupiny. Ďalej sa, teda budeme zameriavať na ich vytváranie.

Používateľský účet v systéme vieme vytvoriť príkazom: sudo useradd <meno používateľa. Tento príkaz nám však vytvorí len používateľa bez domovského adresára bez možnosti používateľa prihlásiť sa. Aby sa však novo vytvorený používateľ vedel aj prihlásiť do svojho účtu musíme použiť nasledujúci príkaz: sudo useradd -m -s /bin/bash <meno používateľa>. Predchádzajúci príkaz nám okrem používateľa vytvorí aj jeho domovský adresár. Pre zaistenie bezpečnosti musíme nastaviť heslo používateľovi, čo dosiahneme príkazom: sudo passwd <meno používateľa>. Ďalej do konzole len vpíšeme heslo pre používateľa. Mazanie používateľov v operačnom systéme vieme dosiahnuť dvomi spôsobmi. Prvý zahŕňa samostatné vymazanie používateľa a ďalej samostatné vymazania jeho domovského adresára. Druhý a lepší spôsob je vymazanie týchto dvoch častí naraz. Príkaz pre druhý spôsob by vyzeral nasledovne: sudo deluser --remove-home <meno používateľa>.

Ako sme už spomínali, používateľské skupiny v systéme Linux hrajú obrovskú rolu z hľadiska jednoduchosti manažmentu používateľov. Znamená to, že rovnako, ako

používateľom cez príkaz chmod aj skupinám nastavujeme povolenia r,w alebo x a následne prideľujeme používateľov do vytvorených skupín odkiaľ tieto povolenia zdedia. Vytváranie používateľských skupín realizujeme cez príkaz: sudo groupadd <meno skupiny>. Používateľov do skupiny následne pridávame za pomoci príkazu: sudo usermod -aG <meno skupiny> <meno používateľa>. Správnou praktikou je vytvorenie skupiny a nastavenie povolení predtým, ako pridáme používateľa do skupiny, aby sme predišli možným nepovoleným operáciám, ktoré používateľ vie uskutočniť a nemal by podľa hierarchie spoločnosti mať oprávnenia na vykonanie týchto operácií. Mazanie používateľov zo skupiny vieme vykonať cez príkaz: sudo gpasswd -d <meno používateľa> <meno skupiny>. V prípade, že niektorá z existujúcich skupín v systéme nie je potrebná, vieme ju vymazať príkazom: sudo groupddel <meno skupiny>.

1.3.2 Bezpečnosť operačného systému Linux

Táto časť vysvetľuje, ako chrániť systém Linux pred vnútornými aj vonkajšími útokmi. Tieto techniky môžu zahŕňať používanie úložísk na zabezpečenie systému, používanie anti-vírusu na kontrolu, či stiahnutý softvér nie je napadnutý vírusmi, opatrnosť pri spúšťaní softvéru Windows na systémoch Linux, aktualizáciu softvéru, konfiguráciu pravidiel brány firewall, správu hesiel a používanie rôznych prístupových povolení pre rôznych používateľov.

1. **Zabezpečenie prostredníctvom úložísk** - Softvér v linuxových distribúciách sa často sťahuje a inštaluje cez úložiská, ktoré obsahujú veľké množstvo balíkov, ktoré môžu používatelia používať a sťahovať. [11] Kali Linux napríklad obsahuje nástroje navrhnuté špeciálne na penetračné testovanie. [2] Pretože vývojári týchto linuxových distribúcií schválili repozitáre, ktoré sú povolené s predvolenou dodávkou operačného systému, tento spôsob inšta-

lácie softvéru je všeobecne považovaný za vysoko bezpečný. [21] Repozitáre však nedávajú všetko. Vo všeobecnosti by sa používatelia mali snažiť vyhnúť inštalácii softvéru z internetu alebo z iných zdrojov a namiesto toho sa zamerať na používanie repozitárov, ktoré ponúka linuxová distribúcia, ktorú si vybrali.

