EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

jihomoravský kraj

OPERAČNÍ SYSTÉMY

Pokročilá práce s protokoly v Linuxu

Metodický list

Autor: Ing. Marek Kocan, Metodik: Ing. Roman Koláčný

Recenzent: Mgr. Jiří Činčura

Rok vydání: 2023

Obsah

Cíle

Studenti se seznámí s teoretickými základy v oblasti protokolů u operačních systémů linuxového typu a osvojí si dovednosti pro pokročilou práci s protokoly, a to na základě konkrétních příkladů předvedených vyučujícím i na základě samostatně vyzkoušených úkolů.

Student bude schopen vlastními slovy vysvětlit co jsou protokoly, co je základem protokolů v Linuxu, kde se protokoly v Linuxu nachází a jak v základu linuxové protokoly fungují. Student zvládne vlastními slovy vyjmenovat a popsat další příkazy operačního systému Linux pro práci s textovými soubory obecně. V dlouhodobém horizontu bude student schopen samostatně využít nové znalosti a dovednosti pro zjednodušení správy operačního systému.

Dovednosti

Student bude schopen vypsat obsah hlavních protokolů operačního systému, zajistit průběžné vypisování protokolů na terminálu a vyhledávat podle obsahu, druhu i závažnosti. Dále student zvládne pracovat ve více částech terminálu současně, provést instalaci lokálního poštovního serveru a demonstrovat zaslání alertu v návaznosti na výskyt záznamu v protokolu V dlouhodobém horizontu bude student schopen aplikovat získané dovednosti pro efektivní práci s protokoly operačního systému.

Pracovní prostředí

Výuku lze realizovat v prostředí:

- Cylab JCEKB, operační systém Debian (linuxový server)

Pro práci postačí standardní nástroje, některé další budou v rámci průběhu scénáře nainstalovány.

1 Pokročilá práce s protokoly v Linuxu – teoretický základ

Obdobně jako každý moderní operační systém i operační systémy linuxového typu pracují s tzv. protokoly událostí (logy). Záznamy v protokolech mohou vytvářet jak procesy jádra OS, tak i démoni (serverové služby) a aplikační programy.

1.1 Protokoly v Linuxu

Takřka vše, s čím se lze v rámci souborových systémů v Linuxu setkat, jsou soubory. Jinak tomu není ani v případě protokolů, drtivá většina z nich je uložená v souborech – za výjimku lze považovat speciální datovou strukturu pro protokoly jádra, tzv. *kruhový buffer jádra* (kernel ring buffer). Tato struktura umožňuje zaznamenat a uchovat informace o událostech ještě před startem procesů, které mají na starosti přijímání zpráv a jejich ukládání do souborů s protokoly (zpravidla jde syslog/syslog-ng démon; další informace včetně formátů lze nalézt mj. v RFC 5424 a RFC 5427). Některé protokoly jsou přímo čitelné prostým vypsáním či zobrazením, některé pouze pomocí speciálních příkazů. Vedle základních systémových nástrojů lze využít i pokročilé aplikace či centralizovaná serverová řešení.

Do souborů s protokoly jsou ukládány podle nastavení informace od kritických až po zcela běžné informativní (nejčastěji je využívána škála 0 až 7, *výstraha/stav nouze* až *ladění*). Jde o tzv. *log level*. Zejména v případě logů zpracovaných pomocí syslogu a kompatibilních služeb jsou jednotlivé záznamy kategorizovány do tzv. *druhu zprávy*. Aby se předešlo nepřehlednosti a zaplnění úložného prostoru, jsou protokoly tzv. *rotovány* (služba logrotate) – za definovaných podmínek jsou zkopírovány do archivních souborů, případně také zkomprimovány (například z `kern.log` tak vznikne `kern.log.1`, `kern.log.2.gz` apod.).

