Corso di Laurea in Informatica Applicata
Laboratorio di Sistemi Operativi
Anno Accademico 2010 – 2011

File Wiping

1 Introduzione

La rimozione di un file attraverso, ad esempio, il comando `rm` su Linux non provoca la cancellazione dei dati del file ma semplicemente la dealoccazione delle risorse assegnate al file rimosso. I dati del file non vengono cancellati (ad esempio scrivendo sui blocchi assegnati al file tutti zero) e possono perciò essere facilmente recuperati: questo problema ǹoto come *data remanence*.

In caso di dati sensibili e privati è quindi opportuno sovrascrivere il contenuto dei file in modo da impedirne il recupero (*file wiping* o *file whredding*). I tre metodi più diffusi per il file wiping sono:

- zeroing: il contenuto del file viene sovrascritto con 0. Questo semplice metodo impedisce il recupero dei dati usando semplicemente chiamate di sistema.
- 7-passes wipe: il contenuto del file viene sovrascritto sette volte usando vari pattern di bit.
- 35-passes wipe, il più sicuro.

Oggi viene normalmente usato un singolo passo poiché è molto complesso, se non impossibile, recuperare dati sovrascritti su hard disk moderni. Per una semplice overview del problema si consulti la pagina di wikipedia sulla 'data remanence'.

2 Programa richiesto

Implementare il programma `wipe` per il 7-passes file wiping. I pattern da usare sono i seguenti:

1. 0xF6;
2. 0x00;
3. 0xFF;
4. random;
5. 0x00;
6. 0xFF;
7. random.

Il programma riceverà il nome del file da sovrascrivere da linea di comando. Il programma quindi sovrascriverà il file prima con 0xF6, poi con 0x00, ..., infine con un pattern causuale di bit.

2.1 Consigli sull'implementazione

Suddividere il codice in almeno due file sorgenti, uno contenente la procedura main e il secondo contenente tutte le funzioni necessarie per effettuare il wiping.

Il file non dovrà essere aperto e chiuso ad ogni iterazione dell'algoritmo (riposizionare l'indicatore della posizione del file con `rewind`).

Verificare gli errori per ogni funzione invocata. In caso di errore stampare un messaggio comprensibile sullo standard error.

Durante lo sviluppo usate file (da sovrascrivere) molto piccoli in modo da verificare più facilmente le operazioni effettuate dal vostro programma.

2.2 Strutture dati e chiamate di sistema

- il file verrà aperto con la chiamata di sistema `open`;
- la dimensione del file e la dimensione dei blocchi su disco possono essere conosciuti accedendo alla struttura dati `stat` attraverso la chiamata `fstat`. In particolare, nella struttura `stat` è indicata la dimensione ottimale del buffer da usare nelle operazioni di I/O su disco.
- Nei passi 2 e 5 il buffer contenente zero può essere creato con la chiamata `bzero` o leggendo i byte dallo special device `/dev/zero`.
- Nei passi 4 e 7 il buffer contenente bit 'casuali' può essere letto dallo special device `/dev/random`.
- In tutti gli altri passi usare una delle funzioni del gruppo `memset_pattern` (consiglio: usare `memset_pattern4`).
- Il file dovrà essere cancellato attraverso l'invocazione della funzione `unlink`. Nel codice proposto come soluzione, le righe per la cancellazione sono commentate per evitare di creare il file per ogni esecuzione dell'eseguibile prodotto.

2.3 Disclaimer

- Il codice sorgente del programma è una prima stesura e non è stato verificato. Segnalate quindi eventuali errori e linee di codice poco chiare in modo che possa pubblicare un codice migliore.
- Così come il codice sorgente non è stato oggetto di analisi scrupolose, il programma non è stato testato estensivamente. Segnalate quindi eventuali errori in modo che possa correggere la soluzione proposta.
- La soluzione non è da considerarsi ottimale: sarò lieto di sostituirla con una da voi proposta o di correggere il codice in caso di vostre osservazioni.

2.4 Soluzione

La soluzione contiene i seguenti file:

- `Makefile`;
- `wipe.c`: file contenente il metodo `main`;
- `wipefns.c`: file contenente le funzioni che implementano il file wiping;
- `wipefns.h`: file header con le funzioni pubbliche esportate da `wipefns.c`;
- `43.txt, divina.txt`: due file di testo sufficientemente grandi da usare durante lo sviluppo.

Ignorate i messaggi di warning in fase di compilazione perché fanno riferimento a due funzioni usate in fase di sviluppo e test.