2. **Použitie antivírusu: ClamAV** - Pokiaľ používateľ nemá inú možnosť, ako stiahnuť softvér, ktorý sa nachádza mimo repozitára jeho distribúcie, mal by použiť antivírusový program. Dostupný priamo v repozitároch distribúcií alebo na oficiálnej stránke, ClamAV je jednou z možností.[21] Vo svojej podstate je podobný známemu antivírusovému programu Windows Defender. Oba obsahujú možnosť plánovania kontroly systému, v prípade, že používateľ obľubuje písanie skriptov vie ClamAV ignorovať dané repozitáre.[7] Obsahuje konzolovú verziu, ale aj verziu s GUI pre používateľov menej zdatných s konzolou. Databáza hrozieb pre ClamAV je neustále aktualizovaná a práve aj z tohoto dôvodu považujeme program za valídnu možnosť pre Linux distribúcie.

3. **Aktualizácia softvéru** - Stiahnuté aplikácie aj systémové balíky je potrebné aktualizovať, aby bol systém Linux bezpečný. Pre aktualizáciu celého softvéru, ktorý bol získaný z úložísk, ako je prehliadač alebo jadro, je potrebné navštíviť buď webovú lokalitu, z ktorej bol softvér stiahnutý, a nainštalovať najnovšie verzie, alebo použiť terminál na vykonanie pre distribúciu špecifického sledu príkazov. Záverom možno povedať, že tragédiám, ako je strata údajov, sa dá predísť udržiavaním aktuálnych systémov Linuxu.

4. **Firewall** - Firewally môžu byť nainštalované na ochranu pred útokmi zo zariadení pripojených k rovnakej sieti. Firewall je súbor pravidiel, ktoré určujú, ktoré porty sú prístupné počítačom zvonku a ktoré správy môže lokálny

počítač prenášať na iné počítače. [3] Napríklad SSH(Secure shell protocol) je sieťový protokol, ktorý umožňuje bezpečné spojenie medzi počítačmi cez sieť. Uskutočňovanie určitých spojení by však nemalo byť povolené, ak sú ostatní používatelia siete neznámi alebo ak ide o verejnú sieť umiestnenú vo verejnej oblasti. Pracovná stanica s príliš veľkým počtom otvorených portov je tiež zraniteľná voči útokom DDoS(Distributed denial of service), čo môže spôsobiť nefunkčnosť systému, kým nebude reštartovaný.[21] Pravidlá môžu byť nastavené na zastavenie takýchto útokov pomocou softvéru iptables, ktorý je prítomný vo väčšine linuxových repozitárov. Napríklad brána firewall určená výhradne pre domácu sieť môže byť menej prísna, aby umožňovala pripojenia ako SSH(Secure shell protocol). Pre verejné siete je možné vytvoriť nový súbor smerníc, ktoré povolia iba HTTP(Hypertext Transfer Protocol) a HTTPS(Hypertext Transfer Protocol Secure), aby boli aktívne, ale zakázali ostatným používateľom v tej istej sieti používať a útočiť na porty SSH(Secure shell protocol) a FTP(File transfer protocol). Keď sú tieto dve sady pravidiel vytvorené, možno ich zameniť pomocou príkazu „iptables-restore < rules“, kde pravidlá sú názov súboru pravidiel brány firewall, ktorý sa nachádza v aktuálnom pracovnom adresári.[21]

5. **Správa hesiel** - Používatelia sú často požiadaní, aby si zaregistrovali používateľské konto pri inštalácii distribúcie Linuxu na počítač, ktorý generuje adresár pre všetky údaje tohto používateľa v rámci súborového systému. Aby sa predišlo neúmyselnému alebo úmyselnému zničeniu dôležitých systémových súborov používateľa, tieto súbory musia byť uložené oddelene od súborov používateľa root. Okrem toho musia byť pre každého používateľa v systéme vytvorené samostatné účty, ak existuje niekoľko používateľov. Jednotlivé používateľské súbory môžu byť oddelené od seba vďaka tejto izolácii, ktorá tiež zabraňuje neoprávneným používateľom v prístupe k údajom. Po-

užívateľské heslá a heslá root sa musia navzájom líšiť. Dobrou praktikou je používať silné heslá, ktoré ideálne používateľ nepoužíva nikde inde.