Anglický pojem *log* je v kontextu operačního systému linuxového typu využíván jak pro označení jednotlivého záznamu v souboru s protokolem, tak i pro samotný soubor protokolu. Anglickému pojmu *log level* odpovídá český *úroveň protokolu*, využívá se ale také *závažnost* (*severita*) a je možné se setkat i s pojmy *úroveň priority* či *úroveň závažnosti*. *Druh zprávy* je využíván jako česká alternativa pojmu *facility*. V českých zdrojích panuje značná nejednoznačnost v pojmosloví, proto je vhodné, aby vyučující na tento fakt upozornil.

1.1.1 Umístění v rámci adresářové struktury

V případě Linuxu jsou obvykle protokoly umístěné v adresáři `/var/log` včetně podadresářů dle jednotlivých služeb/programů. Nicméně je důležité zdůraznit, že se mohou nacházet i v jiných částech stromové struktury, případně mohou být dle nastavení ukládány i do dalších struktur či zasílány do externích služeb. Nicméně lze oprávněně předpokládat, že bez úprav konfigurací jsou všechny systémové protokoly umístěné právě v adresáři `/var/log`.

1.1.2 Hlavní protokoly

Následující seznam není vyčerpávajícím přehledem všech možných protokolů na operačních systémech linuxového typu, vztahuje se ke standardní instalaci výukového prostředí. Vedle níže uvedených protokolů mohou vznikat i další v návaznosti na doinstalování programů a služeb.

Vše ve `/var/log`, T označuje textový soubor, P vyžaduje použití příkazu.

- `syslog` – hlavní systémový protokol se zprávami z mnoha různých zdrojů podle nastavení (T)
- `messages` – hlavní systémový protokol o aktivitách v systému včetně služeb (T)
- `alternatives.log` – protokol pro update-alternatives (spravuje sym-linky pro defaultní příkazy) (T)
- `apt/` – adresář pro protokoly související s instalačním nástrojem apt
- `auth.log` – protokol související s autorizačními požadavky a přihlášeními (T)
- `boot.log` – protokol související se startem operačního systému (T)
- `btmp` – protokol s informacemi o chybných přihlášeních k systému (P: `last -f /var/log/btmp`, `lastb`)
- `cups/` – adresář pro protokoly související s tiskem
- `daemon.log` – protokol se zprávami od démonů (serverových služeb) (T)
- `debug` – protokol s debug zprávami od systému (T)
- `dpkg.log` – protokol se zprávami od správce balíčků dpkg (T)
- `faillog` – protokol s informacemi o chybných přihlášeních k systémů (pro uživatele) (P: `faillog`)
- `kern.log` – protokol se zprávami souvisejícími s jádrem operačního systému (T)
- `lastlog` – protokol s informacemi o přihlášeních k systému pro uživatele (P: `lastlog`)
- `user.log` – protokol se zprávami na uživatelské úrovni (T)
- `wtmp` – protokol s informacemi o přihlášeních k systému (P: `last, last -f /var/log/wtmp`)
- `xrdp.log` – protokol se zprávami pro RDP (vzdálenou plochu) (T)
- `xrdp-sesman.log` – protokol správce XRDP spojení (T)

2 Pokročilá práce s protokoly v Linuxu – praktické ukázky a cvičení

V této části vyučující představí jednotlivé příklady pokročilé práce s protokoly. Na tyto ukázky bude průběžně navazovat samostatná práce studentů (kontrolní body).

2.1 Pracovní prostředí

Výuku lze realizovat v prostředí Cylab JCEKB – Linuxový server (Debian).

