Správa uživatelů a přístupů v Linuxovém operačním systému

Ľuboš Lukáčik

Bakalářská práce
2018

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ľuboš Lukáčik**
Osobní číslo:	**A15272**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Forma studia:	**kombinovaná**
Téma práce:	**Správa uživatelů a přístupů v Linuxovém operačním systému**
Téma anglicky:	**User and Access Management in the Linux Operating System**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na téma Správa uživatelů a přístupů v Linuxovém operačním systému.**
2. **Detailně popište**
   - **Přihlašování uživatelů (použití různých metod zabezpečení, lokální přístup a přístup přes SSH).**
   - **Audit aktivity uživatelů systému.**
   - **Správu oprávnění uživatelů a skupin.**
   - **Konfigurace sudo.**
3. **Proveďte implementaci a konfiguraci v Linuxovém systému tak, aby byly splněny požadavky na bezpečnost.**
4. **Sestavte dokumentaci provedené konfigurace a možností nastavení.**

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Linux: dokumentační projekt. 4., aktualiz. vyd. Přeložil Lubomír PTÁČEK. Brno: Computer Press, 2007. ISBN 978-80-251-1525-1.**
2. **SSH Communications Security. SSH Communications Security. SSH (SECURE SHELL). [Online] 2017. https://www.ssh.com/ssh/.**
3. **DOSTÁLEK, Libor a KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. Brno : Computer Press, a.s., 2008. 978-80-251-2236-5.**
4. **KRČMÁR, Petr. Linux: Tipy a triky pro bezpečnost. Praha: Grada Publishing, 2004. ISBN 80-247-0812-4.**
5. **Sudo: Sudo Manual Pages [online]. 2017 [cit. 2017-11-18]. Dostupné z: https://www.sudo.ws/man.html.**

Vedoucí bakalářské práce:	**doc. Ing. Martin Sysel, Ph.D.** Ústav počítačových a komunikačních systémů
Datum zadání bakalářské práce:	**12. prosince 2017**
Termín odevzdání bakalářské práce:	**24. května 2018**

Ve Zlíně dne 12. prosince 2017

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

Ing. Jan Valouch, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne…….

podpis diplomanta

ABSTRAKT

Bakalárska práca je zameraná na správu užívateľov linuxových systémov a spôsoby akým sa prihlasujú na systém. Teoretická časť práce opisuje históriu Unixových operačných systémov, typy prihlásenia, správu oprávnení a zaznamenávanie aktivít užívateľov. Praktická časť práce sa venuje zhrnutiu parametrov, ktoré je možné nastaviť pre čo najefektívnejšie zabezpečenie systému v rámci správy užívateľov. Súčasťou praktickej časti je aj nastavenie logovania a auditu, ktoré slúži na jednoduchšie dohľadanie príčin problémov na systéme.

Kľúčové slová: Linux, správa užívateľov, PAM, SSH, Sudo

ABSTRACT

The bachelor thesis is aimed at managing the users of Linux systems and how they log on to the system. The theoretical section of the thesis describes the history of Unix operating systems, types of logging, authorization management and recording user activities. The practical part deals with all the parameters that can be set for the most accurate system security within the user administration. Part of the practical section is logging and auditing setup, which makes it easier to track the causes of problems on the system.

Keywords: Linux, user management, PAM, SSH, Sudo

Chcel by som poďakovať vedúcemu mojej diplomovej práce pánovi doc. Ing. Martinovi Syslovi, Ph.D. za jeho rady a pripomienky, ktoré mi poskytol pri vypracovaní tejto práce.

OBSAH

ÚVOD

Exponenciálny nárast digitalizácie sveta je spätý s nárastom využitia serverov. Či už hovoríme o rozsiahlych datacentrách, alebo sa zameriame na domáce servery tvorené jednoduchými počítačmi typu Raspberry Pi, hrozí tu riziko útoku tretej strany. Korektné vyhodnotenie rozsahu týchto rizík nám poskytuje možnosť dostatočnej a včasnej prevencie pred útokmi.

Súčasťou každého operačného systému sú užívatelia, ktorým administrátor musí nastaviť bezpečnostnú pravidlá takým spôsob, aby zodpovedali aktuálnym štandardom. Pravidlá nastavené administrátorom jednoznačne určujú právomoci užívateľa. Neodlúčiteľnou súčasťou správy užívateľov je spôsob ich prihlasovania, pričom najmä sa využíva protokol SSH, s ktorým sa bližšie zoznámime v teoretickej časti práce.

Cieľom práce je zhrnúť bezpečnostné nastavenia jednotlivých programov, protokolov a démonov v spojení s užívateľmi. Rozsiahlejšie sa oboznámime i so zoznamom parametrov, ktorých implementáciou dokáže administrátor zdokonaliť bezpečnosť na systéme.

Napriek implementácii všetkých nastavení je vyžadované sledovať aktivity na systéme a logovať ich. K tomuto účelu je nasledujúcim cieľom práce zdokumentovať nastavenie logovania a auditu, ktoré sledujú a zaznamenávajú zadefinované aktivity na systéme. Nastavením auditu sa docieli vyššia úroveň prevencie a kontroly nad systémom a tým aj väčšia bezpečnosť.

I. TEORETICKÁ ČÁST

1 OPERAČNÉ SYSTÉMY

Primárna funkcia operačného systému (ďalej OS) spočíva v poskytovaní podpory pre realizáciu počítačových programov. Z uvedeného dôvodu je napríklad možné používať editor a vytvárať dokumenty. Editor potrebuje ku svojej činnosti interakciu s terminálom, súbormi a ďalším technickým vybavením počítača. Tieto služby zabezpečuje OS. Pôvodne bol OS Unix navrhnutý ako zjednodušenie OS Multics. Filozofia OS Unix spočíva v tom, že všetky funkcie by sa mali rozdeliť do čo najjednoduchších programov. Nové funkčné vlastnosti sa dajú získať vhodnou kombináciou jednoduchých programov. Neustále sa objavujú nové obslužné programy, ktoré sa dajú jednoducho integrovať do existujúcich nástrojov a tým sa môže OS neustále rozširovať. Kľúčovou časťou OS je tzv. jadro, ktoré v unixových systémoch plní funkcie ako: spúšťanie programov, prideľovanie systémových zdrojov, prideľovanie času procesora súčasne bežiacim procesom a pod [1].

1.1 Jadro

Jadro je dôležitým centrom každého unixového systému. Je to relatívne malý program, ktorý sa nahráva z celého systému do pamäti ako prvý. Zaisťuje nízkoúrovňovú prácu s hardwarom a dohliada na bezpečný chod celého systému. Jadro má za všetkých okolností prvé aj posledné slovo. Preto akákoľvek bezpečnostná chyba, alebo riziko dokáže ohroziť celý systém. Linux patrí k najsledovanejším systémom vďaka otvorenosti jeho kódu je malá pravdepodobnosť výskytu kritickej chyby. Aj tu sa nájdu výnimky a taká chyba sa nájde, preto je dôležité sledovať vývoj a pri podobných problémoch okamžite aktualizovať OS.

Dnes existujú v podstate dva druhy jadier. Tzv. „vanilla“ je jadro získané priamo zo serveru kernel.org (alebo jeho mirrorov). Jedná sa priamo o kód, ktorý vzniká počas hlavného vývoja a je teda „čistý“ a najbezpečnejší. Mnoho vývojárov distribúcií (Red hat, SuSE) však používa vlastné úpravy a do jadra pridávajú vlastnú časť kódu, ktorú vanilla verzia jadra neobsahuje.

Moduly sú časti jadra, ktoré nie sú skompilované priamo do neho, ale nachádzajú sa v samostatných súboroch v adresári `/lib/modules`. Ich výhodou je možnosť mať preložené všetky časti jadra, ktoré by sme mohli potrebovať ihneď k dispozícii, bez toho aby zaberali miesto v pamäti. Hneď po ich uvedení sa objavili prvé snahy o napadnutie systému

pomocou falošných modulov pre jadro. Keďže sa každý modul po zavedení začne správať ako súčasť jadra (beží v móde jadra), má neobmedzené možnosti.

Existujú dve riešenia problému:

- Prvým riešením je vytvorenie monolitického jadra, ktoré neumožňuje zavádzanie modulov. Tím sa však pripravíme o spomenuté výhody modulov.
- Druhým riešením je využitie systému LIDS, ktorý umožňuje (okrem iného) uzamknutie modulov tak, že zabráni akejkoľvek manipulácii s nimi. Na jeho použitie stačí behom štartu systému, ihneď po zavedení všetkých potrebných modulov spustiť príkaz:

  ```
  lidsadm -I
  ```

Od tejto chvíle sú moduly „zamrazené“ a akákoľvek manipulácia s nimi je zakázaná [2].

1.2 História operačného systému Unix

V roku 1965 pracovali spoločnosti Bell Telephone Laboratories (divízia AT&T) a General Electric na projekte „MAC of MIT“, ktorého cieľom bolo vytvorenie OS Multics. Od tejto spolupráce neskôr Bell Telephone Laboratories odstúpila. Ken Thompson a Dennis Ritchie začali vyvíjať OS, ktorý by bol vhodnejší pre Bell Telephone Laboratories. V roku 1973 bol Unix kompletne prepísaný do jazyka C, ktorý vyvinul Dennis Ritchie. Koncom 70-tych rokov protimonopolný úrad zakázal AT&T činnosť v oblasti počítačových technológii. Následkom čoho AT&T previedlo licencie Unixu na niektoré univerzity a tým sa stal tento OS populárny v akademických okruhoch a postupom času sa začal presadzovať aj v komerčnej sfére [1].

1.3 História operačného systému Linux

Linus Torvalds vytvoril OS Linux počas štúdia na univerzite v Helsinkách, keď si v roku 1991 zaobstaral osobný počítač od IBM s OS MS-DOS. Linus nebol s týmto operačným systémom spokojný a tak chcel začať používať Unix, na ktorý bol zvyknutý z univerzity. Zistil, že najlacnejšia verzia Unix OS stála $5,000 USD. Táto skutočnosť ho viedla k vytvoreniu OS podobnému Unix-u. Linus a približne 100 ďalších vývojárov pracovalo na vývoji Linuxu a v roku 1994 vydali 1.0 verziu Linuxového jadra. Linux má voľne šíriteľnú licenciu, každý môže tento OS používať, kopírovať, študovať alebo upravovať pokým zostane zdrojový kód otvorený a voľne šíriteľný. Je nutné poznamenať že Linux nie je derivátom Unix-u, ale majú mnoho spoločných príkazov.

1.3.1 Distribúcie Linuxu

Pod distribúciou Linuxu je možné si predstaviť jadro Linuxu a kolekciu programov, ktoré spolu tvoria OS. Každá distribúcia má svoje ciele a oblasti, na ktoré sa zameriava.

Distribúcie je možné rozdeliť na:

- komerčné – sú za nimi korporácie a je možné si od nich zakúpiť podporu,
- nekomerčné – sú udržiavané komunitou dobrovoľníkov.

V súčasnosti existujú stovky distribúcii, medzi najznámejšie patria:

- RedHat Enterprise Linux (RHEL),
- Debian,
- Ubuntu,
- SuSE Linux Enterprise Server (SLES),
- Linux Mint [3].

2 AUTENTIZÁCIA UŽÍVATEĽOV

2.1 Heslá

Heslá sú v súčasnej dobe najrozšírenejším spôsobom overovania identity. Ak chce mať administrátor vo svojich systémoch bezpečné heslá, je veľmi dôležitá osveta užívateľov. Pokiaľ je narušiteľ mimo systém, nemôže nič spraviť, ale ak sa dostane dovnútra a má možnosť zadávať príkazy (aj s minimálnymi právami), môže začať v systéme hľadať ďalšie medzery v bezpečnosti. Preto nestačí ak pravidlá bezpečnosti pozná iba systémový administrátor.

V minulosti boli heslá v unixových systémoch uložené v súbore `/etc/passwd`, ktorý je bežne dostupný všetkým užívateľom. Boli v ňom uložené heslá s ďalšími informáciami pre prácu s užívateľskými účtami. Neskôr, ako ochrana proti novým útokom boli heslá presunuté do súboru `/etc/shadow`. K súbor `/etc/shadow` má prístup iba užívateľ root aby mohol overovať heslá. Súbor `/etc/passwd` sa v systéme zachoval a stále uchováva informácie o užívateľských účtoch, ale už bez hesla.

Heslá v unixových systémoch sú jednosmerne šifrované. Vyberá sa postup, pri ktorom nie je možné z utajenej podoby hesla získať originál. Na tento účel sa používajú hash algoritmy [2]. Typ použitého hash algoritmu je možné odvodiť z druhej hodnoty záznamu v súbore `/etc/shadow` [22].

