

1. Úloha IOS (2023)

Popis úlohy

Česká kontrarozvědka FDTO (**F**akt **D**ěsně **T**ajná **O**rganizace) má podezření, že v jejích řadách působí krtek a služba tak byla kompromitována. Obsah některých přísně tajných digitálních dokumentů byl nejenže vyzrazen tajným službám cizích mocností, ale zřejmě i modifikován, což způsobilo chaos v organizaci některých tajných operací a částečně tak paralyzovalo fungování celé kontrarozvědky. V zájmu národní bezpečnosti je zapotřebí krtek urychleně najít a polapit. Hledání však musí být maximálně diskrétní, aby krtek netušil, že se okolo něj utahuje smyčka.

Protože zavedení nových oficiálních bezpečnostních opatření při práci s utajovanými digitálními dokumenty by mohlo být podezřelé a krtek vyplašit, rozhodlo se vedení kontrarozvědky, že bude potřeba tato bezpečnostní opatření zavést utajeně. Tedy tak, aby si nikdo (kromě nejužšího vedení a Vás) nebyl vědom toho, že nějaká nová opatření vešla v platnost.

Vzhledem k Vaší léty prověřené loajalitě a bezchybné pracovní historii v IT jednotce kontrarozvědky jste byli pro tuto tajnou misi vybráni právě Vy. Vaším úkolem je vytvořit skript, který bude prezentován jako nový interní nástroj pro zvýšení efektivity při práci s elektronickými dokumenty, zatímco skrytě bude použit pro tajné zaznamenávání (tzv. logging) a ohlašování informací o tom, se kterými dokumenty (a kdy) daný uživatel pracoval.

Citlivé dokumenty jsou v kontrarozvědce zásadně uchovávány na serverech a přístup k nim probíhá vzdáleně pomocí příslušných nástrojů. Skript **mole** (**M**akes **O**ne's **L**ife **E**asier) tedy bude fungovat jako tzv. wrapper nad textovými editory, což znamená, že textový editor bude spouštěn skrze skript **mole**. Skript si bude pamatovat, které soubory byly v jakém adresáři prostřednictvím skriptu **mole** editovány. Jednotlivé soubory je zároveň možné přiřazovat do skupin pro jednodušší filtrování při práci s velkým množstvím souborů. Pokud bude skript spuštěn bez parametrů, vybere skript soubor, který má být editován.

Specifikace chování skriptu

JMÉNO

- **mole** – wrapper pro efektivní použití textového editoru s možností automatického výběru nejčastěji či posledně modifikovaného souboru.

POUŽITÍ

- **mole -h**
- **mole [-g GROUP] FILE**
- **mole [-m] [FILTERS] [DIRECTORY]**
- **mole list [FILTERS] [DIRECTORY]**
- **mole secret-log [-b DATE] [-a DATE] [DIRECTORY1 [DIRECTORY2 [...]]]**

Popis

- `-h` – Vypíše nápovědu k použití skriptu (volba `secret -log` by neměla být v nápovědě uvedena; nechceme krčka upozornit, že sbíráme informace).
- `mole [-g GROUP] FILE` – Zadaný soubor bude otevřen.
 - Pokud byl zadán přepínač `-g`, dané otevření souboru bude zároveň přiřazeno do skupiny s názvem `GROUP`. `GROUP` může být název jak existující, tak nové skupiny.
- `mole [-m] [FILTERS] [DIRECTORY]` – Pokud `DIRECTORY` odpovídá existujícímu adresáři, skript z daného adresáře vybere soubor, který má být otevřen.
 - Pokud nebyl zadán adresář, předpokládá se aktuální adresář.
 - Pokud bylo v daném adresáři editováno skriptem více souborů, vybere se soubor, který byl pomocí skriptu otevřen (editován) jako **poslední**.
 - Pokud byl zadán argument `-m`, tak skript vybere soubor, který byl pomocí skriptu otevřen (editován) **nejčastěji**.
 - Pokud bude při použití přepínače `-m` nalezeno více souborů se stejným maximálním počtem otevření, může `mole` vybrat kterýkoliv z nich.
 - Výběr souboru může být dále ovlivněn zadanými filtry `FILTERS`.
 - Pokud nebyl v daném adresáři otevřen (editován) ještě žádný soubor, případně žádný soubor nevyhovuje zadaným filtrům, jedná se o chybu.
- `mole list [FILTERS] [DIRECTORY]` – Skript zobrazí seznam souborů, které byly v daném adresáři otevřeny (editovány) pomocí skriptu.
 - Pokud nebyl zadán adresář, předpokládá se aktuální adresář.
 - Seznam souborů může být filtrován pomocí `FILTERS`.
 - Seznam souborů bude lexikograficky seřazen a každý soubor bude uveden na samostatném řádku.
 - Každý řádek bude mít formát `FILENAME: <INDENT>GROUP_1, GROUP_2, . . .`, kde `FILENAME` je jméno souboru (i s jeho případnými příponami), `<INDENT>` je počet mezer potřebných k zarovnání a `GROUP_*` jsou názvy skupin, u kterých je soubor evidován.
 - Seznam skupin bude lexikograficky seřazen.
 - Pokud budou skupiny upřesněny pomocí přepínače `-g` (viz sekce `FILTRY`), uvažujte při výpisu souborů a skupin pouze záznamy patřící do těchto skupin.
 - Pokud soubor nepatří do žádné skupiny, bude namísto seznamu skupin vypsán pouze znak `-`.
 - Minimální počet mezer použitých k zarovnání (`INDENT`) je jedna. Každý řádek bude zarovnán tak, aby seznam skupin začínal na stejné pozici. Tedy např:

```
FILE1:  grp1,grp2
FILE10: grp1,grp3
FILE:   -
```