6. **Povolenia na prístup k súborom** - V systéme musia byť nakonfigurované rôzne povolenia, aby sa ostatným používateľom zabránilo v prístupe k súborom, ku ktorým by nemali mať prístup. Na dokončenie je možné použiť príkazy ako chmod. Príkaz chgrp je možné použiť aj na vytváranie skupín používateľov, aby bolo možné nastaviť povolenia pre viacerých používateľov naraz. Používatelia nebudú môcť manipulovať alebo spúšťať súbory alebo potenciálne meniť systém spôsobom, akým by nemali, nastavením povolení pre osoby, ktoré môžu čítať, zapisovať a spúšťať súbory.[21]

1.4 Laravel

V nasledujúcej sekcii sa pozrieme na framework Laravel a prečo sme si vybrali práve tento framework oproti iným dostupným riešeniam ako WordPress. Obrovskou výhodou frameworku Laravel je jeho rýchlosť. Je zameraný tak aby minimalizoval počet potrebných krokov od začiatku práce po jeho publikáciu. Dosiahnuté to je najmä plytkou krivkou učenia. Všetko od interakcií databáz cez autentifikáciu až po caching je zahrnuté v moduloch Laravel[17]. Ďalej má nami zvolený framework výhodu v tom, že poskytuje vlastný ekosystém ktorý značne pomáha pri implementácií riešení. Ide o súbor funkcionalít ako napríklad manažment serverov, pokročilé nasadenie či miestny vývoj. Laravel sa snaží svojou funkcionalitou a možnosťami čo najviac uľahčiť prácu programátorov tým, že ich dáva na prvé miesto.

Oproti WordPress je Laravel rýchlejší z hľadiska spracovávania požiadaviek. Nakoľko obsahuje WordPress veľké množstvo funkcií, značne to spomaľuje jeho schopnosť odpovedať na požiadavky. Laravel má ďalšiu výhodu oproti WordPress, a to síce náročnosť vytvárania vlastných pluginov. WordPress nepatrí medzi najzloži-

tejšie nástroje pre spomínanú činnosť no Laravel to rieši podstatne jednoduchšie.

1.5 Flask

V sekcii Flask sa pozrieme na teto framework, porovnanie s framework-om Django a prečo sme si vybrali Flask. Python balík Flask slúži ako webový framework, ktorý umožňuje vytváranie webových aplikácií. Jeho jadro je kompaktné a jednoducho sa rozširuje. Je to mikroframework bez správcu vzťahov s objektmi alebo podobných schopností. Ponúka veľa funkcií, ako napríklad nástroj šablón a smerovanie adries URL. Je to framework webovej aplikácie pre WSGI(Web server gateway interface). Flask vytvoril pán Armin Ronacher. Súbor nástrojov Werkzeg(súprava nástrojov WSGI) a šablónový engine Jinja2(nástroj šablón pre Python) slúžia ako základ pre framework Flask.

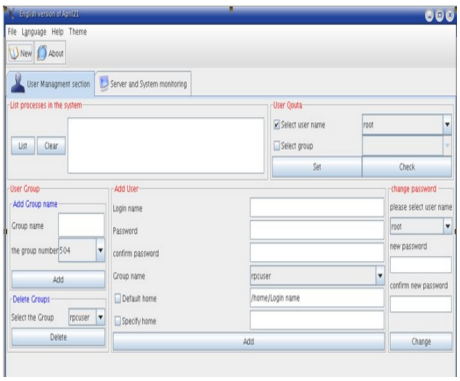
Webový Python framework Django umožňuje rýchle vytváranie bezpečných a spoľahlivých webových stránok. Django umožňuje sústrediť sa na vývoj aplikácie bez toho, aby používateľ musel znova vymýšľať koleso. Django ponúka takmer všetko, čo by vývojári potrebovali hneď po inštalácii. Všetko, čo je potrebné, je obsiahnuté v jedinom produkte, takže všetko funguje jednotne a dodržiava rovnaké dizajnové štandardy. Oproti Djangu je Django Full stack web framework.