Tabuľka 1:Typ hash algoritmu použitého na zabezpečenie hesla[22]

Hodnota v súbore /etc/shadow	**Použitý hash algoritmus**
1	MD5
$2a$	Blowfish
$2y$	Blowfish (správne zaobchádzanie s 8bit znakmi)
5	SHA-256
6	SHA-512

2.1.1 **Hash algoritmy**

Hash algoritmy sú jednosmerné funkcie, ktoré prevádzajú vstup o ľubovoľnej dĺžke na výstup, ktorý má pevne stanovenú dĺžku (odtlačok). Vlastnosťou týchto funkcii je, že akákoľvek zmena na vstupe (aj o jeden bit) zmení výsledok od originálneho. Táto vlastnosť je demonštrovaná na nasledovnom obrázku.

```
[czz31972@oc8640468720 ~]$ echo hello |shasum
f572d396fae9206628714fb2ce00f72e94f2258f  -
[czz31972@oc8640468720 ~]$ echo Hello |shasum
1d229271928d3f9e2bb0375bd6ce5db6c6d348d9  -
[czz31972@oc8640468720 ~]$ echo hell |shasum
512ba0e938d862261a9914c7f5370dab3d7c1695  -
[czz31972@oc8640468720 ~]$ echo heLLo |shasum
568dd151565bc8e1f6bd851bf515a5c02aa09144  -
```

Obrázok 1: Porovnanie výsledkov SHA256 hash funkcie [zdroj: Autor]

Tieto vlastnosti sú výborné na ochranu hesiel. Heslá môžeme uložiť vo forme, ktorá ich ochráni aj v prípade, ak je súbor s heslami kompromitovaný. Zároveň nám umožňujú overovať či užívateľ zadal správne heslo, bez toho aby bolo heslo niekde uložené v textovej forme.

Štandardné poradie krokov pri registrácii účtu a overenie totožnosti v systéme založenom na hash funkciách:

1. Užívateľ si vytvorí účet.
2. K jeho heslu sa pridá náhodne vygenerovaný reťazec znakov nazývaný soľ (salt), potom je heslo zašifrované pomocou hash funkcie a uložené v súbore `/etc/shadow`.
3. Keď sa užívateľ pokúsi prihlásiť, zašifrované heslo ktoré zadal sa porovná so zašifrovaným, ktoré je uložené v súbore `/etc/shadow`.
4. Ak sa zašifrované heslá zhodujú, užívateľovi je povolený prístup. V opačnom prípade je vyzvaný aby zadal meno a heslo znovu.
5. Krok 3 a 4 sa opakujú vždy keď sa niekto snaží prihlásiť na účet.

Ak je zadané zlé prihlasovacie meno alebo heslo vo štvrtom kroku, je odporúčané zobraziť neurčité oznámenie ako „Nesprávny login alebo heslo“. Toto opatrenie zamedzí útočníkovi zistiť správne prihlasovacie meno bez znalosti hesla [4].

2.1.2 Najčastejšie typy útokov

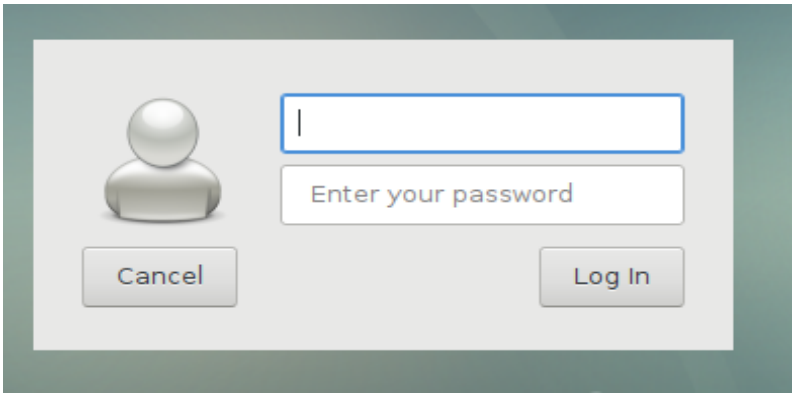
Bolo by jednoduché myslieť si, že spracovaním hesla hash funkciou je heslo v bezpečí. Existuje niekoľko metód, ako zistiť heslo, priamo zo zašifrovanej verzie hesla. Medzi najpoužívanejšie metódy patria „slovníkový útok“ a „útok hrubou silou“. V oboch prípadoch útočník musí poznať hash hesla, ktoré chce získať.

- **Slovníkový útok** využíva súbor (slovník), ktorý obsahuje slová, frázy, často používané heslá alebo iný reťazec znakov, ktorý môže byť potenciálnym heslom. Každé slovo v tomto súbore je spracované hash funkciou a porovnané s hash hodnotou originálneho hesla. Tieto slovníky sú vytvorené extrahovaním slov s dlhých textov, alebo priamo z databáz hesiel.
- **Útok hrubou silou** skúša všetky možné kombinácie znakov zadanej dĺžky. Tieto útoky sú náročné na výpočtový výkon, sú drahé a najmenej efektívne v pomere počet procesorov na čas potrebný k prelomeniu hesla. Heslo by malo byť dostatočne dlhé nato, aby útok hrubou silou zabral toľko času, že sa neoplatí s ním ani začať.

Nie je možné vyhnúť sa týmto typom útokov, ale je možné znížiť ich efektivitu [4].

2.2 Lokálny prístup

Lokálne prihlásenie je základným typom pripojenia. K serveru alebo počítaču máme pripojený monitor a klávesnicu a hneď po spustení OS sa môžeme prihlásiť.



Obrázok 2: Prihlasovacia tabuľka [zdroj: Autor]

Postup prihlásenia prebieha v nasledujúcich krokoch:

1. Proces init overí, či pre dané terminálové spojenie beží program getty.

2. Getty čaká na prihlasovacie meno užívateľa. Po jeho zadaní je getty nahradenýprocesom login, ktorý si následne vyžiada heslo. Heslo je skryté a nie je možné ho vidieť na obrazovke.
3. Loginporovná heslo so záznamom súbore `/etc/shadow`. V prípade nezhody, login znovu zažiada o heslo. Toto sa zopakuje niekoľko krát a po dosiahnutí nastaveného limitu odpojí terminál.
4. Ak boli prihlasovacie údaje správne, login zozbiera a nastaví parametre užívateľa podľa parametrov v súbore `/etc/passwd`.
5. Na terminál sa zobrazí obsah súboru `/etc/motd`.
6. Login spustí príkazový interpreter podľa nastavenia prihlasovaného užívateľa. [1][5].

2.3 Telnet

Protokol Telnet (TELetype NETwork) je jedným z najstarších protokolov sietí TCP/IP. Zaisťuje obojsmernú, znakovo orientovanú komunikáciu vzdialených terminálových zariadení cez sieť. Výhodou protokolu Telnet je schopnosť pripojenia sa na fyzický terminál počítača k inému zariadeniu v sieti, ako keby bol jeho fyzickým terminálom. Protokol Telnet pracuje s modelom klient - server. Klient predstavuje proces požadujúci prístup ku vzdialenému zariadeniu. Server protokolu Telnet je proces, ktorý beží na vzdialenom zariadení a tento proces sa správa ako lokálny terminál.

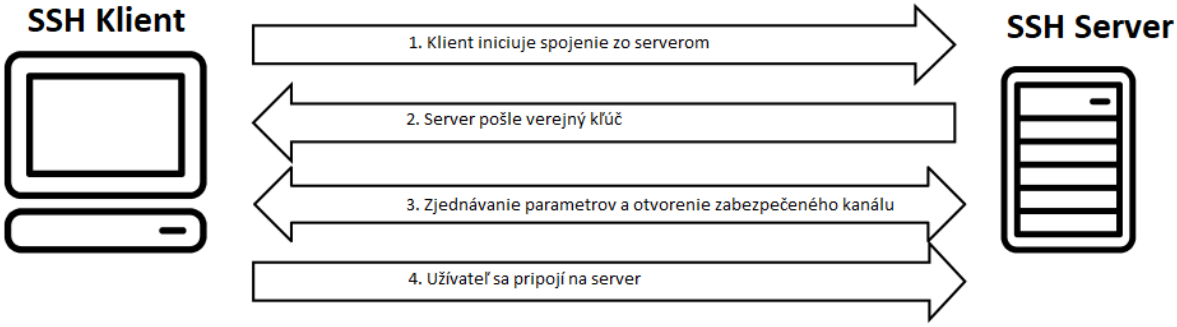
Problémom protokolu Telnet je zabezpečenie. V dobe jeho vytvorenia, bola väčšina pripojených užívateľov ľudia z vládnych alebo výskumných inštitúcii. Nebol dôvod používať šifrovanie a bezpečnostné prvky. Dôsledkom nedostatočnej bezpečnosti, je že pri monitorovaní siete je veľmi jednoduché zistiť heslá užívateľov [14] [15].

2.4 SSH

Ide o bezpečnú (šifrovanú) komunikáciu medzi dvoma počítačmi. SSH (Secure Shell) vznikol ako reakcia na zle zabezpečené protokoly a služby typu Telnet. Vo svojej najzákladnejšej podobe je možné pomocou SSH ovládať vzdialený počítač príkazovým riadkom. Vďaka SSH tunelom je existuje možnosť ovládať aj vzdialenú plochu a teda i grafické rozhranie systému vrátane programov [6].

2.4.1 Princíp protokolu SSH

SSH protokol pracuje s modelom klient – server. To znamená, že pripojenie je iniciované SSH klientom, ktorý sa chce pripojiť na SSH server. SSH klient riadi proces vytvorenia pripojenia, k čomu využíva šifrovanie verejným kľúčom na overenie identity SSH servera. Po prípravnej fáze začne SSH protokol využívať symetrickú šifru a hash algoritmy na zabezpečenie súkromia a integrity dát, ktoré sú vymieňané medzi klientom a serverom. Na nasledujúcom obrázku je možné vidieť zjednodušený proces vytvorenia spojenia [7] :



Obrázok 3: Komunikácia medzi SSH klientom a serverom(spracované podľa [7])

2.4.2 Zabezpečenie autentizácie užívateľov

Existuje niekoľko metód ako zabezpečiť prihlasovanie užívateľov. Medzi najčastejšie metódy patrí používanie hesla a prihlasovanie pomocou verejných kľúčov.

Prihlasovanie pomocou verejných kľúčov je primárne využívané na automatizáciu, alebo systémovými administrátormi na zjednodušenie a urýchlenie prihlásenia. V princípe ide o vytvorenie kľúčového páru – verejného a privátneho kľúča. Nastavenie verejného kľúča na SSH servery slúži na autorizáciu kohokoľvek kto vlastní privátny kľúč [7].

2.4.3 Overovanie klientov pomocou SSH kľúčov

V momente, keď sa klient pokúsi o pripojenie na server pomocou verejného kľúča, informuje o tomto zámere server a povie mu, ktorý verejný kľúč sa chystá použiť. Server následne skontroluje, či sa verejný kľúč nachádza v „authorized_keys“ súbore. Vygeneruje náhodný reťazec, ktorý zašifruje pomocou tohto kľúča a server pošle správu klientovi, aby otestoval či klient vlastní privátny kľúč. Po prijatí správy klient dešifruje správu svojím privátnym kľúčom a správu skombinuje s číslom relácie, ktoré bolo vopred dohodnuté. Klient vytvorí MD5 hash a ten pošle späť serveru, ktorý pozná pôvodnú správu a aj číslo relácie a tak si vie porovnať tieto dve hash hodnoty [8].

2.4.4 Generovanie kľúčového páru

Generovanie kľúčového páru je prvým krokom k prihlasovaniu sa na vzdialený server bez použitia hesla. Na vygenerovanie kľúčového páru sa používa príkaz:

```
ssh-keygen
```

Po zadaní príkazu je možné vybrať umiestnenie, kam sa uloží novo vygenerovaný privátny kľúč. Odporúčané umiestnenie je domovský adresár užívateľa, v ktorom je skrytý (začína znakom „ . “) adresár .ssh. V nasledujúcej výzve je voliteľná možnosť zadať zabezpečujúcu frázu ľubovoľnej dĺžky. SSH túto frázu následne vyžaduje pri každom pripojení. Tento krok nie je povinný, dá sa odignorovať stlačením klávesy ENTER. Je dôležité myslieť na to, ak sa niekto v takomto prípade dostane k privátnemu kľúču môže sa bez problémov pripojiť na server s verejným kľúčom. Po ukončení procedúry vzniknú dva nové súbory:

- `~/.ssh/id_rsa – privátny kľúč`
- `~/.ssh/id_rsa.pub – verejný kľúč`

Základná dĺžka kľúča je 2048 bitov, táto dĺžka sa momentálne považuje za bezpečnú. Je ale možné túto hodnotu zmeniť pomocou prepínača -b napr. [8] :

```
ssh-keygen -b 4096
```

2.4.5 Kopírovanie verejného kľúča na server

Ak má užívateľ prístup na server pomocou hesla a nainštalovaný program `ssh-copy-id` je jednoduché nakopírovať jeho verejný kľúč na server. Program `ssh-copy-id` je súčasťou mnohých distribúcii Linuxu, alebo balíkov OpenSSH. Syntax je :

```
ssh-copy-id užívateľ@vzdialený_systém
```

Po zadaní príkazu bude užívateľ vyzvaný k zadaniu hesla a obsah súboru `~/.ssh/id_rsa.pub` („~” znamená domovský adresár užívateľa) bude prekopírovaný do súboru `~/.ssh/authorized_keys` na servery. Teraz sa môže sa prihlásiť na server bez zadávania hesla [8].