- `mole secret-log [-b DATE] [-a DATE] [DIRECTORY1 [DIRECTORY2 [. . .]]]` – Skript za účelem dopadení krčka vytvoří tajný komprimovaný log s informacemi o souborech otevřených (editovaných) skrze skript `mole`.
 - Pokud byly zadány adresáře, tajný log bude obsahovat záznamy o otevřených (editovaných) souborech pouze z těchto adresářů. Neexistující adresáře nebo adresáře bez záznamů budou ignorovány.
 - Pokud nebyl zadán žádný adresář, tajný log bude obsahovat záznamy ze všech evidovaných adresářů.
 - Otevřené (editované) soubory, které mají být v tajném logu zaznamenány, je možné dále omezit pomocí filtrů `-a` a `-b` (viz níže).

Filtry

FILTERS může být kombinace následujících filtrů (každý může být uveden maximálně jednou):

- `[-g GROUP1[, GROUP2[, . . .]]]` – Specifikace skupin. Soubor bude uvažován (pro potřeby otevření nebo výpisu) pouze tehdy, pokud jeho spuštění spadá alespoň do jedné z těchto skupin.
- `[-a DATE]` - Záznamy o otevřených (editovaných) souborech před tímto datem **včetně (volitelně lze implementovat i jako striktně před uvedeným datem; UPDATED 22.3.)** nebudou uvažovány.
- `[-b DATE]` - Záznamy o otevřených (editovaných) souborech po tomto datu **včetně (volitelně lze implementovat i jako striktně po uvedeném datu; UPDATED 22.3.)** nebudou uvažovány.
- Argument DATE je ve formátu YYYY-MM-DD.

Nastavení a konfigurace

- Skript si pamatuje informace o svém spuštění v souboru, který je dán proměnnou MOLE_RC. Formát souboru není specifikován.
 - Pokud není proměnná nastavena, jedná se o chybu.
 - Pokud soubor na cestě dané proměnnou MOLE_RC neexistuje, soubor bude vytvořen včetně cesty k danému souboru (pokud i ta neexistuje).
- Skript spouští editor, který je nastaven v proměnné EDITOR. Pokud není proměnná EDITOR nastavená, respektuje proměnnou VISUAL. Pokud ani ta není nastavená, použije se příkaz vi.

Formát tajného logu

- Tajný log vygenerovaný pomocí příkazu `secret -log` bude uložen v adresáři `.mole` umístěném v domovském adresáři (tedy např. `/home/$USER/.mole/`). Název souboru bude ve formátu `log_USER_DATETIME.bz2`, kde USER odpovídá jménu aktuálního uživatele a DATETIME odpovídá datu a času vytvoření tajného logu.
 - Tajný log bude obsahovat záznamy o všech známých manipulacích (tedy otevření skrze skript `mole`) s vybranými soubory, případně dále omezených na daný časový úsek pomocí přepínačů `-a`, `-b`, nebo jejich kombinací.
 - Formát záznamů v logu bude `FILEPATH; DATETIME_1; DATETIME_2; . . .`, kde
 - `FILEPATH` je reálná cesta k souboru,
 - `DATETIME_N` je datum a čas chronologicky N-tého známého otevření souboru buď napříč celou známou historií, nebo v daném časovém úseku.
 - Záznamy v tajném logu budou seřazeny lexikograficky podle hodnoty `FILEPATH`.
- Formát hodnot `DATETIME` a `DATETIME_N` je `YYYY-MM-DD_HH-mm-ss`.
- Tajný log komprimujte pomocí utility `bzip2`.