Rozdiely medzi framework-ami Django a Flask sú nasledujúce: Django je oproti Flasku omnoho robustnejšie a poskytuje viac funkcionalít hneď po inštalácii. Predstavuje to však aj nevýhody ako zložitosť navrhovania či rýchlosť. Vo všeobecnosti je Flask brány, ako jednoduchší framework a je preferovaný hlavne pri malých projektoch. Avšak oproti Flasku má Django podporu dynamických HTML stránok. Znamená to, že v Djangu vie používateľ vytvoriť viac animovanú a interaktívnu stránku. Django je tiež ťažšie na učenie oproti Flasku. Najdôležitejším rozdie-

lom pre nás však bola podpora API nakoľko to bude nevyhnutná súčasť našej práce rozhodli sme sa pre framework Flask nakoľko oproti Djangu má podporu pre API.

1.6 Porovnanie dostupných WCMS funkcií na manažment používateľov s Linuxom

Podobným riešením našej bakalárskej práce bol projekt z roku 2006 s názvom JLAT(Java Linux administration tool).[4] Jednalo sa o projekt, ktorý mal za úlohu administráciu operačného systému Linux cez rozhranie napísané v jazyku Java. Na obrázku 1.3[4] je ukázané, ako nástroj vyzeral. JLAT bol projekt, ktorý nefungoval na WCMS, ako to bude v prípade našej bakalárskej práce, ale miesto toho to bolo aplikácia, ktorú si používateľ mohol nainštalovať do operačného systému Linux. Jednou nevýhodou JLAT bolo požívanie. Nástroj mohol používať len super user[4](používateľ počítačového systému so špeciálnymi oprávneniami potrebnými na správu a údržbu systému; správca systému) nakoľko mnoho príkazov, ktoré boli implementované do GUI potrebovali sudo oprávnenia. V našom riešení sa chceme vyhnúť podobným problémom nakoľko naša implementácia WMCS by mala fungovať aj pre koncových používateľov.



Obr. 1.3: Java Linux administration tool GUI. Obrázok prevzatý z[4].

1.7 Stručný opis riešenia

Na implementáciu našej bakalárskej práce sme sa rozhodli pre linuxovú distribúciu Ubuntu vo verzii 22.04, ktorá vychádza z distribúcie Debian. Distribúciu Ubuntu sme volili aj z dôvodu jednoduchosti inštalácie a používania oproti iným distribúciám, ako napríklad Arch Linux. Ďalším dôvodom prečo sme volili spomínanú distribúciu bola stabilita tohoto operačného systému. Dôležitým faktorom pri výbere distribúcie bola aj aktuálnosť a používanosť. Nakoľko je Ubuntu neustále aktualizované a zároveň najpopulárnejšie medzi linuxovými distribúciami, zaručí nám to aktuálne pomôcky pri riešení našej bakalárskej práce. Ďalej budeme používať Laravel na posielanie požiadaviek z frontendu na backend Flasku.