2.2 Pokročilé nástroje pro práci s logy

2.2.1 Užitečné příkazy

Příkaz `wc` umožňuje spočítat počet řádků v textovém souboru, což lze v případě protokolů využít pro zjištění počtu záznamů. Následující příklad ukazuje, jak lze zjistit počet záznamů v protokolu `syslog`.

```
$ sudo wc -l /var/log/syslog
```

Výstup (dle skutečného obsahu):

```
$ 4739 /var/log/syslog
```

Příkaz `head` umožňuje vypsat n prvních záznamů textového souboru, v případě protokolů tedy nejstarších. Následující příklad ukazuje, jak zobrazit 5 nejstarších záznamů z protokolu `syslog`.

```
$ sudo head -n 5 /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 20 10:36:46 debian systemd[1]: Starting Network Manager Script Dispatcher Service...
  Apr 20 10:36:46 debian rsyslogd:  [origin software="rsyslogd" swVersion="8.1901.0" ...
  Apr 20 10:36:46 debian dbus-daemon[385]: [system] Successfully activated service ...
  Apr 20 10:36:46 debian systemd[1]: Started Network Manager Script Dispatcher Service.
  Apr 20 10:36:46 debian systemd[1]: Started GNOME Display Manager.
```

Protikladem příkazu `head` je `tail`, který umožňuje vypsat n posledních záznamů textového souboru, v případě protokolů tedy nejnovějších. Následující příklad ukazuje, jak zobrazit 5 nejnovějších záznamů z protokolu `syslog`.

```
$ sudo tail -n 5 /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 29 16:45:45 debian systemd[1]: Started Network Manager Script Dispatcher Service.
  Apr 29 16:45:45 debian nm-dispatcher: req:1 'dhcp4-change' [ens18]: new ...
  Apr 29 16:45:45 debian nm-dispatcher: req:1 'dhcp4-change' [ens18]: start ...
  Apr 29 16:45:45 debian dhclient[492]: bound to 192.168.100.252 -- renewal in 240 seconds.
  Apr 29 16:45:55 debian systemd[1]: NetworkManager-dispatcher.service: Succeeded. ...
```

Příkaz `nl` umožňuje očíslovat řádky v textovém souboru, což lze v případě protokolů využít pro snadnější orientaci či odkazy. Následující příklad ukazuje, jak lze očíslovat řádky v protokolu `syslog` (nejde o fyzické očíslování v souboru, ale ve výstupu).

```
$ sudo nl /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ 4758 Apr 22 11:22:37 debian nm-dispatcher: req:1 'dhcp4-change' [ens18]: ...
  4759 Apr 22 11:22:38 debian nm-dispatcher: req:1 'dhcp4-change' [ens18]: start ...
  4760 Apr 22 11:22:38 debian dhclient[492]: bound to 192.168.100.252 -- renewal in 245 s.
```

Kontrolní bod

Studenti si u příkazů `head` a `tail` vyzkouší, co se stane v případě vynechání parametru `n`.

2.2.2 Využití příkazu `grep`

Pro práci s protokoly je užitečný příkaz `grep`, který zajišťuje vyhledávání. Následující příklad ukazuje, jak pomocí příkazu `grep` vyhledat v protokolu `auth.log` informace související s uživatelem sysadmin. V příkladu je uvedený tento uživatel s velkým počátečním písmenem, aby bylo možné demonstrovat funkčnost přepínače `-i`, který zajistí vyhledávání bez ohledu na velikost písma.

```
$ sudo grep -i Sysadmin /var/log/auth.log
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 26 14:29:15 debian sudo: ... session opened for user root by sysadmin(uid=0)
  Apr 26 14:29:16 debian sudo: sysadmin : TTY=pts/1 ; PWD=/home/sysadmin ; USER=root ...
```

Vstupní parametry lze uzavřít do uvozovek a hledat tak řetězec jako celek, viz následující příklad, který demonstruje vyhledání textu *port 67* (DHCP) v protokolu `syslog`.

```
$ sudo grep -i "port 67" /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 26 12:37:36 ... DHCPREQUEST for 192.168.100.252 on ens18 to 192.168.100.1 port 67
  Apr 26 12:42:13 ... DHCPREQUEST for 192.168.100.252 on ens18 to 192.168.100.1 port 67
  Apr 26 12:47:09 ... DHCPREQUEST for 192.168.100.252 on ens18 to 192.168.100.1 port 67
  Apr 26 12:51:45 ... DHCPREQUEST for 192.168.100.252 on ens18 to 192.168.100.1 port 67
```