Poznámky

- Můžete předpokládat, že nebude zadána skupina se jménem `-` a že názvy skupin nebudou obsahovat znak čárky
- Můžete předpokládat, že názvy souborů (ani jejich cesty) nebudou obsahovat znaky středníku nebo dvojtečky.
- Skript nebere v potaz otevření nebo editace, které byly provedeny mimo skript `mole`.
- Stejně tak pro příkaz `mole [-m] [FILTERS] [DIRECTORY]` skript nebere v potaz soubory,

se kterými dříve počítal a které jsou nyní smazané (u ostatních příkazů není potřeba kontrolovat existenci souboru). Například, pokud byl *posledně* editovaný soubor smazán, volání `mole` otevře *předposledně* editovaný soubor, pokud byl i ten smazán, bude otevřen *předpředposledně* editovaný soubor, atp. **UPDATED 6.3.**

- Při rozhodování relativní cesty adresáře je doporučeno používat reálnou cestu (`realpath`). Důvod např.:

```
$ mole .  
$ mole `pwd`
```

Návratová hodnota

- Skript vrací úspěch v případě úspěšné operace nebo v případě úspěšné editace. Pokud editor vrátí chybu, skript vrátí stejný chybový návratový kód. Interní chyba skriptu bude doprovázena chybovým hlášením.

Implementační detaily

- Skript by měl mít v celém běhu nastaveno `POSIXLY_CORRECT=yes`.
- Skript by měl běžet na všech běžných shellech (`dash`, `ksh`, `bash`). Můžete použít GNU rozšíření pro `sed` či `awk`. Jazyk Perl nebo Python povolen není.
- Skript by měl ošetřit i chybový případ, že na daném stroji utilita `realpath` není dostupná (např. ukončením programu s chybovým kódem).
- **UPOZORNĚNÍ:** některé servery, např. `merlin.fit.vutbr.cz`, mají symlink `/bin/sh` -> `bash`. Ověřte si proto, že skript skutečně testujete daným shellem. Doporučujeme ověřit správnou funkčnost pomocí virtuálního stroje níže.
- Skript musí běžet na běžně dostupných OS GNU/Linux, BSD a MacOS. Studentům je k dispozici virtuální stroj, na kterém lze ověřit správnou funkčnost projektu, s obrazem ke stažení zde: <http://www.fit.vutbr.cz/~lengal/public/trusty.ova> (pro VirtualBox, login: `trusty` / heslo: `trusty`).
- Skript nesmí používat dočasné soubory. Povoleny jsou však dočasné soubory nepřímo tvořené jinými příkazy (např. příkazem `sed -i`). Soubor `MOLE_RC` ani tajné log soubory nejsou v tomto případě chápány jako dočasné soubory.
- Není potřeba řešit možnost souběhu několika instancí skriptu, v takovém případě je chování nedefinované.

Možná rozšíření

- Implementuje u příkazů `mole [-m] [FILTERS] [DIRECTORY]`, `list` a `secret-log` nový přepínač `-r` (recursive), který způsobí, že `mole` bude hledat vyhovující záznamy o otevření (editaci) souborů i mezi záznamy, které se vztahují ke vnořeným adresářům vzhledem k `DIRECTORY` (resp. aktuálnímu adresáři, pokud nebyl `DIRECTORY` zadán).
 - Tedy například příkaz `mole list ~/proj1` by na výstup vypsalo jak soubor `main.c`, tak soubor `.git/config` za předpokladu, že má v logu `MOLE_RC` uloženy záznamy o otevření souborů `~/proj1/main.c` a `~/proj1/.git/config`.
- Implementujte u příkazů `mole [-m] [FILTERS] [DIRECTORY]` a `list` nový přepínač `-d` (default), který způsobí, že `mole` bude pracovat pouze se záznamy spuštění (editace) souborů bez specifikované skupiny.
 - Tedy např. při použití příkazu `mole -d ~` bude ignorován záznam o spuštění `mole -g bash ~/.bashrc`.