[10] [20] [14] [13] [15] [22] [1] [9] [8] [6] [19] [12] [18] [5] [16] [11] [2] [21] [7] [3] [17] [4]

Kapitola 2

Opis riešenia

2.1 Opis požiadaviek

2.1.1 Funkčné požiadavky

1. Prvú funkčnú požiadavku sme si určili ako prevenciu neautorizovaného prístupu do infraštruktúry, aplikácie alebo k dátam.
2. Ako ďalšiu požiadavku v tejto kategórii sme zaradili ukladanie údajov používateľov a poverení.
3. V našom projekte budeme ďalej zabezpečovať poskytovanie prihlasovacieho mechanizmu pre koncových používateľov.
4. Systém je bez používateľov len prázdnou entitou. A preto sme si ako funkčnú požiadavku určili aj registráciu, nastavovanie a resetovanie hesiel.
5. Keďže budeme administrátormi nami vytvoreného systému, ako požiadavku sme si určili aj prideľovanie používateľských prav k systémom a službám
6. V neposlednej rade požiadavka ktorá vychádza z predchádzajúcej bude správa

užívateľských oprávnení v rámci služieb a systémov.

2.1.2 Nie-funkčné požiadavky

Do nie-funkčných požiadaviek radíme rýchlosť, správnosť a efektívnosť. Môžeme ich aplikovať napríklad na autorizácie keďže chceme, aby používateľ nečakal dlho pri prihlasovaní a rovnako, aby sa správne kontrolovali údaje poskytnuté používateľom s databázou používateľov. Jednoduchosť používania sme rovnako zaradili do tohoto druhu požiadaviek. Chceme, aby administrácia používateľov v našom WCMS bola priamočiara a nespôsobovala ťažkosti správcom. Keďže aplikácie sú často neprehľadné naším cieľom bude vytvoriť administráciu, v ktorej sa používatelia/administrátori nebudú strácať a zároveň bude aj ľahko naučiteľná a jednoducho zapamätateľná pre opakované používanie.

2.2 Návrh

Naše riešenie budeme realizovať cez spomínanú distribúciu Linuxu, Ubuntu. Náš backend bude tvoriť webový framework flask, do ktorého expose-neme framework Laravel. Spomínaný Laravel nám, teda bude poskytovať požiadavky z webu, ktoré bude následne posielať na náš backend, ktorý bude požiadavky spracovávať a následne na ne odpovedať. Výslednú odpoveď pošle Flask opäť na Laravel. Registráciu plánujeme riešiť ako žiadosť na backend, čo bude tvoriť serverovú stranu, ktorá overí či sa daný používateľ už nenachádza v naše databáze a v prípade, že nie, vytvorí nového s poskytnutými informáciami. V prípade, že používateľ existuje (pod poskytnutím menom) backend pošle odpoveď, ktorá bude obsahovať správu o chybe. Rovnaká odpoveď bude zaslaná na frontend aj v prípade, že používateľ zadal heslo v zlom tvare, teda použil nepovolené znaky, bolo príliš krátke/dlhé a podobne. Po vytvorení používateľa mu backend zároveň aj vytvorí a pridelí adre-

sár, ktorý bude obsahovať používateľovu prácu. Do priečinka nebude mať prístup nikto iný okrem konkrétneho vlastníka priečinka a administrátorov. V prípade, že už používateľ účet má pokračuje priamo k prihláseniu. Autentifikáciu budeme riešiť voči backendu(serveru), kde existuje inštancia PAM(Pluggable authentication module). Po prihlásení bude používateľ vidieť GUI, ktoré bude obsahovať nejaké jeho priečinky a prípadne adresáre, kde má prístup. Všetko, čo bude vidieť bude samozrejme aj môcť zdielať medzi inými používateľmi alebo skupinami cez systém. Pre zdieľanie bude musieť používateľ poznať meno skupiny alebo druhej osoby, s ktorou chce obsah zdielať. Po vybraní možností zdieľania používateľ odošle súbor, čo vytvorí požiadavku voči serveru, kde to spravuje ACL(Anterior cruciate ligament) a odošle sa odpoveď odosielateľovi či všetko prebehlo v poriadku, teda sa prípadne odošle chybová hláška ak niečo nebolo dobre. Môže sa jednať napríklad o príliš veľký súbor a podobne. Používateľské a skupinové oprávnenia budeme držať v databáze a s entitami budú spojené cez ID.