Během práce s protokoly lze využívat všechny možnosti příkazu grep, následující příklad ukazuje vyloučení záznamů protokolu `syslog`, které obsahují text `NetworkManager`.

```
$ grep -v NetworkManager /var/log/syslog
```

Další možností je využití regulárních výrazů, následující příklad ukazuje vypsání všech záznamů z protokolu `syslog`, které nějak souvisí s privátním IP rozsahem 192.168.0.0/32.

```
$ sudo grep -E '192\.168\.[0-9]{1,3}\.[0-9]{1,3}' /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 26 12:39:36 ... [1651590207.8475] dhcp4 (ens18):   address 192.168.100.252
  Apr 26 12:39:36 ... [1651590207.8476] dhcp4 (ens18):   gateway 192.168.100.1
  Apr 26 12:39:36 ... [1651590207.8476] dhcp4 (ens18):   nameserver '192.168.100.1'
```

Následující příklad demonstruje využití přepínačů `-B` a `-A`, které zajistí vypsání příslušného počtu řádků před a za nalezeným výskytem, v uvedeném příkladu 5 před a 6 za (tyto hodnoty jsou zjištěné experimentálně tak, aby výstup dával smysl).

```
$ sudo grep -B 5 -A 6 "Power-on or device reset occurred" /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 30 10:47:44 ... scsi 2:0:0:0: Direct-Access     QEMU     QEMU HARDDISK    2.5+ PQ: 0
  Apr 30 10:47:44 ... scsi target2:0:0: tagged command queuing enabled, command queue ...
  Apr 30 10:47:44 ... scsi target2:0:0: Beginning Domain Validation
  Apr 30 10:47:44 ... scsi target2:0:0: Domain Validation skipping write tests
  Apr 30 10:47:44 ... scsi target2:0:0: Ending Domain Validation
  Apr 30 10:47:44 ... sd 2:0:0:0: Power-on or device reset occurred
  Apr 30 10:47:44 ... sd 2:0:0:0: [sda] 209715200 512-byte logical blocks:(107 GB/100 GiB)
  Apr 30 10:47:44 ... sd 2:0:0:0: [sda] Write Protect is off
```

```
  Apr 30 10:47:44 ... sd 2:0:0:0: [sda] Mode Sense: 63 00 00 08
  Apr 30 10:47:44 ... sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, ...
  Apr 30 10:47:44 ... sda: sda1 sda2 < sda5 >
  Apr 30 10:47:44 ... sd 2:0:0:0: [sda] Attached SCSI disk
```

Kontrolní bod

Studenti si vyzkouší pomocí příkazu `grep` najít v protokolu `auth.log` informace o chybném přihlášení.

2.2.3 Využití příkazu `awk`

Práce s protokoly je v drtivé většině případů o analýze textu – vhodným příkazem je `awk`, který umožňuje podrobný rozbor.

Následující příklad ukazuje využití příkazu `awk` pro nalezení záznamů v protokolu `auth.log`, které poukazují na špatné zadání hesla. Současně ale vypíše pouze 9. položku záznamu, což odpovídá názvu uživatele. Oddělení je definováno tzv. bílým znakem.

```
$ sudo awk '/Failed password/ { print $9 }' /var/log/auth.log
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ sysadmin
  sysadmin
  sysadmin
```

Obdobou předchozího příkladu je vyhledání a zobrazení „položek“ v protokolu `daemon.log` souvisejících s požadavkem na DHCP server (zobrazeny jsou pouze položky pro datum a čas, rozhraní, přiřazenou IP adresu a DHCP server (*vyučující může upozornit na to, že pořadí neodpovídá původnímu pořadí položek*).

```
$ sudo awk '/DHCPREQUEST/ { print $1,$2,$3,$10,$8,$12 }' /var/log/daemon.log
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 30 12:16:44 ens18 192.168.100.252 192.168.100.1
  Apr 30 12:20:32 ens18 192.168.100.252 192.168.100.1
  Apr 30 12:24:21 ens18 192.168.100.252 192.168.100.1
```