2.4.6 Zabezpečenie SSH servera

Na strane servera je omnoho viac možností ako zvýšiť bezpečnosť prihlasovania pomocou SSH ako na strane užívateľa. Patria medzi ne:

- vypnutie prihlasovania pomocou hesla,

- zmena portu ktorý využíva SSH,
- limitovanie užívateľov ktorí sa môžu pripojiť,
- zakázanie pripojenia užívateľa root,
- a iné.

V praktickej časti je popísané ako implementovať niektoré nastavenia na strane servera [8].

2.4.7 SFTP

SFTP (SSH File Transfer Protocol) je sieťový protokol, ktorý slúži na zabezpečený prenos súborov medzi užívateľom a serverom. Vo väčšine prípadov je súčasťou inštalácie SSH. Používa rovnaký port ako SSH (22) a plne podporuje všetky bezpečnostné funkcie SSH. SFTP nahradilo zastaraný FTP (File Transfer Protocol). Poskytuje rovnakú funkcionalitu ako FTP. Zároveň je viac spoľahlivý a bezpečný pri jednoduchšej konfigurácii. Poskytuje ochranu proti „password sniffing“ a „man in the middle“ útokom. Chráni integritu dát pomocou hash funkcii a autentifikuje server aj užívateľa.

2.4.8 SCP

Program scp (secure copy) je zároveň aj príkaz na prenos súborov pre SFTP v Linuxových systémoch. Rozhranie príkazového riadku pre tento príkaz bolo navrhnuté podľa starého príkazu `rcp` (remote copy) s BSD Unix-u. Syntax príkazu `scp` na kopírovanie súboru od užívateľa na server:

```
scp [-r] súbor ... [užívateľ@]server:[cesta]
```

Týmto spôsobom užívateľ môže skopírovať jeden alebo viac súborov na server. Pri použití prepínača `-r` môžu byť skopírované aj celé adresáre a všetky rekurzívne podadresáre. Kopírovanie v opačnom smere [9] :

```
scp [-r] [užívateľ@]server:súbor cesta
```

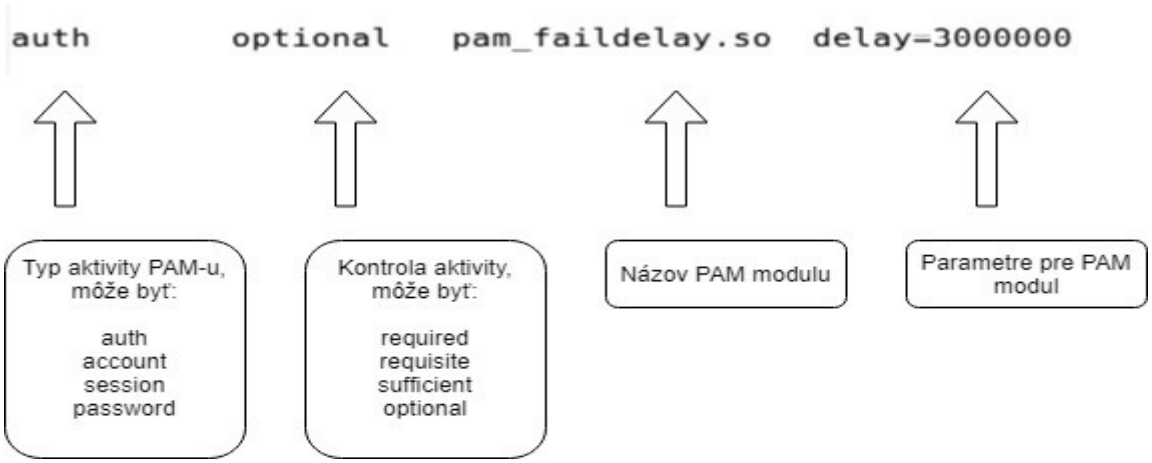
2.5 PAM

Pred rokom 1995, každá aplikácia v unixovom systéme, ktorá potrebovala overiť práva užívateľa, musela vytvoriť vlastné postupy na overovanie užívateľov. Toto prinášalo radu nevýhod a bezpečnostných rizík. Jedným z nich bola komplikácia pri písaní programu, kde sa programátor musí zaoberať vymýšľaním spôsobu ako overiť užívateľov, pričom to nie je primárnou funkciou jeho programu. Ak autor nie je expertom na bezpečnosť, je veľká pravdepodobnosť, že sa v programe vyskytnú bezpečnostné diery, ktoré môže útočník

neskôr odhaliť a využiť. Ďalší problém je skôr filozofický, tento prístup porušuje základnú ideu unixových systémov a to, že každý program by mal robiť iba svoju prácu a nič iné [2].

2.5.1 PAM konfigurácia

PAM (Pluggable Authentication Modules) je vo väčšine linuxových distribúcii predinštalovaný. Jeho hlavný konfiguračný súbor je `/etc/pam.conf`, alebo je jeho konfigurácia rozdelená do viacerých súborov v adresári `/etc/pam.d/`. Ak adresár `/etc/pam.d/` existuje tak je súbor `/etc/pam.conf` ignorovaný. Jadrom programu PAM sú autentizačné moduly. Jedná sa zdieľané objektové súbory, ktoré je možné nájsť v adresári `/lib/security/`. V prípade, že aplikácia chce používať PAM, musí mať vytvorený súbor v `/etc/pam.d/`, ktorý má rovnaký názov ako samotná aplikácia.



Obrázok 4: Syntax konfiguračného súboru PAM (spracované podľa [18])

Prvým parametrom (smerom zľava doprava) je druh aktivity (type) spojeným s prihlasovaním, môže nadobudnúť tieto hodnoty:

- **auth** – počas prihlasovania overuje identitu užívateľa, prípadne iné kontrolné činnosti,
- **account** – prevádza kontrolu účtu, ktorá nie je priamo spojená s identitou užívateľa,
- **session** – aktivuje sa pred a po prevedení služby, bežne nastavuje prostredie užívateľa a obmedzuje systémové prostriedky,
- **password** – v tejto aktivite sa moduly starajú o kontrolu komplexnosti hesiel, aktualizáciu hesiel a iné.

Nasledujúcim parametrom je kontrola aktivity (control), určuje aký vplyv má modul na autentizačný proces, môže nadobudnúť tieto hodnoty:

- **required** – ak modul zlyhá, PAM pošle chybný výsledok aplikácii a nespúšťa ďalšie moduly,
- **requisite** – ak modul zlyhá, PAM pošle chybný výsledok aplikácii a pokračuje v spúšťaní ďalších modulov,
- **sufficient** – ak modul uspeje, PAM pošle výsledok aplikácii a nespúšťa ďalšie moduly,
- **optional** – nezáleží na výsledku tohto modulu, používa sa skôr na vykonanie špecifickej operácie.

Tretím parametrom v syntaxe je samotný modul. Existuje mnoho modulov, niektoré sú rovnaké pre všetky distribúcie Linuxu, iné sa nachádzajú iba v niektorých distribúciách a ďalšie sa dajú nájsť na internete. Tieto moduly patria medzi najčastejšie používané:

- **pam_nologin.so** – ak existuje súbor `/etc/nologin`, nikto okrem užívateľa root sa nemôže prihlásiť,
- **pam_cracklib.so** – kontrola sily hesla podľa zadaných parametrov. Zabráni nastaveniu príliš slabého hesla,
- **pam_permit.so -** autentizácia vždy uspeje,
- **pam_deny.so** – autentizácia vždy zlyhá,
- **pam_limits.so** – nastavenie systémových limitov pre užívateľa (množstvo využitej pamäte, počet spustených procesov a pod.),
- **pam_unix.so** – štandardný unixový modul na autentizáciu, kontroluje užívateľov a heslá podľa `/etc/passwd` a `/etc/shadow` [2] [18] [19].

3 SPRÁVA OPRÁVNENÍ UŽÍVATEĽOV A SKUPÍN

3.1 Súborový systém

Súborový systém je spôsob, akým systém ukladá dáta na disk. Má pevne stanovené pravidlá, ktoré jadro systému používa a dodržuje. Sú potrebné k jednoduchému vyhľadávaniu správnych dát a k ich čo najrýchlejšiemu použitiu.

Základným stavebný prvkom súborového systému je súbor. V OS typu Unix je takmer všetko reprezentované ako súbor a ku všetkému je tak možné pristupovať. Každý port, terminál, disk, adaptér majú svoj súbor, ktorého otvorením s týmto zariadením začnem pracovať. Bežne k týmto súborom pristupujú iba systémové príkazy (programy), užívateľ do nich nezasahuje.

Súbory sú usporiadané do adresárovej štruktúry, na vrchole ktorej je adresár označovaný lomkou, ktorému sa hovorí koreňový adresár. Všetko ostatné je vo vnútri tohto adresára a vnorené do ďalších podadresárov.

Všetky súbory a adresáre majú v rámci svojho umiestnenia jedinečné meno, dátum a čas vytvorenia. Dôležitou súčasťou každého unixového súboru sú prístupové práva. Tie určujú, kto a do akej miery smie zo súborom manipulovať. Výpis sa skladá s desiatich písmen, napríklad:

```
drwxrwxrwx
```

Prvý znak označuje typ súboru, v tomto prípade sa jedná o adresár (d = directory). Ostatné znaky sú tri skupiny práv po troch znakoch. Tieto znaky označujú postupne právo na čítanie (r = read), zápisu (w = write), spustenia (x = execute). V prípade adresárov "x" znamená právo na vyhľadávanie v adresári pre danú skupinu.Prvá trojica je určená pre majiteľa súboru, druhá trojica je pre skupinu vlastníkov a tretia je pre všetkých ostatných užívateľov. Podľa týchto troch skupín sa model nazýva UGO (User Group Others). V prípade ak nie je pridelené niektoré z práv, nahradí ho systém znakom pomlčka (-).

Práva k súborom je možné meniť pomocou príkazu `chmod`, vlastníka pomocou príkazu `chown` a skupinu vlastníkov pomocou príkazu `chgrp`. Z bezpečnostných dôvodov bežní užívatelia nemôžu meniť vlastníka a ani skupinu vlastníkov súborom a to ani súborom, ktoré vlastnia. Túto operáciu môže robiť iba užívateľ root.

Príkaz chmod je možné použiť v dvoch módoch. Prvý mód sa nazýva oktálny a pracuje s číselnými hodnotami 4, 2, 1. Súčty číselných hodnôt predstavujú kombináciu práv. Významovo sa postupuje zľava doprava. Prvé číslo je vlastník súboru, druhé skupina vlastníkov a tretie sú ostatní užívatelia. Hodnota 770 predstavuje číslo 7 (sčítanie hodnôt 1, 2, 4) pre vlastníka a aj skupinu vlastníkov. Číslo 1 predstavuje právo na spúšťanie súboru, číslo 2 predstavuje právo na zapisovanie a číslo 4 právo na čítanie. Po sčítaní týchto čísiel dostaneme hodnotu 7. Číslo 0 na konci hodnoty 770 znamená že ostatní užívatelia nemajú žiadne práva pre daný súbor alebo adresár.

Druhý je tzv. symbolický mód. Formát zápisu je:

```
chmod [ ugoa...][[+ - =] [práva...]...]
```

Pre výber skupiny, ktorej práva sa budú meniť je použitý model UGO s pridaním písmena "a", ktoré znamená zmenu práv pre všetky skupiny užívateľov. Druhou časťou príkazu sú matematické znamienka, pre pridanie práv "+", pre odobranie práv "-" a pre zmenu práv natotožné ako zadané "=". Hodnota práv sa zadáva adekvátnou kombináciou "rwxXst" symbolov. Prehľad symbolov, binárnych a oktálnych hodnôt [13] :

Symbol	*Binárne*	*Oktálne*
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwx	111	7

Obrázok 5: Prevodová tabuľka [13]

Pri pokuse o prístup k súboru sa operačný systém pozrie na výpis jeho prístupových práv, skontroluje jeho majiteľa a zistí, či má daný užívateľ právo operáciu vykonať. Pokiaľ áno, užívateľ ani nezistí že kontrola prebehla, v opačnom prípade je užívateľ systémom informovaný že požadovanú operáciu nemôže vykonať.

