Kurs administrowania systemem Linux 2022

Lista zadań na pracownię nr 14

Na zajęcia 20 i 22 czerwca 2022

Zadanie 1 (1 pkt). Przygotuj krótkie omówienie następujących programów i pakietów oprogramowania:

- `e2fsprogs`
- `e2tools`
- The Sleuth Kit
- `disktype`

Zadanie 2 (2 pkt). Zapoznaj się z poleceniem `mke2fs(8)` i konfiguracją `mke2fs.conf(5)`. Przygotuj krótkie omówienie dostępnych możliwości wpływania na parametry tworzonego systemu plików. Utwórz obraz niewielkiego dysku (np. 1 GiB) wydając np. polecenie

```
truncate -s 1G disk.img
```

Zauważ, że plik `disk.img` nie zajmuje żadnego miejsca na dysku, choć jego rozmiar wynosi 1 GiB. Zamaż go losowymi danymi.

Dygresja: bardzo szybkim sposobem zamazania pliku/dysku `disk.img` jest wydanie następującej sekwencji poleceń:

```
sudo cryptsetup open --type=plain --key-file=/dev/urandom disk.img tmpdisk
sudo dd if=/dev/zero of=/dev/mapper/tmpdisk bs=1M oflag=direct conv=fsync status=progress
sudo cryptsetup close tmpdisk
```

Wyjaśnij, co się powyżej wydarzyło. Porównaj szybkość powyższej procedury z metodą korzystającą wprost z generatora losowego, np.

```
dd if=/dev/urandom of=disk.img bs=1M count=1024 oflag=direct conv=fsync status=progress
```

oraz z zapisaniem całego dysku zerami.[1] Ile miejsca na dysku zajmuje plik `disk.img` po zamazaniu losowymi danymi?

Załóż na obrazie dysku `disk.img` system plików ext2. Nie rezerwuj przy tym miejsca dla użytkownika `root` (zwykle rezerwacja wynosi 5%). Nadaj systemowi plików jakąś ładną etykietę. Obejrzyj zawartość pliku `disk.img`. Co się stało z losowymi danymi? Ile miejsca na dysku zajmuje teraz ten plik? Jakie dane zajmują to miejsce? Nie kasuj pliku `disk.img`. Przyda się w następnych zadaniach!

Zadanie 3 (2 pkt). Za pomocą polecenia `dd` skojarzonego z poleceniem `hd` (tj. `hexdump(1)` z opcją `-C`) ujawnij zawartość drugiego kilobajtu pliku `disk.img`. Co się tam znajduje? Skorzystaj z tabeli opisującej znajdującą się tam zawartość — zob. np.

```
https://www.kernel.org/doc/html/latest/filesystems/ext4/globals.html#super-block
```

i odpowiedz na pytanie, jakie parametry ma system plików znajdujący się na tym urządzeniu. Porównaj zebrane informacje z wynikami działania programów `dumpe2fs(8)` i `fsstat` z pakietu The Sleuth Kit.

[1] Różnicę widać szczególnie dla oryginalnego PRNG napisanego przez Teodora Ts'o w 1994 roku. Np. na Intel® Core™ 2 Duo E8400 @3.00 GHz z jądrem 3.16 spowolnienie wynosi 7.2 raza. Nowa implementacja PRNG w Linuksie (również zrobiona przez Ts'o) obecna w jądrze 3.8 i nowszym, używająca ChaCha20 Bernsteina, jest znacznie szybsza.

Zadanie 4 (2 pkt). Co to jest alokator Orłowa (Orlov allocator)? Przygotuj krótkie omówienie. Jakie inne alokatory można wykorzystać podczas montowania systemu plików ext234?

Zadanie 5 (1 pkt). Przygotuj omówienie narzędzi `filefrag(8)` i `e2freefrag(8)`. Zamontuj system plików znajdujący się w pliku `disk.img` i zapisz do niego jakiś duży plik `plik1`. Ujawnij jego fragmentację oraz histogram pozostałych wolnych fragmentów.

Zadanie 6 (2 pkt). Przygotuj omówienie programu `debugfs(8)`. Zamontuj ponownie system plików znajdujący się w pliku `disk.img` (zakładamy, że znajduje się na nim plik `plik1` z poprzedniego zadania). Uruchom program `debugfs`.