Následující příklad vyhledá a vypíše takové záznamy z protokolu `syslog`, pro které platí, že obsahují hodnotu network nebo device. Konstrukce `tolower` zajišťuje nezohledňování velikosti písma, které není v případě příkazu `awk` tak intuitivní jako například u příkazu `grep`.

```
$ sudo awk 'tolower($0) ~ /network|device/' /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ Apr 30 13:15:21 ... Starting Network Manager Script ...
  Apr 30 18:19:38 ... device (ens18): carrier: link connected
```

Kontrolní bod

Studenti vypíšou z protokolu `auth.log` pouze datumové položky, a to v pořadí datum a měsíc.

2.2.4 Průběžné tipy a triky

V závislosti na nastaveném rotování jsou soubory s protokoly archivovány včetně zazipování. V některých případech není nutné pro jejich zpracování soubory dekomprimovat (a využívat například techniky s pipe), ale použít alternativu umožňující přímou práci. K dispozici jsou především příkazy/skripty `zcat` pro zobrazení a `zgrep` pro vyhledávání. Způsoby použití demonstrují následující příklady.

```
$ sudo zcat /var/log/auth.log.2.gz
```

```
$ sudo zgrep -i sysadmin /var/log/auth.log.2.gz
```

2.3 Filtrování s ohledem na druh zprávy a závažnost

2.3.1 Příkaz dmesg

Příkaz `dmesg` umožňuje omezit výpis podle druhu zprávy a podle závažnosti jednotlivých záznamů uložených v kruhovém bufferu jádra. Druh zprávy (facility) může nabývat hodnoty:

- kern – kernel messages
- user – random user-level messages
- mail – mail system
- daemon – system daemons
- auth – security/authorization messages
- syslog – messages generated internally by syslogd
- lpr – line printer subsystem
- news – network news subsystem

Závažnost (severita) může nabývat hodnoty:

- emerg – system is unusable
- alert – action must be taken immediately
- crit – critical conditions
- err – error conditions
- warn – warning conditions
- notice – normal but significant condition
- info – informational
- debug – debug-level messages

První příklad z této skupiny ukazuje vypsání záznamů souvisejících s jádrem operačního systému.

```
$ sudo dmesg -f kern
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ [ 9.555887] sd 2:0:0:0: Attached scsi generic sg0 type 0
  [10.269320] Adding 2094076k swap on /dev/sda5.  Priority:-2 extents:1 across:2094076k FS
```

Další příklad vypsání záznamů souvisejících s jádrem operačního systému a démony (službami).

```
$ dmesg -f kern,daemon
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ [ 6.109151] systemd[1]: Inserted module 'autofs4'
  [10.269320] Adding 2094076k swap on /dev/sda5.  Priority:-2 extents:1 across:2094076k FS
```

V případě závažnosti můžeme pro omezení využít výše uvedené hodnoty, například vypsat všechny záznamy spadající do kategorií varování a informativní.

```
$ dmesg -l warn,info
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ [    4.266542] sd 2:0:0:0: Power-on or device reset occurred
  [10702.718540] fuse init (API version 7.27)
```

2.3.2 Ostatní protokoly

S omezením na kontext i závažnost lze s jistými omezeními pracovat i mimo příkaz `dmesg`, například záznamy související s jádrem operačního systému máme k dispozici v protokolu `kern.log` (*vyučující může upozornit na rozdíl kruhového bufferu jádra a souboru protokolu, viz teoretická část)*.