- Přepínač -d bude výlučný s přepínačem -g, tedy v rámci jednoho spuštění mole nemohou být zadány oba přepínače zároveň.
- Implementace přepínačů -d a -r je nepovinná; korektní implementace může vynahradit jiné bodové ztráty.

Odevzdávání

Odevzdávejte pouze skript mole (nebalte ho do žádného archivu) do IS VUT.

Rady

- Dobrá dekompozice problému na podproblémy Vám může značně ulehčit práci a předejít chybám.
- Naučte se *dobře* používat **funkce** v shellu.
- Následující utility Vám mohou při řešení projektu velmi pomoci:
 - [getopts](#)
 - [date](#)
 - [cut](#)
 - [sed](#)
 - [wc](#)
 - [printf](#)
 - [awk](#)

Příklady použití

Následující příklady předpokládají, že skript mole je dostupný v jedné z cest v proměnné PATH.

Předpokládejte, že ukázky chování v bodech 2 až 6 jsou na sobě nezávislé a každá z nich navazuje přímo na ukázkou č. 1. Tedy například důsledky ukázky č. 2 se nepromítanou do výsledků ukázek č. 3, 4, atd. (**UPDATED 21.3.**)

1. Editace různých souborů:

```
$ export MOLE_RC=$HOME/.config/molerc
$ date
Thu Feb 16 01:37:14 PM CET 2023
$ mole ~/.ssh/config
$ mole -g bash ~/.bashrc
$ mole ~/.local/bin/mole
$ mole -g bash ~/.bashrc                                     # (D)
$ mole ~/.indent.pro
$ mole ~/.viminfo

$ date
Mon Feb 20 07:21:09 PM CET 2023
$ mole -g bash ~/.bash_history
$ mole -g git ~/.gitconfig
$ mole -g bash ~/.bash_profile                               # (C)
$ mole -g git ~/proj1/.git/info/exclude
$ mole ~/.ssh/known_hosts                                    # (A)
$ mole -g git ~/proj1/.git/config
$ mole -g git ~/proj1/.git/COMMIT_EDITMSG
$ mole ~/proj1/.git/COMMIT_EDITMSG
$ mole -g git ~/proj1/.git/config                             # (F)
$ mole -g project ~/proj1/main.c
```

```

$ mole -g project ~/proj1/struct.c
$ mole -g project ~/proj1/struct.h
$ mole -g project_readme ~/proj1/README.md

$ date
Fri Feb 24 03:52:34 PM CET 2023
$ mole -g git2 ~/.gitconfig
$ mole ~/proj1/main.c
$ mole ~/.bashrc # (E)
$ mole ~/.indent.pro
$ mole ~/.vimrc # (B)

```

2. Opětovná editace:

```

$ cd ~/.ssh
$ mole
... # spustí se editace souboru ~/.ssh/known_hosts (odpovídá řádku A)
$ mole ~
... # spustí se editace souboru ~/.vimrc (odpovídá řádku B)
$ mole -g bash ~
... # spustí se editace souboru ~/.bash_profile (odpovídá řádku C)
$ mole -g bash -b 2023-02-20 ~
... # spustí se editace souboru ~/.bashrc (odpovídá řádku D)
$ cd
$ mole -m
... # spustí se editace souboru ~/.bashrc (odpovídá řádku E)
$ mole -m -g git ~/proj1/.git
... # spustí se editace souboru ~/proj1/.git/config (odpovídá řádku F; ve skupině git
byl daný soubor editován jako jediný dvakrát, zbytek souborů jednou)
$ mole -m -g tst
... # ! chyba, nebyl nalezen žádný soubor k otevření
$ mole -a 2023-02-16 -b 2023-02-20
... # ! chyba, nebyl nalezen žádný soubor k otevření

```

3. Zobrazení seznamu editovaných souborů (OPRAVENO 11.3.):

```

$ mole list $HOME
.bash_history: bash
.bash_profile: bash
.bashrc: bash
.gitconfig: git,git2
.indent.pro: -
.viminfo: -
.vimrc: -
$ mole list -g bash $HOME
.bash_history: bash
.bash_profile: bash
.bashrc: bash
$ mole list -g project,project_readme ~/proj1
main.c: project
README.md: project_readme
struct.c: project
struct.h: project
$ mole list -b 2023-02-20 $HOME
.bashrc: bash
.indent.pro: -
.viminfo: -
$ mole list -a 2023-02-23 $HOME
.bashrc: -
.gitconfig: git2
.indent.pro: -
.vimrc: -
$ mole list -a 2023-02-16 -b 2023-02-24 -g bash $HOME
.bash_history: bash
.bash_profile: bash
$ mole list -a 2023-02-20 -b 2023-02-24 $HOME
$ mole list -g grp1,grp2 $HOME