Príkladom súboru, kam bežný užívateľ nesmie pristupovať je napríklad `/etc/shadow`, ktorý má práva:

```
-rw-r----- 1 root shadow 1017 Nov 5 15:20 /etc/shadow
```

Z adresárov to môže byť adresár `/sbin`, ktorý obsahuje systémové programy, alebo domovský adresár administrátora `/root`. Väčšina linuxových distribúcii má po inštalácii nastavené práva k dôležitým súborom dostatočne dobre a nie je potrebné ich meniť.

3.1.1 Sticky bit

Pri práci s Linuxom je možné si všimnúť, že okrem práv rwx existujú aj práva s a t:

```
drwxrwxrwt 14 root root 4096 Feb 25 17:13 /tmp
-rwxr-sr-x 1 root crontab 30612 May 19 2015 /usr/bin/crontab
```

Výraz "sticky bit", nazývaný aj "t-bit", je odvodený od jeho vlastnosti. Ak je spustený súbor s týmto atribútom, po jeho ukončení nebude odstránený z RAM (Random Access Memory) pamäti ("stick" - nalepiť). Sticky bit sa označuje písmenom „t“, ktoré nahradilo znak spustenia „x“ pre ostatných užívateľov, prípadne „T“ ak ostatní užívatelia nemajú právo spustenia.

Bežne, ak existuje adresár, kde ktokoľvek môže umiestniť súbory, potom ich aj ktokoľvek môže vymazať (ak má adresár nastavený atribút "writable" - w), čím môže vymazať súbory niekoho iného. Atribút sticky bit toto správanie mení. Ak má adresár nastavený sticky bit, tak iba vlastník súborov smie zmazať svoje súbory.

Sticky bit má ešte jeden význam, ktorý bol v minulosti jeho hlavnou funkciou. Bežne, ak sa program skončí, je uvoľnený z RAM a miesto, kde sa nachádzal bude použité niečím iným. V prípade že kód programu v RAM zostane, pri ďalšom spustení bude program rýchlejšie načítaný, pretože kód nemusí byť opätovne načítaný z disku. Sticky bit je možné nastaviť príkazom `chmod [ugoa]+t <súbor>` alebo `chmod 1000 <súbor>` (namiesto núl je potreba zadať požadované práva) [13].

3.1.2 SUID a SGID

Dôležitým atribútom, ktorý môže spôsobiť bezpečnostné problémy, je takzvaný setuid bit (SUID), ktorý nahradí klasické spúšťacie právo. Pokiaľ je takto označený súbor spustený, dôjde k dôležitej zmene. Bežne, spustený súbor získa rovnaké práva (zdedí ich), ako proces, ktorý ho spustil(užívateľ). Program s nastaveným setuid bitom nezíska práva toho, kto ho spustil, ale práva vlastníka súboru. Toto je veľmi užitočná vlastnosť, ktorú využívajú programy, ktoré na svoju činnosť potrebujú práva užívateľa root. Príkladom takéhoto programu je `passwd`, tento program slúži na zmenu hesla. Aby mohol vykonávať svoju činnosť potrebuje prístup do súboru `/etc/shadow`, ale ako je spomenuté vyššie, tak

do tohto súboru má práva zapisovať iba užívateľ root. V prípade že by parameter setuid neexistoval, nikto iný okrem užívateľa root by nesmel zmeniť heslá užívateľom. Program `passwd` má nasledujúce práva:

```
-rwsr-xr-x 1 root root45648 May 17 2017 passwd
```

Setuid bit sa označuje písmenom „s“, ktoré nahradilo znak spustenia „x“, prípadne „S“ ak vlastník alebo skupina vlastníkov nemajú právo spustenia.

Aj napriek svojej užitočnosti, s funkcie setuid plynie aj niekoľko zásadných rizík. Nepodstatná chyba v programe, môže jednoducho dovoliť útočníkovi preniknúť do systému. Napríklad tým, že by pomocou neho spustil shell, čím by tento program prevzal práva programu s nastaveným setuid, čiže root práva.

Ak má systém viac administrátorov je dobré občas skontrolovať, ktoré súbory vlastnené užívateľom root majú nastavený setuid. Je možné použiť nasledujúci príkaz:

```
find / -perm -4000 –uid 0 -print
```

Tento príkaz prehľadá celú adresárovú štruktúru, začínajúc koreňovým adresárom a vypíše všetky súbory vlastnené užívateľom root s nastaveným setuid bitom [2][13].

3.1.3 **Umask**

Program umask je obdobou programu chmod, ale pracuje s opačnými bitmi a slúži pre všeobecné – globálne nastavenie práv súborov a adresárov. Program Umask nastavuje implicitné (default) práva pre vytvorené, alebo skopírované súbory a adresáre. Funkcia programu je pre súbory a adresáre totožná, ale jeho správanie pri súboroch je mierne odlišné.

Adresár, bude mať implicitne také práva, ktoré sa zadefinovali pomocou príkazu `umask`. Ak chce užívateľ priradiť všetky práva vlastníkovi(7), skupine vlastníkov chce dať právo čítať a vyhľadávať v adresári (5) a ostatným užívateľom nedať žiadne práva (0), čo inak docieli zadaním `chmod 750 /tmp/test/`. Ďalším krokom je odčítanie každého čísla uvedeného v zátvorke od čísla 7, a tak dostane 0 - pre vlastníka (7 - 7 = 0), 2 - pre skupinu vlastníkov (7 - 5 = 2) a 7 - pre ostatných (7 - 0 = 0). Následne do inicializačného skriptu pridá príkaz:

```
umask 027
```

Po pridaní príkazu, bude mať užívateľ nastavený parameter umask s hodnotou 027 pri každom prihlásení a spustení inicializačného skriptu.

Rozdiel pri vytváraní súborov je odčítavanie hodnôt práv od čísla 6. Dôvodom tohto rozdielu je zamedzenie vytvárania okamžite spustiteľných súborov. Užívateľ musí vždy po vytvorení alebo kopírovaní súboru priradiť práva na spúšťanie ručne [13].

3.2 Sudo

Sudo má pomerne dlhú históriu, jeho prvá verzia pracovala na BSD OS okolo roku 1980. Je to slobodne šíriteľný softvér dostupný pod „Internet system Consortium“(ICS) licenciou. Preto, že má sudo vplyv na bezpečnosť, odporúča sa nainštalovať stabilnú verziu (momentálne 1.8.22 vydaná 16.1 2018) [11].

Sudo je skratka, ktorá znamená buď „substitute user do “(náhradný užívateľ robí) alebo „super user do“(super užívateľ robí). Sudo povolí užívateľovi spustiť program ako iný používateľ (najčastejšie to je root), táto možnosť je nevyhnutná pre mnohé linuxové distribúcie.

3.2.1 Rozdiel medzi sudo a su

V tradičnom linuxovom prostredí sa na získanie práv root používa príkaz `su`. Je možné použiť príkaz `su -` na priame prepnutie sa na užívateľa root. Tento prístup sa nepovažuje za najlepší a prináša mnohé bezpečnostné riziká.

V distribúciách založených na programe sudo nie je možné prihlásiť sa ako užívateľ root. Napríklad v distribúcii Ubuntu je užívateľ root úplne „zablokovaný“ a nefunguje ani príkazu `su -`. K dosiahnutiu administrátorských je nutné všetky príkazy spúšťať pomocou programu sudo [10].

3.2.2 Používanie programu sudo

Vo svojej podstate je používanie suda veľmi jednoduché. Napríklad pri inštalácii programu na prehrávanie hudby a videa mplayer pomocou príkazu zypper, ako bežný užívateľ. Po zadaní príkazu `apt-get install mplayer2` systém vypíše chybové hlásenie o nedostatku práv na spustenie tohto príkazu. Aby bolo možné program nainštalovať, pred pôvodný príkaz je potrebné napísať `sudo`. Ak má užívateľ pridelené potrebné práva v súbore`/etc/sudoers`, príkaz teraz prebehne bez chybového hlásenia a program sa nainštaluje [10].

3.2.3 Konfigurácia a nastavenie programu sudo

Hlavným konfiguračným súborom programu sudo je `/etc/sudoers`. Tento súbor má práva „400“, čo znamená že nikto do neho nemôže písať alebo ho spúšťať a čítať ho môže iba vlastník root. Aby bolo možné modifikovať tento súbor je potrebné použiť špeciálny program visudo. Visudo sa stará o práva, nastavuje zámok na súbor ktorý sa ním modifikuje aby ho dvaja používatelia nemohli upravovať naraz a ešte pred uložením skontroluje či nevznikla úpravou chyba. Posledná vlastnosť je veľmi dôležitá, ak sa v tomto súbore vyskytne chyba, tak sudo prestane úplne pracovať a nikto nebude môcť používať privilegované príkazy. V prípade že heslo priamo na užívateľa root nie je známe, nastane situácia kde je nutné prepnúť sa do „single-user“ módu a heslo pre užívateľa root tam zmeniť.

3.2.4 Formát sudoers konfiguračného súboru

Formát `/etc/sudoers` súboru pozostáva z dvoch druhov zápisov:

Aliases a **užívateľské špecifikácie** (určujú kto môže čo spúšťať). Ak sa pre jedného užívateľa zhoduje viacero záznamov, aplikujú sa po poradí zo začiatku súboru.

Existujú štyri druhy aliasov:

- **user alias**: špecifikuje alias pre jedného užívateľa alebo skupinu užívateľov. Jeden užívateľ môže byť zaradený do viacerých aliasov,
- **„Run as“ alias**: špecifikuje iných užívateľov pod ktorými môže sudo užívateľ pracovať. Štandardne sa príkazy spúšťajú ako užívateľ root,
- **host alias**: špecifikuje systém na ktorý sa práva vzťahujú. Tento alias sa využíva v prostredí kde je viac počítačov alebo serverov,
- **command alias**: špecifikuje synonymum pre konkrétny príkaz. Napríklad je jednoduchšie napísať APT ako písať absolútnu cestu príkazu `/usr/sbin/apt-get`.

Používanie aliasov nie je povinné, ale zjednodušujú budúce úpravy sudoers konfigurácie.

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Obrázok 6: Predvolené nastavenie /etc/sudoers súboru [zdroj: Autor]

Je možné zaradiť ďalšie sudoers súbory, ktoré budú zahrnuté do pridelenia práv užívateľom pomocou `#include` a `#includedir` príkazov. Táto možnosť sa môže použiť na zachovanie globálneho sudoers súboru a navyše sa bude používať lokálna sudoers konfigurácia. Na zahrnutie súboru `/etc/sudoers.local` je potrebné zapísať nasledovný riadok do súboru `/etc/sudoers`:

```
#include sudoers.local
```

V prípade ak zahrnieme do konfigurácie:

```
#includedir /etc/sudoers.d
```

Sudo prestane spracovávať súčasný súbor a začne čítať každý súbor v `/etc/sudoers.d/`, preskočí súbory ktoré začínajú znakom ‘~’ a obsahujú v názve znak ‘.’ [11][12].

3.3 SELinux

Každý operačný systém používa mechanizmus riadenia prístupu. Medzi dva najznámejšie patrí diskrétne riadenie prístupu, ktoré sa používa v klasickom Linuxe, a direktívne riadenie prístupu, ktoré využíva SELinux (Security Enhanced Linux). Oba prístupy riadenia môžu spolu existovať na systéme zároveň.

SELinux je bezpečnostné rozšírenie pre Linuxové systémy, ktoré pridáva nové možnosti užívateľom a administrátorom kontrolovania prístupov k zdrojom systému. SELinux je implementáciou direktívneho riadenia prístupu, vynucuje administrátorom zadefinovanú

bezpečnostnú politiku nad všetkými objektmi v systéme. Je tak možné nastaviť citlivejšie prístupové práva k dátam a zamedziť ich zmenám [20].