Následující příklad ukazuje vyhledávání záznamů z protokolu `syslog`, které by mohly souviset s chybami (hledáme text error a fail).

```
$ sudo awk 'tolower($0) ~ /error|fail/' /var/log/syslog
```

Výstup (dle skutečného obsahu, zkráceno):

```
$ May 1 14:27:11 ... g_variant_new_string: assertion 'string != NULL' failed
  May 1 14:27:11 ... g_variant_new_string: assertion 'string != NULL' failed
  May 1 17:05:51 debian gnome-session-binary[2091]: GLib-GIO-WARNING: Error releasing name
```

Kontrolní bod

Studenti vypíšou pomocí příkazu `dmesg` chyby, ale nesmí použít omezení na severitu ani kontext.

2.4 Průběžné zobrazování nových záznamů protokolu

2.4.1 Příkaz `tail`

Vyučující předvede následující postup pro průběžné zobrazování jednoho protokolu:

1. `sudo tail -f /var/log/auth.log`
2. otevře další terminál tak, aby byl výstup z bodu (1) viditelný
3. `sudo -i`
4. vloží 3x špatné heslo a současně komentuje výstup v terminálu z bodu (1)
5. `sudo -i`
6. vloží správné heslo a současně komentuje výstup v terminálu z bodu (1)
7. `exit`
8. ukončí výpis v terminálu z bodu (1) pomocí Ctrl+c

2.4.2 Využití příkazu `screen`

Vyučující předvede následující postup pro průběžné zobrazování dvou protokolů:

1. `sudo apt update`
2. `sudo apt install screen`
3. `screen` a na výzvu stiskne Enter
4. stiskne Ctrl+A a následně S
5. do aktivní (horní části) terminálu vloží `tail -f /var/log/auth.log`
6. stiskne Ctrl+A a následně Tab
7. stiskne Ctrl+A a následně c (malé c)
8. do aktivní (dolní části) terminálu vloží `tail -f /var/log/apt/history.log`
9. otevře další terminál tak, aby byly výstupy z bodů (5) a (8) viditelné
10. `sudo -i`
11. vloží správné heslo a současně komentuje výstup dle bodu (5)
12. `apt install mc`
13. provede instalaci a současně komentuje výstup dle bodu (8)
14. `exit`
15. stiskne Ctrl+c
16. stiskne Ctrl+A a následně Tab
17. stiskne Ctrl+c
18. `exit`

Kontrolní bod

Studenti pomocí příkazů `screen` a `tail` budou vypisovat nové záznamy z protokolu *`apt/history.log` – funkčnost budou vhodným způsobem demonstrovat.*

2.5 Zaslání mailové zprávy při výskytu záznamu v protokolu

2.5.1 Využití příkazu `mail`

Vyučující předvede následující postup:

1. `sudo -i`
2. `apt update`
3. `apt install bsd-mailx`
4. `adduser logy` (heslo jakékoli, ale je nutné si jej zapamatovat)
5. v rámci jednoho celku:

```
$ tail -n1 -f /var/log/auth.log | while read line
do case "$line" in
       *"authentication failure"*) echo "$line" | mail -s "Selhání přihlášení" logy;
      ;;
   esac
done
```

6. otevře další terminál
7. `sudo -i`
8. vloží 3x špatné heslo
9. `su – logy`
10. `mail`
11. komentuje výstupy
12. `exit`
13. ukončí kód v terminálu z bodu (5) pomocí Ctrl+c

(*vyučující může komentovat využití lokální pošty – je možné upozornit na to, že pořadí neodpovídá původním pořadí položek*).

Kontrolní bod

Studenti si zopakují postup předvedený vyučujícím.

Shrnutí a závěr

Studenti se seznámili jak se základy protokolů v operačních systémech linuxového typu, tak i s pokročilými technikami práce včetně podpory pro operativní dohled. Dle svého uvážení může vyučující připravit pro studenty například test, a to jak z pohledu teorie, tak i praktického opakování nad zdokumentovanými postupy.

Seznam použitých zdrojů

Editor RFC. *Dokumenty RFC*. Dostupné z: https://www.rfc-editor.org

SPI. *Manuálové stránky*. Dostupné z: https://manpages.debian.org