1. Wyświetl poleceniem `stats` zawartość superbloku (zauważ, że jest to wywołanie `dumpe2fs`).
2. Wyświetl zawartość katalogu poleceniem `ls`. Wypróbuj opcje `-l` oraz `-d`.
3. Wyświetl histogram wolnych bloków poleceniem `freefrag` i fragmentację pliku `plik1` poleceniem `filefrag plik1`.
4. Wyświetl zawartość i-węzła pliku `plik1` poleceniem `inode_dump plik1`.
5. Zlokalizuj położenie i-węzła pliku `plik1` poleceniem `imap plik1`. Obejrzyj plik `disk.img` poleceniem `hexdump` i sprawdź, że w podanym miejscu faktycznie znajduje się i-węzeł pliku `plik1`.
6. Zinterpretuj zawartość i-węzła pliku `plik1` poleceniem `stat plik1`.
7. Wyświetl listę bloków należących do pliku `plik1` poleceniem `blocks plik1`.
8. Zauważ, że zamiast posługiwać się nazwami plików w poprzednich poleceniach, możesz podawać numery i-węzłów, nawet nieużywanych, np. `stat <13>`.
9. Sprawdź poleceniami `testi` i `testb`, czy dany i-węzeł bądź blok są w użyciu. Wyświetl za pomocą polecenia `hexdump` obraz dyskietki i sprawdź, że odpowiednie bity w bitmapach i-węzłów oraz bloków zgadzają się z informacjami podawanymi przez polecenia `testi` i `testb`.
10. Użyj poleceń `seti`, `setb`, `freei` i `freeb` aby zmienić odpowiednie bity w bitmapach węzłów i bloków. Odmontuj system plików i zrób porządek z poprzestawianymi bitami za pomocą programu `e2fsck(8)`.
11. Skopiuj na dysk mały plik tekstowy. Obejrzyj jego i-węzeł i zanotuj numery bloków należących do tego pliku. Użyj polecenia `block_dump` w celu wyświetlenia zawartości bloków tego pliku. Obejrzyj obraz dysku za pomocą polecenia `hexdump` i sprawdź, że i-węzeł tego pliku zawiera poprawne numery bloków. Zanotuj numer tego i-węzła.
12. Skasuj ten plik. Wyświetl zawartość katalogu wraz z usuniętymi plikami. Wyświetl i-węzeł skasowanego pliku. Jaki rozmiar i numery bloków zawiera ten i-węzeł? Czy nie jest to przerażające? Sprawdź, że przynajmniej zawartość pliku jest dokładnie w tych blokach, gdzie była.

Zadanie 7 (1 pkt). Jaki jest najmniejszy rozmiar urządzenia blokowego w bajtach, na którym można założyć system plików ext2? Dlaczego tyle?

Zadanie 8 (2 pkt). Przygotuj obraz niewielkiego dysku i załóż na nim system ext z księgowaniem (np. ext4). Zamontuj go, zapisz na nim jakiś plik tekstowy, np. `/etc/mke2fs.conf`, zsynchronizuj dyski, skasuj plik i odmontuj system plików. Użyj programu `debugfs` do przejrzenia zawartości księgi. Odczytaj z niej zawartość i-węzła tego pliku sprzed skasowania i porównaj z bieżącą zawartością. Odczytaj mapę bloków i zlokalizuj zawartość tego pliku na dysku.

Zadanie 9 (2 pkt). Wypróbuj następujące programy analizujące księgę w celu odzyskania pliku:

- `ext4magic`,
- `ext3grep`,
- `extundelete`.

oraz następujące programy przeszukujące sektory dysku w celu odzyskania pliku:

- `magicrescue`,
- `scalpel`,
- `foremost`.

Zadanie 10 (1 pkt). Utwórz system plików ext4 z opcją `encrypt`. Zamontuj go, skopiuj do niego jakiś plik tekstowy, np. `/etc/mke2fs.conf`, utwórz pusty katalog i zaszyfruj go poleceniem

```
# e4crypt add_key katalog
```

Spróbuj przenieść plik tekstowy do tego katalogu poleceniem `mv`. Skopiuj następnie ten plik do zaszyfrowanego katalogu. Odmontuj system plików. Za pomocą programu `debugfs` obejrzyj zawartość zaszyfrowanego katalogu i skopiowanego pliku.