4 AKTIVITA UŽÍVATEĽOV A SYSTÉMU

4.1 Logovanie

Logovanie je činnosť, ktorú vykonávajú všetky operačné systému unixového typu. Jedná sa o vedenie záznamov o dôležitých aktivitách a udalostiach, ktoré sa stanú na systéme. Tieto záznamy sú dôležité pri hľadaní problémov, ktoré sa vyskytnú počas behu systému, alebo pri napadnutí systému. Je možné spätne vyhľadať aktivitu užívateľov a nájsť tak presný čas a užívateľa, ktorý spôsobil problém, alebo vykonával aktivitu na systéme. Hlavný súbor so systémovými logmi je `/var/adm/syslog`. Tento a ďalšie súbory sú vytvárané rsyslog démonom. Rsyslog démon zachytáva všetky dôležité hlásenia a triedi ich do príslušných súborov. Jeho konfiguračný súbor je `/etc/rsyslog.conf`, v tomto súbore je sekcia pre globálne pravidlá programu. Tieto pravidlá musia začínať znakom „$“. Nachádza sa tam napríklad nastavenie vlastníka logov, skupinu vlastníkov, alebo veľkosť a rotáciu logov a iné.

```
###########################
#### GLOBAL DIRECTIVES ####
###########################

#
# Use traditional timestamp format.
# To enable high precision timestamps, comment out the following line.
#
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

#
# Set the default permissions for all log files.
#
$FileOwner root
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
```

Obrázok 7: Globálne nastavenia pre rsyslog démon[zdroj: Autor]

Nasledujúcou sekciou sú pravidlá, kde sa každé pravidlo (riadok) skladá z dvoch častí. Prvá časť je ešte rozdelená do dvoch podčastí a tie sú oddelené bodkou. Prvá podčasť označuje oblasť prichádzajúcich hlásení, druhá ich prioritu. Priorita môže nadobúdať hodnoty napríklad: info, notice, warning, debug a iné. Druhá časť pravidla označuje súbor, kam sa logy budú zapisovať, ak vyhovujú zadanému pravidlu [2][16].

```
###############
#### RULES ####
###############

#
# First some standard log files.  Log by facility.
#
auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log
```

Obrázok 8: Pravidlá nastavené pre rsyslog démon [zdroj: Autor]

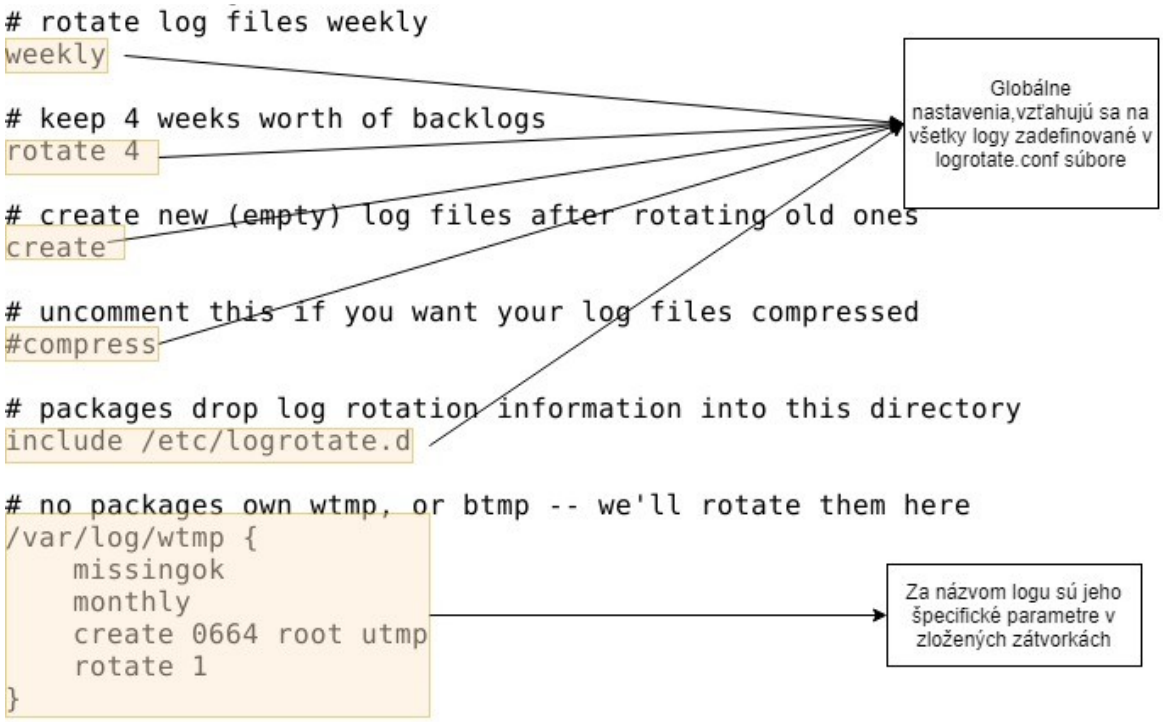
4.2 Systémové logy

Pre každého kto pracuje v unixovom alebo linuxovom prostredí je veľkou výhodou znalosť jednotlivých logovacích súborov. V nasledujúcej časti sú popísané najdôležitejšie logy, ktoré je možné nájsť v linuxovom systéme:

1. **/var/log/messages** – Obsahuje všeobecné systémové správy, vrátane správ vygenerovaných počas štartovania systému. Neobsahuje debugovacie správy ani kritické správy.
2. **/var/log/auth.log** – Obsahuje informácie o prihlasovaní užívateľoch a spôsobu, akým boli autentifikovaní.
3. **/var/log/daemon.log** – Obsahuje informácie o démonoch ktorý bežia na pozadí systému.
4. **/var/log/dpkg.log** – Obsahuje správy generované počas inštalácie alebo odstraňovaní balíčkov príkazom dpkg.
5. **/var/log/kern.log** – Obsahuje správy vytvorené jadrom. Vhodné na vyhľadávanie porúch, v prípade používania vlastného alebo upraveného jadra.
6. **/var/log/lastlog** – Ukladajú sa tam informácie o posledných prihláseniach užívateľov. Tento súbor nie je vo formáte ASCII, na čítanie logu je potrebné použiť príkaz lastlog.
7. **/var/log/maillog /var/log/mail.log –** Obsahuje správy z poštového serveru, ktorý beží na systéme.
8. **/var/log/user.log** – Obsahuje všetky logy o užívateľoch
9. **/var/log/cron** – Kedykoľvek cron démon spustí úlohu, informácie o tejto aktivite sa uložia do tohto súboru.
10. **/var/log/faillog** – Obsahuje informácie o neúspešných prihláseniach užívateľov. Na zobrazenie obsahu tohto súboru je nutné použiť príkaz faillog.

11. **/var/log/secure** – Obsahuje informácie spojené s prihlasovaním a autorizovaním práv. Napríklad sa tu nachádzajú správy z sshd démona.

Toto boli niektoré z mnohých logovacích súborov, ktoré sa nachádzajú na linuxových systémoch. Tieto súbory môžu časom narásť do desiatok až stoviek megabajtov, ak na systéme nie je nastavené rotácie logovacích súborov. Na túto rotáciu logov sa často používa program „logrotate“, ktorý pomocou jednoduchej konfigurácie umožní nastaviť rotáciu separátne pre každý log. Logrotate sa štandardne spúšťa každý deň z crontabu užívateľa root (interval je možné zmeniť). Pričom prechádza konfiguračný súbor `/etc/logrotate.conf`, prípadne zo súborov nachádzajúcich sa v `/etc/logrotate.d` [17].



Obrázok 9: Nastavenie rotácie logov pomocou logrotate,(spracované podľa [17])

4.3 Audit

Auditovací démon(ďalej Audit) v linuxovom OS je rozhranie, ktoré umožňuje zaznamenávať zadefinované udalosti na systéme. Audit je zodpovedný za zápis relevantných informácii na disk. Taktiež poskytuje užitočné príkazy, ako napr., *ausearch* a *aureport*, na prehľadávanie a analýzu systémových záznamov vytvorených auditom. Konfigurácia pravidiel auditu sa vykonáva príkazom *auditctl.*

Pomocou tohto rozhrania, systém môže sledovať mnoho typov udalostí, ako napríklad:

- zaznamenať, kto upravil špecifický súbor,
- odhaliť neautorizované zmeny súborov,
- monitorovanie volaní a funkcii systému,
- ukladanie príkazov zadaných užívateľmi,
- odhaliť neočakávané ukončenie procesov,
- a iné.

4.3.1 Konfigurácia

Konfigurácia audit je kontrolovaná nasledujúcimi súbormi:

- **audit.conf** - slúži na nastavenie auditu so zameraním na to, kam a ako sa budú záznamy o udalostiach ukladať. Taktiež môže definovať, ako sa audit bude správať pri rotácii logu, pri zaplnení disku, počet uložených rotácii,
- **audit.rules** – slúži na nastavenie udalostí, ktoré majú byť zaznamenané. Po inštalácii nie sú zadefinované žiadne pravidlá.

4.3.2 Princíp fungovania auditu v praxi

V súbore `/etc/audit/audit.rules` existuje nasledujúce pravidlo:

```
-w /etc/ssh/sshd_config -p warx -k sshd_config
```

Ak beží audit, a užívateľ root spustí nasledujúci príkaz:

```
# cat /etc/ssh/sshd_config
```

V súbore `/etc/audit/audit.log` sa zobrazí záznam o aktivite, podľa zadefinovaného pravidla.

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 suc-
cess=no exit=-13 a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686
pid=3538 auid=500 uid=500 gid=500 euid=500 suid=500 fsuid=500 egid=500
sgid=500 fsgid=500 tty=pts0 ses=1 comm="cat" exe="/bin/cat"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="sshd_config"

type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"

type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config"
inode=409248 dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00
obj=system_u:object_r:etc_t:s0
```

Obrázok 10: Záznam zo súboru audit.log [21]

Záznam uvedený na obrázku 10 pozostáva z troch častí (každá začína type=), ktoré zdieľajú totožnú časovú stopu a sériové číslo [21].

Tabuľka 2 Popis zápisu v súbore /etc/audit/audit.log [21]

Pole	**Popis Pola**
type=SYSCALL	Pole type obsahuje typ záznamu. Hodnota SYSCALL špecifikuje, či tento záznam bol spustený systémovým volaním jadra.
msg=audit(1364481363.243:24287):	Pole msg sa zapisuje vo formáte audit(časová_stopa:ID). Niekoľko zápisov môže zdieľať rovnakú časovú stopu a ID, v prípade ak boli vygenerované totožnou udalosťou.
arch=c000003e	Pole arch obsahuje informácie o architektúre procesora na systéme. Hodnota je preložená do hexadecimálnej sústavy. V tomto prípade sa jedná o x86_64.
syscall=2	Pole syscall zaznamenáva typ systémového volania. V tomto prípade sa jedná o volanie „open“.
success=no	Pole success zaznamenáva výsledok operácie
exit=-13	Pole exit zaznamenáva hodnotu, ktorá špecifikuje návratový kód systémového volania.
a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a	Polia a0 až a3 zaznamenávajú prvé 4 argumenty systémového volania v hexadecimálnej sústave.
ppid=268	Pole ppid zaznamenáva číslo rodičovského (parent) procesu.
comm=“cat“	Pole comm zaznamenáva príkaz, ktorý spustil analyzovací proces.
exe=“/bin/cat“	Pole exe zaznamenáva celú cestu k príkazu, ktorý spustil analyzovací proces
Type=CWD	Hodnota CWD (current working directory) zaznamená aktuálny adresár z ktorého bolo systémové volanie vyvolané. Cieľom tohto záznamu je schopnosť zostaviť absolútnu cestu.

Pole	**Popis Pola**
cwd="/home/shadowman"	Pole cwd zaznamená cestu k adresáru z ktorého bolo systémové volanie vyvolané.
type=PATH	Hodnota PATH zaznamená všetky cesty, ktoré boli použité ako argument k systémovému volaniu.
name="/etc/ssh/sshd_config"	Pole name zaznamenáva cestu, ktorá bola použitá ako argument k systémovému volaniu.
inode=409248	Pole inode zaznamenáva inode číslo spojené zo súborom, ktorý bol použitý ako argument k systémovému volaniu.
dev=fd:00	Pole dev zaznamenáva hlavné a vedľajšie číslo zariadenia na ktorom sa zaznamenaný adresár nachádza.
mode=0100600	Pole mode zaznamenáva práva súboru alebo adresára v číselnej podobe.
ouid=0	Pole ouid zaznamenáva ID vlastníka súboru.
ogid=0	Pole ouid zaznamenáva ID skupinu vlastníkov súboru.
rdev=00:00	Pole rdev zaznamenáva identifikátor zariadenia pre špeciálne súbory. V tomto prípade sa jedná o bežný súbor, preto nulová hodnota.
obj=system_u:object_r:etc_t:s0	Pole obj zaznamenáva SELinux kontext s akým bol súbor alebo adresár v čase vyvolania označený.

II. PRAKTICKÁ ČÁST

5 IMPLEMENTÁCIA A KONFIGURÁCIA V OS LINUX

Praktická časť práce je venovaná zabezpečeniu OS Linux v spojitosti so správou užívateľov a prístupov do systému. Implementácia je rozdelená do piatich samostatných kategórii. Každú z nich je možné použiť nezávisle na ostatných, prípadne vybrať iba niektoré nastavenia pre špecifické potreby systému. Každý bod je rozdelený do troch častí:

- **Popis** – opisuje čo nastavenie robí
- **Zdôvodnenie** – opisuje prečo sa nastavenie implementuje
- **Realizácia** – opisuje ako sa nastavenie implementuje

Po implementácii uvedených nastavení by mal byť výsledkom bezpečnejší, spoľahlivejší a transparentnejší systém pre užívateľov aj administrátorov.

5.1 PAM konfigurácia

Autentifikácia pomocou hesiel je kľúčová pre bezpečnosť systému. Ak sú heslá príliš slabé, pravdepodobnosť že sa útočníkovi podarí získať, alebo prelomiť heslo sa veľmi zvyšuje. V nasledujúcich bodoch je popísané, ako môže administrátor nastaviť PAM, tak aby upravil komplexnosť hesiel podľa bezpečnostných požiadavkou, ako predchádzať recyklovaniu hesiel, alebo ako zablokovať užívateľa po istom počte neúspešných pokusoch o prihlásenie.

5.1.1 Nastavenie kritérií na vytváranie hesiel

Popis: Modul pam_cracklib.so kontroluje silu hesiel. Vykonáva kontroly ako napr., či nie je heslo v slovníku, jeho dĺžku, či obsahuje špecifické znaky (veľké, malé písmeno, číslo, špeciálny znak a iné). Zoznam parametrov, ktoré budú implementované:

- try_first_pass - získa heslo s predchádzajúceho PAM modulu, ak nie je k dispozícii, požiada užívateľa o heslo,
- retry=3 - povolí 3 pokusy o zadania hesla,
- minlen=10 - minimálna akceptovaná veľkosť hesla,
- dcredit=-1 - heslo musí mať minimálne jedno číslo,
- ucredit=-1 - heslo musí mať minimálne jeden veľký znak,
- ocredit=-1 - heslo musí mať minimálne jeden špeciálny znak,
- lcredit=-1 - heslo musí mať minimálne jeden malý znak.

Zdôvodnenie: Silnejšie heslá poskytujú lepšiu ochranu systému pred útokmi hrubou silou.

Realizácia: Pridať do súboru `/etc/pam.d/common-password` nasledujúci riadok:

```
password requisite pam_cracklib.so try_first_pass retry=3 minlen=10 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1
```

5.1.2 Zablokovanie užívateľa po neúspešných prihláseniach

Popis: Zablokovanie užívateľa po *n* neúspešných pokusoch o prihlásenie sa delí na dve časti. Prvou časťou je úprava PAM konfiguračných súborov. Druhou časťou je aplikovanie týchto zmien priamo do programov (v rámci programu PAM), ktoré môžu blokovať užívateľov. Odporúča sa skontrolovať dokumentáciu všetkých sekundárnych programov, ktoré túto aktivitu vykonávajú a nastaviť ich tak aby pracovali s PAM programom.

Zdôvodnenie: Zablokovanie užívateľa po *n* neúspešných po sebe nasledujúcich pokusoch o prihlásenie zmierňuje útoky hrubou silou na systém.

Realizácia: Pridať do súboru `/etc/pam.d/common-auth` nasledujúci riadok:

```
auth required pam_tally2.so onerr=fail audit silent deny=5 unlock_time=900
```

Pridať do súboru `/etc/pam.d/common-account` nasledujúci riadok:

```
account required pam_tally2.so
```

5.1.3 Obmedzenie znovu použitia hesla

Popis: Súbor `/etc/security/opasswd` uskladňuje staré heslá užívateľov, ktoré slúžia ako kontrola proti recyklácii hesiel.

Zdôvodnenie: Prinútiť užívateľa aby nemohol používať 5 starých hesiel, zvyšuje šancu že útočník neuhádne jeho heslo.

Realizácia: Upraviť súbor `/etc/pam.d/common-password` pridaním nasledujúceho riadku:

```
password required pam_pwhistory.so remember=5
```

5.1.4 Nastavenie hash algoritmu pre heslá

Popis: Zmena enkrypcie hesiel z MD5 na SHA-512 algoritmu. Všetci existujúci užívatelia musia byť vyzvaní zmeniť si heslo, aby ich heslá boli vytvorené novým algoritmom.

Zdôvodnenie: SHA-512 je silnejší hash algoritmus ako MD5, čím zvyšuje úsilie potrebné na prelomenie hesla útočníkom.

Realizácia: Upraviť súbor `/etc/pam.d/common-password` pridaním nasledujúceho riadku:

```
password required pam_unix.so sha512
```

5.2 Užívatelia a ich prostredie

Nad rámec komplexnosti hesiel je vhodné upraviť aj ďalšie parametre, ktoré vynucujú od užívateľov kroky na zvýšenie bezpečnosti systému. Tieto a iné parametre sú zadefinované v súbore `/etc/login.defs`. Tento súbor poskytuje parametre väčšine príkazov spojených zo správou užívateľov v linuxových systémoch (napr., useradd, usermod, groupadd, a iné). Po zmene niektorého z parametrov, je nutné zmeniť túto hodnotu ručne všetkým užívateľom, ktorý na systému už existujú a majú nastavenú pôvodnú hodnotu. Ďalšie nastavenia sa budú zaoberať preventívnymi opatreniami na zníženie hrozby útoku a zredukovať útočníkove možnosti využiť slabinu na systéme.

5.2.1 Nastavenie exspirácie hesiel

Popis: Parameter PASS_MAX_DAYS v súbore `/etc/login.defs` umožňuje administrátorovi nastaviť dobu, po ktorej heslo exspiruje a užívateľ ho musí zmeniť. Odporúčaná doba exspirácie je 90 alebo menej dní.

Zdôvodnenie: Príležitosť pre útočníka zneužiť kompromitované prihlasovacie údaje alebo úspešne získať prihlasovacie údaje útokom hrubou silou je limitovaná vekom hesla. Z tohto dôvodu, zredukovanie maximálneho veku hesla zmenšuje útočníkovi šancu na získanie prihlasovacích údajov a ich zneužitie.

Realizácia: Upraviť parameter PASS_MAX_DAYS v súbore `/etc/login/defs`:

```
PASS_MAX_DAYS 90
```

Zmeniť tento parameter pre všetkých existujúcich užívateľov:

```
# chage --maxdays 90 <užívateľ>
```

5.2.2 Nastavenie minimálnej doby na zmenu hesla

Popis: Parameter PASS_MIN_DAYS v súbore `/etc/login.defs` umožňuje administrátorovi zabezpečiť, aby si užívateľ nemohol zmeniť heslo skôr ako uplynie zadaná doba. Odporúčaná doba je 7 alebo viac dní.

Zdôvodnenie: Obmedzením frekvencie s akou môže užívateľ meniť heslo, administrátor môže predísť snahe užívateľa opakovane meniť heslo. Čím by efektívne mohol obísť pravidlo na znovu použitie starých hesiel (vid. 5.1.3).

Realizácia: Upraviť parameter PASS_MIN_DAYS v súbore `/etc/login/defs`:

```
PASS_MIN_DAYS 7
```

Zmeniť tento parameter pre všetkých existujúcich užívateľov:

```
# chage --mindays 90 <užívateľ>
```

5.2.3 **Nastavenie zablokovania neaktívnych účtov**

Popis: Užívateľské účty, ktoré boli neaktívne danú dobu po exspirácii hesla môžu byť automaticky zablokované. Odporúčaná doba na zablokovanie účtu je 30 dní po exspirácii hesla.

Zdôvodnenie: Neaktívne účty predstavujú bezpečnostnú hrozbu pre systém tým, že sa neprihlasujú a teda nie sú vyzvaný k zmene hesla.

Realizácia: Spustením nasledujúceho príkazu sa nastaví hodnota „nečinnosti hesla“ pre nových užívateľov na 30 dní:

```
# useradd -D -f 30
```

Zmeniť tento parameter pre všetkých existujúcich užívateľov:

```
# chage --inactive 30 <užívateľ>
```

5.2.4 **Obmedzenie prihlasovania systémových účtov**

Popis: S inštaláciou každej distribúcie Linxu sú vytvorení systémový užívatelia, ktorí sa využívajú na správu aplikácii. Interaktívny shell nie je určení pre týchto užívateľov.

Zdôvodnenie: Je dôležité aby systémový užívatelia, ktorí nie sú používaní bežnými užívateľmi nemohli používať interaktívny shell. Je odporúčané nastaviť shell na hodnotu `/sbin/nologin`. Toto zabraňuje zneužitiu týchto účtov na spúšťanie akýchkoľvek príkazov.

Realizácia: Nastavenie parametru shell na `/sbin/nologin` pomocou príkazu:

```
# usermod –s /sbin/nologin <užívateľ>
```

5.2.5 Nastavenie hodnoty umask

Popis: Hodnota parametru umask určuje s akými právami sa vytvorí súbor užívateľom. Vlastník súboru sa môže sám rozhodnúť, aké práva nastaví vytvorenému súboru pre skupinu vlastníkov a pre ostatných užívateľov.

Zdôvodnenie: Nastavením bezpečnejšej hodnoty umask slúži ako prevencia, aby užívateľ nevytvoril súbory s príliš voľnými právami. V tomto prípade bude musieť vedome zmeniť práva súboru na voľnejšie.

Realizácia: Upraviť súbory `/etc/bashrc.local` a `/etc/profile.local` (a súbory pre ostatné používane shell-y na systéme) pridaním nasledujúceho riadku:

```
umask 027
```

5.2.6 Nastavenie hesla všetkým užívateľom

Popis: Ak účet nemá pridelené heslo, ktokoľvek sa naň môže prihlásiť bez zadávania hesla.

Zdôvodnenie: Každý účet na systéme musí mať pridelené heslo, alebo musí byť zablokovaný, pretože inak by mohol neautorizovaný užívateľ získať prístup do systému.

Realizácia: Ak ktorýkoľvek užívateľ nemá heslo v súbore `/etc/shadow`, je nutné ho zablokovať nasledujúcim príkazom:

```
# passwd -l <username>
```

5.2.7 Nastavenie práv všetkých domovských adresárov

Popis: Administrátor môže zabezpečiť bezpečné práva pre domovské adresáre užívateľov pomerne jednoducho, ale užívatelia môžu tieto práva v priebehu času zmeniť.

Zdôvodnenie: Práva pri ktorých sú ostatní užívatelia alebo skupina vlastníkov oprávnení zapisovať do domovského adresára, môžu viesť k zmene alebo zneužitiu dát v tomto adresári.

Realizácia: Úprava práv domovských adresárov bez informovania užívateľov môže viesť k neočakávaným problémom. Preto sa odporúča nastaviť pravidelnú kontrolu domovských adresár a informovať užívateľov v prípade zmeny ak práva nesúhlasia s nastavenými štandardom.

5.3 Zabezpečenie SSH servera

SSH je bezpečná a kryptovaná náhrada za zastarané prihlasovacie metódy ako telnet, ftp, rlogin, rsh a rcp. Je odporúčané opustiť spomenuté staršie metódy prihlasovania a používať SSH, ako ochranu pred odchytením komunikácie a získaním citlivých dát. Pre docielenie vyššej bezpečnosti je po inštalácii programu SSH nutné nastaviť niekoľko dôležitých parametrov. V nasledujúcej časti je popísané, ktoré parametre to sú a ako ich nastaviť.

5.3.1 Nastavenie práv pre /etc/ssh/sshd_config súbor

Popis: Súbor `/etc/ssh/sshd_config` obsahuje konfiguráciu sshd démona.

Zdôvodnenie: Súbor `/etc/ssh/sshd_config` musí byť chránený pred zmenami neoprávnenými užívateľmi. Mal by byť čitateľný pre všetkých užívateľov, informácie z tohto súboru sú využívané mnohými neprivilegovanými programami.

Realizácia: Spustením nasledujúcich príkazov sa upravia vlastníci a práva súboru:

```
# chown root:root /etc/ssh/sshd_config
# chmod og-wx /etc/ssh/sshd_config
```

5.3.2 Nastavenie parametru Protocol

Popis: SSH podporuje dva rôzne a nekompatibilné protokoly: SSH v1 a SSH v2.

Zdôvodnenie: SSH v1 dnes už nespĺňa bezpečnostné požiadavky a odporúča sa používať SSH v2.

Realizácia: Pridať nasledujúci riadok do súboru `/etc/ssh/sshd_config`:

```
Protocol 2
```

5.3.3 Nastavenie parametru MaxAuthTries

Popis: Parameter MaxAuthTries určuje maximálny počet neúspešných pokusov o pripojenie. Ak počet neúspešných pripojení dosiahne polovicu nastavenej hodnoty, chybové hlásenie bude zapísané do `/var/log/syslog` súboru s popisom chyby.

Zdôvodnenie: Nastavenie MaxAuthTries na nízku hodnotu minimalizuje úspešnosť útoku hrubou silou na SSH server.

Realizácia: Pridať nasledujúci riadok do súboru `/etc/ssh/sshd_config`:

```
MaxAuthTries 4
```

5.3.4 Nastavenie parametru IgnoreRhosts

Popis: Parameter IgnoreRhosts špecifikuje že súbory `.rhosts` a `.shosts` nemôžu byť použité pri autentifikácii.

Zdôvodnenie: Nastavením tohto parametra je užívateľ požiadaný o zadanie hesla aj v prípade ak použije `HostbasedAuthentication` alebo `RhostsRSAAuthentication`.

Realizácia: Pridať nasledujúci riadok do súboru `/etc/ssh/sshd_config`:

```
IgnoreRhosts yes
```

5.3.5 Nastavenie parametru PermitRootLogin

Popis: Parameter PermitRootLogin určuje, či sa užívateľ root smie prihlásiť priamo na systém.

Zdôvodnenie: Zamedzením priameho prihlásenia užívateľa root vynúti administrátorov používať svoje osobné účty a potom sa pomocou príkazu `sudo` alebo `su` prepnúť na užívateľa root. Týmto krokom sa dosiahne jasná dohľadateľnosť dôkazu v prípade bezpečnostného incidentu.

Realizácia: Pridať nasledujúci riadok do súboru `/etc/ssh/sshd_config`:

```
PermitRootLogin no
```

5.3.6 Nastavenie parametru PermitEmptyPassword

Popis: Parameter PermitEmptyPassword určuje či SSH server prijme prihlásenie s prázdnym heslom.

Zdôvodnenie: Znemožnenie prihlásenia s prázdnym heslom znižuje riziko neoprávneného prístupu na systém.

Realizácia: Pridať nasledujúci riadok do súboru `/etc/ssh/sshd_config`:

```
PermitEmptyPassword no
```

5.3.7 Nastavenie parametrov ClientAliveInterval a ClientAliveCountMax

Popis: Parameter ClientAliveInterval určuje ako dlho môže byť SSH pripojenie neaktívne, po tejto dobe sa pripojenie ukončí. Ak je nastavený parameter ClientAliveCountMax, v prípade neaktívneho pripojenia SSH pošle správu užívateľovi toľko krát ako je zadaná

hodnota ClientAliveCountMax. Napríklad hodnota ClientAlive interval bude nastavena na 30 a hodnota ClientAliveCountMax bude 5, pripojenie sa ukončí po 150 sekundách.

Zdôvodnenie: Nenastavením týchto parametrov je systém vystavený riziku, že sa neoprávnený užívateľ dostane k počítaču administrátora (v prípade ak si nezamkne obrazovku). Odporúčaná hodnota ClientAliveInterval je 300 (5 minút) a hodnota pre parameter ClientAliveCountMax je 0. Pri takomto nastavení sa pripojenie ukončí po 5 minútach neaktivity bez upozornenia.

Realizácia: Pridať nasledujúci riadok do súboru `/etc/ssh/sshd_config`:

```
ClientAliveInterval 300
ClientAliveCountMax 0
```

5.4 Nastavenie logovania

Logovanie by malo byť nastavené spôsobom aby neumožňovalo únik dát a zároveň boli logy ukladané na vzdialenom servery. V prípade napadnutia systému je potom možné skontrolovať a vyhodnotiť záznamy z logov bez obavy že boli skompromitované.

5.4.1 Povolenie rsyslog

Popis: Po nainštalovaní programu rsyslog musí byť aktivovaný.

Zdôvodnenie: Ak rsyslog nie je aktívny systémové hlásenia nemusia byť zapísané do log súborov.

Realizácia: Spustením nasledujúceho príkazu sa rsyslog aktivuje:

```
# chkconfig syslog on
```

5.4.2 Nastavenie práv vytvorených logov

Popis: rsyslog vytvára logovacie súbory ak neexistujú na systéme. Toto nastavenie určí s akými právami budú vytvorené.

Zdôvodnenie: Je dôležité aby boli logovacie súbory vytvorené so správnymi právami a ochránili tak citlivé dáta.

Realizácia: Upraviť súbor `/etc/rsyslog.conf` a nastaviť parameter $FileCreateMode na práva 0640 alebo prísnejšie:

```
$FileCreateMode 0640
```

5.4.3 Posielanie logov na vzdialený server

Popis: Program rsyslog podporuje možnosť posielania logov, ktoré zozbiera na vzdialený server, alebo prijímať správy zo vzdialených serverov a zjednodušiť tak administráciu logov.

Zdôvodnenie: Uložiť logy na vzdialenom servery umožňuje zachovať integritu logov, v prípade že sa útočníkovi podarí dostať na systém a bude sa snažiť upraviť alebo zmazať stopy po jeho aktivite.

Realizácia: Upraviť súbor `/etc/rsyslog.conf` a pridať nasledujúci riadok(kde `vzdialeny.server.com` je potrebné nahradiť funkčným vzdialeným severom):

```
*.* @@vzdialeny.server.com
```

Po pridaní je nutné reštartovať rsyslogd nasledujúcim príkazom:

```
# pkill -HUP rsyslogd
```

5.5 Nastavenie auditu

Audit systému pomocou audit démona, umožňuje administrátorovi monitorovanie vopred zadefinovaných situácii, ako napríklad neoprávnené prihlásenie alebo úpravu dát na systéme. Všetky udalosti sa zapisujú do `/var/log/audit/audit.log`. Za normálnych okolností zapísané udalosti v logu zaberajú relatívne málo miesta na disku. V prípade zadefinovania mnohých situácii alebo pri špecifických udalostiach je lepšie prideliť viac priestoru na disku pre auditd démona. Je preto dôležité prispôsobiť veľkosť logov a počet rotácii na prostredie, v ktorom sa budú tieto pravidlá aplikovať. Štandardne je maximálna veľkosť logu nastavená na 5MB a počet rotácii na 4. Každá staršia verzia bude zmazaná.

Pre 64bitové systémy, ktoré majú nastavený parameter arch v pravidle, je nutné zadefinovať pravidlá dva krát: prvé pre 64 bitové systémy a druhé pre 32 bitové systémy. V prípade 32 bitového systému stačí pravidlo len jedno. V tejto práci sú popísané pravidlá pre 64 bitové systémy.

Niektoré pravidlá majú v sebe identifikátor (prepínač -k), ktorý slúži k jednoduchšiemu vyhľadávaniu záznamov alebo ich filtrovaniu. Identifikátor je možné ľubovoľne zmeniť podľa potreby.

Po každej úprave súboru `/etc/audit/audit.rules`, je nutné reštartovať auditd démon príkazom:

```
# service auditd reload
```

5.5.1 Nastavenie maximálnej veľkosti audit logov

Popis: Nastavenie maximálnej veľkosti logu, v momente keď dosiahne nastavenú veľkosť bude log uložený a nahradený novým.

Zdôvodnenie: Je dôležité aby maximálna veľkosť logu odpovedala potrebám systému, aby nedochádzalo k stratám dát pre audit.

Realizácia: Nastavením nasledujúceho parametru v súbore`/etc/audit/auditd.conf`:

```
max_log_file = <MB>
```

5.5.2 Správanie systému pri zaplnení log súborov

Popis: V prípade zaplnenia audit logov, auditd démom môže okamžite zastaviť systém.

Zdôvodnenie: Toto nastavenie je možné použiť v systémoch, kde sa kladie veľký dôraz na bezpečnosť aj na úkor dostupnosti.

Realizácia: Nastavením nasledujúcich parametrov v súbore `/etc/audit/auditd.conf`:

```
space_left_action = email
action_mail_acct = root
admin_space_left_action = halt
```

5.5.3 Zastavenie automatického zmazania logov

Popis: Parameter max_log_file_action určuje ako zaobchádzať s audit logom, ak dosiahne maximálnu veľkosť. Hodnota keep_logs plný log zamení za starý, ale nikdy ich nezmaže.

Zdôvodnenie: V systémoch kde sa kladie dôraz na bezpečnosť, výhoda mať všetky staršie logy prevyšuje cenu miesta na disku.

Realizácia: Nastavením nasledujúceho parametru v súbore`/etc/audit/auditd.conf`:

```
max_log_file_action = keep_logs
```

5.5.4 Monitorovanie udalostí, ktoré upravujú dátum a čas

Popis: Zaznamenávanie udalostí, ktoré súvisia so zmenou času a dátumu na systéme. Parametre v tejto sekcii kontrolujú či `adjtimex` (úprava hodín jadra), `settimeofday` (nastavenie času použitím štruktúry časových pásiem), `stime` (nastavenie sekúnd od 1/1/1970), alebo `clock_settime` (nastavenie času rôznych interných hodín a časovačov)

systémové volania boli spustené a ak áno zapísať to do `/var/log/audit.log` po ukončení. Identifikátor takýchto aktivít je „time-change“.

Zdôvodnenie: Neplánované zmeny času na systéme môžu byť znakom neoprávnených aktivít.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time-change
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k time-change
-a always,exit -F arch=b64 -S clock_settime -k time-change
-a always,exit -F arch=b32 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change
```

Obrázok 11: Parametre na monitorovanie udalostí, ktoré upravujú dátum a čas [zdroj: Autor]

5.5.5 Monitorovanie súborov, spojených s užívateľmi, skupinami a heslami

Popis: Zaznamenávanie udalostí, ktoré modifikujú súbory spojené s užívateľmi, skupinami, heslami užívateľov a skupín. Parametre v tejto sekcii zachytia, ak bol súbor otvorený na zápis alebo nastal pokus o zmenu práv. Identifikátor takýchto aktivít je „identity“.

Zdôvodnenie: Neplánované zmeny v týchto súboroch môžu naznačovať snahu o zakrytie stôp po neoprávnených aktivitách.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
```

5.5.6 Monitorovanie udalostí siete

Popis: Zaznamenávanie udalostí, ktoré modifikujú súbory spojené so sieťou a systémové volania `sethostname` a `setdomainname`.

Zdôvodnenie: Monitorovanie `sethostname` a `setdomainname` odhalí potenciálne zmeny mena systému a domény. Takéto zmeny môžu potenciálne pokaziť bezpečnostné parametre, ktoré sú nastavené na základe týchto hodnôt. Zmena v súbore `/etc/hosts` môže naznačovať snahu útočníka nalákať užívateľov na pripojenie k inému systému ako chcel. Zmenou v súboroch `/etc/issue` a `/etc/issue.net` môže útočník vložiť nepravé informácie a tým nalákať užívateľa k poskytnutiu informácii. Zmena v niektorom zo súborov v adresári `/etc/sysconfig/network` môže viesť k nedostupnosti alebo kompromitácii systému. Identifikátor takýchto aktivít je „system-locale“.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-a always,exit -F arch=b64 -S sethostname -S setdomainname -k system-locale
-a always,exit -F arch=b32 -S sethostname -S setdomainname -k system-locale
-w /etc/issue -p wa -k system-locale
-w /etc/issue.net -p wa -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/sysconfig/network -p wa -k system-locale
```

5.5.7 Monitorovanie udalostí prihlásenia a odhlásenia

Popis: Zaznamenávanie modifikácii súborov, ktoré zaznamenávajú udalosti prihlásenia a odhlásenia užívateľov na systém. Súbor `/var/log/lastlog` zaznamenáva posledné úspešné prihlásenia užívateľov. Súbor `/var/log/faillog` zaznamenáva neúspešné prihlásenia užívateľov. Identifikátor takýchto aktivít je „logins“.

Zdôvodnenie: Monitorovanie prihlasovania užívateľov, môže umožniť administrátorovi jednoduchšiu identifikáciu útočníka.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-w /var/log/lastlog -p wa -k logins
-w /var/log/faillog -p wa -k logins
```

5.5.8 Monitorovanie zmien práv a vlastníkov súborov

Popis: Parametre v tejto sekcii sledujú systémové volania ktoré menia práva a atribúty súborov. Systémové volania `chmod`, `fchmod` a `fchmodat` menia práva spojené zo súbormi. Systémové volania `chown`, `fchownm`, `fchownat` a `lchown` menia atribúty spojené s vlastníkom alebo skupinou vlastníkov súborov. Systémové volania `setxattr`, `lsetxattr`,

`fsetxattr`, `removexattr`, `lremovexattr`, `fremovexattr` menia rozšírené atribúty súborov(napr., symbolický odkaz, i-uzol). Vo všetkých prípadoch bude záznam vytvorený iba ak takúto aktivitu prevádza nesystémový užívateľ (auid >= 500). Identifikátor takýchto aktivít je „perm_mod“.

Zdôvodnenie: Monitorovanie zmien atribútov súborov, môže administrátora upozorniť na neautorizované aktivity.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -F auid>=500 -F auid!=4294967295
-k perm_mod

-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -F auid>=500 -F auid!=4294967295
-k perm_mod

-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -F auid>=500 -F
auid!=4294967295 -k perm_mod

-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -F auid>=500 -F
auid!=4294967295 -k perm_mod

-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S
lremovexattr -S fremovexattr -F auid>=500 -F auid!=4294967295 -k perm_mod

-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S
lremovexattr -S fremovexattr -F auid>=500 -F auid!=4294967295 -k perm_mod
```

Obrázok 12: Parametre na monitorovanie zmien práv a vlastníkov súborov [zdroj: Autor]

5.5.9 Monitorovanie neúspešných pokusov o prístup k súborom

Popis: Parametre v tejto sekcii monitorujú systémové volania spojené s vytváraním (`creat`), otváraním (`open`, `openat`) a osekaním alebo skrátením (`truncate`, `ftruncate`) súborov. Záznam bude vytvorený iba ak sa jedná o neprivilegovaného užívateľa (auid >= 500), nie je to udalosť vytvorené démonom (auid=4294967295) a ak systémové volanie vráti hodnotu EACCESS (nedostatočné práva na prístup k súboru) alebo EPERM (iná chyba spojená so systémovým volaním). Identifikátor takýchto aktivít je „access“.