Literatúra

[1] *2022 vulnerability statistics report*. 2022. URL: https://www.edgescan.com/2022-vulnerability-statistics-report-lp/#form.

[2] Lee Allen, Tedi Heriyanto a Shakeel Ali. *Kali Linux-Assuring security by penetration testing*. Packt Publishing Ltd, 2014.

[3] David Barrera, Ian Molloy a Heqing Huang. "IDIoT: Securing the Internet of Things like it's 1994". In: (dec. 2017). arXiv: 1712.03623 [cs.CR].

[4] Ahmed Bentiba, Ahmed Mohmed a Jamal Zemerly. "Java Linux Administration Tool". In: *2006 IEEE GCC Conference (GCC)*. IEEE. 2006, s. 1–4.

[5] SN Bokhari. "The Linux operating system". In: *Computer* 28.8 (1995), s. 74–79.

[6] *Build a site, Sell your stuff, start a blog amp; more*. URL: https://wordpress.com/?aff=190.

[7] *ClamAV documentation*. URL: https://docs.clamav.net/.

[8] *Common weakness enumeration*. URL: https://cwe.mitre.org/data/definitions/79.html.

[9] *Cross site scripting (XSS)*. URL: https://owasp.org/www-community/attacks/xss/.

[10] Demetra Edwards et al. *What is a web content management system (WCMS)?* 2021. URL: https://www.techtarget.com/searchcontentmanagement/definition/web-content-management-WCM.

[11] J A Galindo, D Benavides a S Segura. "Debian Packages Repositories as Software Product Line Models. Towards Automated Analysis". In: *the 1st International Workshop on Automated Configuration and Tailoring of Applications*. 2010.

[12] HubSpot. *HubSpot website builder and Marketing Free.* URL: https://www.hubspot.com/marketing/am_website-builder-hsmf?irclickid=TWW1z\%3A3vxxyNRuXWgJQMRRfEUkAzBZUpDy-IVo0&irgwc=1\&mpid=11535\&utm_id=am11535\&utm_medium=am\&utm_source=am11535\&utm_campaign=amcid_TWW1z\%3A3vxxyNRuXWgJQMRRfEUkAzBZUpDy-IVo0_irpid_11535\&utm_content=wordpress.

[13] Jose-Manuel Martinez-Caro et al. "A comparative study of web content management systems". In: *Information* 9.2 (2018), s. 27.

[14] Michael Meike, Johannes Sametinger a Andreas Wiesauer. "Security in Open Source Web Content Management Systems". In: *IEEE Security Privacy* 7.4 (2009), s. 44–51. DOI: 10.1109/MSP.2009.104.

[15] *Owasp Top Ten.* URL: https://owasp.org/www-project-top-ten/.

[16] Spencer Shepler et al. *Network file system (NFS) version 4 protocol.* Tech. spr. 2003.

[17] Matt Stauffer. *Laravel: Up & running: A framework for building modern php apps.* O'Reilly Media, 2019, s. 5–9.

[18] Support. *Add HubSpot users.* 2018. URL: https://knowledge.hubspot.com/settings/add-and-remove-users.

[19] WPExperts a Uzair Ahmed. *User management.* 2022. URL: https://wordpress.org/plugins/user-management/.

[20] Ming-Ju Yang et al. “A User-Friendly Web Content Management System”. In: *2008 3rd International Conference on Innovative Computing Information and Control*. IEEE. 2008, s. 367–367.

[21] Matthew R. Yaswinski, Md Minhaz Chowdhury a Mike Jochen. “Linux Security: A Survey”. In: *2019 IEEE International Conference on Electro Information Technology (EIT)*. 2019, s. 357–362. DOI: 10.1109/EIT.2019.8834112.

[22] Imran Yusof a Al-Sakib Khan Pathan. “Mitigating Cross-Site Scripting Attacks with a Content Security Policy”. In: *Computer* 49.3 (2016), s. 56–63. DOI: 10.1109/MC.2016.76.