```

4. Vytvoření tajného logu:

```
$ date
Fri Feb 24 04:13:58 PM CET 2023
$ mole secret-log
$ bunzip2 -k --stdout /home/trusty/.mole/log_trusty_2023-02-24_16-14-01.bz2
/home/trusty/.bash_history;2023-02-20_19-21-13
/home/trusty/.bash_profile;2023-02-20_19-21-38
/home/trusty/.bashrc;2023-02-16_13-37-31;2023-02-16_13-38-02;2023-02-24_15-53-05
/home/trusty/.gitconfig;2023-02-20_19-21-22;2023-02-24_15-52-39
/home/trusty/.indent.pro;2023-02-16_13-38-34;2023-02-24_15-53-18
/home/trusty/.local/bin/mole;2023-02-16_13-37-46
/home/trusty/proj1/.git/COMMIT_EDITMSG;2023-02-20_19-25-12;2023-02-20_19-25-18
/home/trusty/proj1/.git/config;2023-02-20_19-24-59;2023-02-20_19-25-27
/home/trusty/proj1/.git/info/exclude;2023-02-20_19-22-04
/home/trusty/proj1/main.c;2023-02-20_19-25-51;2023-02-24_15-52-48
/home/trusty/proj1/README.md;2023-02-20_19-26-36
/home/trusty/proj1/struct.c;2023-02-20_19-26-03
/home/trusty/proj1/struct.h;2023-02-20_19-26-18
/home/trusty/.ssh/config;2023-02-16_13-37-19
/home/trusty/.ssh/known_hosts;2023-02-20_19-22-48
/home/trusty/.viminfo;2023-02-16_13-38-53
/home/trusty/.vimrc;2023-02-24_15-54-04
$ date
Fri Feb 24 04:15:13 PM CET 2023
$ mole secret-log -b 2023-02-22 ~/proj1 ~/.ssh
$ bunzip2 -k --stdout /home/trusty/.mole/log_trusty_2023-02-24_16-15-22.bz2
/home/trusty/proj1/main.c;2023-02-20_19-25-51
/home/trusty/proj1/README.md;2023-02-20_19-26-36
/home/trusty/proj1/struct.c;2023-02-20_19-26-03
/home/trusty/proj1/struct.h;2023-02-20_19-26-18
/home/trusty/.ssh/config;2023-02-16_13-37-19
/home/trusty/.ssh/known_hosts;2023-02-20_19-22-48
```

5. Použití nepovinného přepínače -r:

```
$ date
Sat Feb 25 09:00:04 AM CET 2023
$ mole list -r -b 2023-02-24 ~/proj1
.git/COMMIT_EDITMSG: git
.git/config: git
.git/info/exclude: git
main.c: project
README.md: project_readme
struct.c: project
struct.h: project
$ mole -m -r -b 2023-02-24 ~/proj1
... # spustí se editace souboru ~/proj1/.git/COMMIT_EDITMSG nebo ~/proj1/.git/config
$ mole -r -m -g git -b 2023-02-24 ~/proj1
... # spustí se editace souboru ~/proj1/.git/config
```

6. Použití nepovinného přepínače -d:

```
$ date
Sat Feb 25 09:08:10 AM CET 2023
$ mole list -d -b 2023-02-20 $HOME
.indent.pro: -
.viminfo: -
$ mole list -a 2023-02-23 -d $HOME
.bashrc: -
.indent.pro: -
.vimrc: -
$ mole -m -d -b 2023-02-20 ~
... # spustí se editace souboru ~/.indent.pro nebo ~/.viminfo
$ mole -m -d -g bash -b 2023-02-20 ~
... # ! chyba, přepínače -d a -g jsou výlučné
```

- [Popis úlohy](#)
- [Specifikace chování skriptu](#)
- [Popis](#)
- [Filtry](#)
- [Nastavení a konfigurace](#)
- [Formát tajného logu](#)
- [Poznámky](#)
- [Návratová hodnota](#)
- [Implementační detaily](#)
- [Možná rozšíření](#)
- [Odevzdávání](#)
- [Rady](#)
- [Příklady použití](#)

Built with [Pandoc](#) using [pandoc-bootstrap](#) theme