Zdôvodnenie: Neúspešný pokus o otvorenie, vytvorenie alebo osekať/skrátiť súbor môže byť indikátorom snahy o neautorizovaný prístup na systém.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-
EACCES -F auid>=500 -F auid!=4294967295 -k access

-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-
EACCES -F auid>=500 -F auid!=4294967295 -k access

-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-
EPERM -F auid>=500 -F auid!=4294967295 -k access

-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-
EPERM -F auid>=500 -F auid!=4294967295 -k access
```

Obrázok 13: Parametre na monitorovanie neúspešných pokusov o prístup k súborom [zdroj: Autor]

5.5.10 Monitorovanie vymazávania súborov

Popis: Parametre v tejto sekcii monitorujú systémové volania `unlink`(zmazanie súborov), `unlinkat` (zmazanie atribútu súboru), `rename` (premenovanie súboru), `renameat` (premenovanie atribútu súboru). Identifikátor takýchto aktivít je „delete“.

Zdôvodnenie: Monitorovanie aktivít tohto typu môže administrátorovi poskytnúť dôkaz v prípade neoprávneného zmazania súborov alebo ich atribútov. Audit bude kontrolovať všetky tieto aktivity, ale administrátor bude chcieť pravdepodobne sledovať hlavne špecifické súbory v jeho prostredí.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-a always,exit -F arch=b64 -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F
auid!=4294967295 -k delete

-a always,exit -F arch=b32 -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F
auid!=4294967295 -k delete
```

Obrázok 14: Parametre na monitorovanie vymazávania súborov[zdroj: Autor]

5.5.11 Monitorovanie konfiguračných súborov suda

Popis: Ak je systém správne nastavený a administrátori sa musia prihlasovať na svojich užívateľov a až potom získať oprávnenia pomocou príkazu `sudo`, odporúča sa sledovať zmeny v sudoers súboroch. Identifikátor takýchto aktivít je „scope“.

Zdôvodnenie: Zmeny v súbore `/etc/sudoers` môžu indikovať snahu o zmenu oprávnení administrátorov.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúcich riadkov:

```
-w /etc/sudoers -p wa -k scope
```

```
-w /etc/sudoers.d -p wa -k scope
```

5.5.12 **Nastavenie neupravitelnosti auditu**

Popis: Toto nastavenie zaisťuje zamedzenie úpravy pravidiel auditu. Každá zmena vykonaná pomocou príkazu `auditctl` sa prejaví až po najbližšom reštartovaní systému.

Zdôvodnenie: Zabraňuje sa tým útočníkovi upraviť pravidlá auditu pred neoprávnenými aktivitami a potom ich vrátiť do pôvodného stavu. S veľkou pravdepodobnosťou si administrátor alebo ostatní užívatelia všimnú reštart systému a tým sa môže zmariť pokus útočníka.

Realizácia: Upraviť súbor `/etc/audit/audit.rules` pridaním nasledujúceho riadku na jeho koniec:

```
-e 2
```

ZÁVER

Únik akejkoľvek citlivej informácie o užívateľoch daného serveru spôsobuje stratu dôveryhodnosti k stranám, ktoré servery prevádzkujú a následne aj finančné straty. Akýkoľvek užívateľ platformy je potenciálnym prvkom útočníka do systému. Pre správne vyhodnotenie rizík spojených so správou užívateľov je nevyhnutné aby administrátor rozumel všetkým zahrnutým komponentom. Primárny prehľad týchto komponentov nájdev teoretickej časti a pri potrebe rozsiahlejšieho rozboru využiť informácie z čerpaných zdrojov.

Teoretická časť problematiky sa zameriava na správu hesiel, spôsoby prihlásenia užívateľov, prideľovanie a správu oprávnení užívateľovi a zaznamenávanie jeho činností. V práci je detailne spracovaný princíp fungovania hesiel a teoretický rozbor protokolu SSH. Keďže je protokol SSH najpoužívanejším typom prihlásenia, útočníci v ňom často vyhľadávajú chyby. Správne nastavenie politiky na správu hesiel a adekvátna konfigurácia protokolu SSH je nevyhnutnosťou, čím administrátor znižuje šancu na prienik. Útočník, ktorý sa nedostane do systému má len veľmi obmedzené možnosti na vykonanie akejkoľvek aktivity, ktorá môže systém poškodiť.

V praktickej časti boli vysvetlené jednotlivé parametre a nastavenia vedúce k lepšiemu zabezpečeniu systému v rámci správy užívateľov. Tieto nastavenia obsahujú popis, odôvodnenie a samotnú implementáciu daných parametrov, ako napr. nastavenie kritérií na vytváranie hesiel, nastavenie parametrov na zabezpečenie SSH servera MaxAuthTries, IgnoreRhosts, PermitRootLogin, PermitEmptyPassword a iné. Ich aplikáciou sa nielen zintenzívňuje stupeň ochrany, ale pomáhajú i predchádzať najpoužívanejším typom útokov tretej strany a výrazne komplikujú útočníkovi efektívne získať kontrolu nad systémom. Ak by útoky z tretej strany aj po aplikácii takýchto parametrov boli naďalej prítomné, prípadne niektoré z útokov boli úspešné, je žiadané rozšíriť ochranu užívateľov využitým ďalších prístupných, ale komplexnejších zabezpečovacích nástrojov, ako napr. program apparmor, fail2ban, alebo rozšíriť zabezpečenie pomocou modulu SELinux. Následne boli vymenované iparametre na nastavenie logovania a auditu. Napriek tomu, že nezvyšujú zabezpečenie systému, významne pomáhajú k riešeniu potenciálnych útokov a neoprávnených aktivít.

Zároveň treba zdôrazniť, že ochranné prvky sa neustále vyvíjajú a zintenzívňujú. Parametre implementované v tejto práci môžu byť tak onedlho

nedostačujúce na primárnu ochranu. Je potrebné myslieť pri implikovaní serverov nielen na prvotné zavedenie takýchto ochranných parametrov, ale taktiež na ich neustálu aktualizáciu po celú dobu životnosti servera.

ZOZNAM POUŽITEJ LITERATÚRY

[1] Linux: dokumentační projekt. 4., aktualiz. vyd. Přeložil Lubomír PTÁČEK. Brno: Computer Press, 2007. ISBN 978-80-251-1525-1.

[2] KRČMÁŘ, Petr. Linux: tipy a triky pro bezpečnost. Brno: Grada Publishing, 2004. ISBN 80-247-0812-4.

[3] What is Linux?. Linux Training Academy [online]. [cit. 2018-04-03]. Dostupné z: https://www.linuxtrainingacademy.com/what-is-linux/

[4] Salted Password Hashing - Doing it Right. CrackStation [online]. [cit. 2018-04-03]. Dostupné z: https://crackstation.net/hashing-security.htm

[5] SURENDRA, Anne. How login process work in Linux ?. The Linux Juggernaut [online]. [cit. 2018-04-03]. Dostupné z: https://www.linuxnix.com/how-login-process-work-in-linux/

[6] SSH – bezpečné používání vzdáleného počítače a kopírování dat. DSL.cz [online]. [cit. 2018-04-03]. Dostupné z: http://www.dsl.cz/jak-na-to/jak-na-ssh

[7] SSH PROTOCOL. SSH.COM - SSH Communications Security [online]. 29. 8. 2017 [cit. 2018-04-03]. Dostupné z: https://www.ssh.com/ssh/protocol/

[8] ELLINGWOOD, Justin. SSH Essentials: Working with SSH Servers, Clients, and Keys. DigitalOcean: Cloud Computing, Simplicity at Scale [online]. 16. 4. 2014 [cit. 2018-04-03]. Dostupné z: https://www.digitalocean.com/community/tutorials/ssh-essentials-working-with-ssh-servers-clients-and-keys#server-side-configuration-options

[9] SFTP – SSH SECURE FILE TRANSFER PROTOCOL. SSH.COM - SSH Communications Security [online]. 10. 10. 2017 [cit. 2018-04-03]. Dostupné z: https://www.ssh.com/ssh/sftp/

[10] WALLEN, Jack. Linux 101: Introduction to sudo. Linux.com | News for the open source professional [online]. 12. 05. 2010 [cit. 2018-04-03]. Dostupné z: https://www.linux.com/learn/linux-101-introduction-sudo

[11] KEREKI, Federico. Sudo, or not sudo: that is the question. Linux.com | News for the open source professional [online]. 07. 02. 2008 [cit. 2018-04-03]. Dostupné z: https://www.linux.com/news/sudo-or-not-sudo-question

[12] Sudoers Manual. Sudo Main Page [online]. [cit. 2018-04-03]. Dostupné z: https://www.sudo.ws/man/1.8.22/sudoers.man.html

[13] ŠÍPOŠ, Juraj. Práva sú výsadou Linuxu. Linux E X P R E S [online]. 29. 05. 2006 [cit. 2018-04-03]. Dostupné z: https://www.linuxexpres.cz/praxe/prava-su-vysadou-linuxu

[14] Protokol Telnet. Úvod | Webhosting BANAN = webové stránky zdarma a hosting [online]. [cit. 2018-04-03]. Dostupné z: https://www.banan.cz/serialy/sitove-protokoly/Sitove-protokoly-XIX-cast-Protokol-Telnet

[15] KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS. 5. aktualiz. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2236-5.

[16] Rsyslog.conf(5) - Linux man page. Linux Documentation [online]. [cit. 2018-04-03]. Dostupné z: https://linux.die.net/man/5/rsyslog.conf

[17] 20 Linux Log Files that are Located under /var/log Directory. The Geek Stuff [online]. 01. 08. 2011 [cit. 2018-04-03]. Dostupné z: https://www.thegeekstuff.com/2011/08/linux-var-log-files/

[18] PAM - správa autentizačních mechanismů. Root.cz - informace nejen ze světa Linuxu [online]. 19. 09. 2000 [cit. 2018-04-03]. Dostupné z: https://www.root.cz/clanky/pam-sprava-autentizacnich-mechanismu/

[19] KEDER, Daniel. Autentizácia v Linuxe pomocou PAM. AbcLinuxu.cz - Linux na stříbrném podnose [online]. 30. 04. 2008 [cit. 2018-04-03]. Dostupné z: http://www.abclinuxu.cz/clanky/bezpecnost/autentizacia-v-linuxe-pomocou-pam

[20] Proč byste měli chtít SELinux?. Root.cz - informace nejen ze světa Linuxu [online]. 17. 12. 2007 [cit. 2018-04-03]. Dostupné z: https://www.root.cz/clanky/proc-byste-meli-chtit-selinux/

[21] 7.6. UNDERSTANDING AUDIT LOG FILES. Red Hat Customer Portal [online]. [cit. 2018-04-03]. Dostupné z: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security_guide/sec-understanding_audit_log_files

[22] Understanding /etc/shadow file. NixCraft - Linux Tips, Hacks, Tutorials, And Ideas In Blog [online]. 02.08. 2018 [cit. 2018-04-29]. Dostupné z: https://www.cyberciti.biz/faq/understanding-etcshadow-file/

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

OS	Operačný Systém
LIDS	Význam druhé zkratky.
IBM	Význam třetí zkratky.
MS-DOS	Microsoft Disk Operating System
Telnet	Teletype Network
SSH	Secure Shell
SFTP	Secure File Transfer Protocol
ASCII	American Standard Code for Information
SCP	Secure Copy
PAM	Pluggable Authentification Module
BSD	Berkeley Software Distribution
FTP	File Transfer Protocol
RAM	Random Access Memory
SUID	Set User ID
GUID	Set Group ID

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK