

THE CORONER'S TOOLKIT



Ciampi Giovanni – Lotti Emanuele

Indice

- Introduzione
- Casi d'uso
- Installazione
- La suite di TCT

2

Introduzione: The Coroner's Toolkit (TCT)

- Presentato per la prima volta nell'Agosto del 1999 dai suoi autori Dan Farmer and Wietse Venema durante un seminario all' IBM T.J. Watson Research Center presso Yorktown Heights (NY, USA).



3

Cosa è un Coroner?

- Un coroner è un avvocato (o un medico in alcuni casi) che agisce da ufficiale giudiziario. Il suo lavoro consiste nell'indagare sui casi di morte che gli vengono segnalati.



4

Cosa è un Coroner?

- Il lavoro del Coroner consiste nel:
 - Scoprire le cause della morte.
 - Determinare l'orario della morte.
 - Determinare se la morte è stata naturale o meno.
- Cosa ha a che fare tutto questo con l'analisi dei sistemi?



5

L'Analisi dei Sistemi

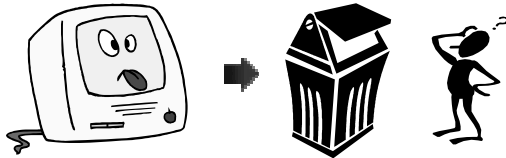
- Un computer contiene un gran numero di informazioni, quali ad esempio:
 - Informazioni sui processi in corso.
 - Dati temporali dei file.
 - Contenuto della memoria.
 - Elenco degli accessi (nel caso di una rete).



6

Sistemi danneggiati

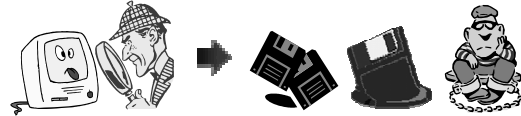
- I dati di un sistema danneggiato in modo grave sono irrecuperabili?



7

Recupero dei Dati

- Assolutamente no! Un'attenta analisi di un sistema danneggiato permette varie cose:
 - Recupero dei dati ancora integri.
 - Recupero dei dati danneggiati.
 - Recupero di informazioni utili per capire le cause del guasto (anche in caso di un sistema violato).



8

The Coroner's Toolkit

- Il Coroner's Toolkit è una suite di applicazioni creata appositamente per il recupero dei dati e l'analisi dei sistemi.



9

Il Recupero dei Dati...

- Per poter recuperare dei dati da un sistema danneggiato è necessario esaminare il contenuto dei seguenti dispositivi:
 - Disco Rigido e Memorie di Massa.
 - Memoria Volatile.
 - Dispositivi di Rete (nel caso di reti locali, anche altri computer).

10

...e l'Analisi del Sistema.

- L'analisi del sistema permette invece di scoprire:
 - Il tipo e numero dei sistemi operativi utilizzati, e la presenza di eventuali aggiornamenti, pacchetti aggiuntivi *et similia*.
 - I programmi installati.
 - La configurazione del sistema.
 - L'elenco delle ultime operazioni effettuate (accessi, utilizzo di programmi o di risorse, ecc. ecc.).

11

Caso d'uso: Recupero file

- Scenari:
 - Violazione di sistemi.
 - Guasti.
 - Errori dell'utente.
 - Cancellazioni volontarie.
- L'eliminazione di un file avviene cancellandone tutti i riferimenti nel filesystem.
- A meno che i settori in cui era scritto non siano stati riutilizzati, il file è ancora fisicamente presente.

12

Precauzioni Recupero File

- Disattivare immediatamente il sistema.
- Smontare tutte le unità di memoria interessate e rimontarle in read-only.
- Effettuare delle copie di sicurezza del materiale.
- Non generare l'output sulla stessa unità di memoria che contiene i dati.

13

Caso d'uso: Analisi del sistema

- Scenari:
 - Indagini su sistemi violati.
 - Guasti o manutenzione.
 - Test eseguiti sul sistema.
- E' necessario ridurre al minimo le operazioni eseguite sul sistema.
- Alcune operazioni, come ad esempio lo svuotamento della memoria, possono causare la perdita di dati importanti.

14

Precauzioni per l'analisi del sistema

- Nel caso sia necessario il recupero della memoria o dei processi in corso, non riavviare il sistema.
- Ridurre il numero di operazioni eseguite al minimo indispensabile.
- Evitare qualsiasi operazione sui file di cui si intende conoscere le informazioni temporali.
- Limitare l'accesso al sistema.

15

Requisiti dell'Analisi 1

- Un analisi accurata deve soddisfare un certo numero di requisiti.
- **1. Deve essere comprensibile anche per chi non ha familiarità con il sistema.**



16

Requisiti dell'Analisi 2

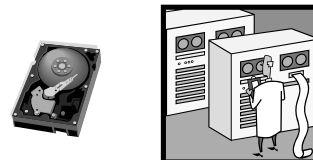
- **2. Deve permettere il recupero di dati utilizzabili in ambito legale.**



17

Requisiti dell'Analisi 3

- **3. Deve essere eseguibile su sistemi o memorie di massa smontabili o difficilmente trasportabili.**



18

Il Problema dell'Analisi

- Nel caso si vogliano analizzare dispositivi di memoria trasportabili è comunque necessario l'uso del TCT?



19

Il Problema dell'Analisi

- Certamente! Anche se un'unità di memoria è offline è comunque possibile che un'analisi superficiale causi un'alterazione dei dati.



20

Problematiche Attuali

1. Tecniche di investigazione molto antiche.
2. Aggiornamenti eseguiti con lentezza.
3. Tools carenti sotto molti punti di vista.
4. Personale tecnico non specializzato.

21

Panoramica del Coroner's Toolkit

- Scritto in Perl e in C
- Diviso in 4 applicativi
 - Grave-robber
 - MacTime
 - C Tools
 - Unrm e Lazarus

22

Panoramica del Coroner's Toolkit

- **Grave-robber** : Recupero porzioni di dati grezzi.
- **MacTime** : Raccolta dati temporali.
- **C Tools** : Recupero informazioni sul sistema.
- **Unrm e Lazarus** : Ripristino file danneggiati.

23

Prerequisiti del TCT

- Un compilatore C.
- Perl ver. 5.0004 (o più recente)
- Utility *Isot* (raccomandata)
- Molto spazio libero.
- Sistemi operativi supportati:
FreeBSD, OpenBSD, BSD/OS,
SunOS, Linux.

24

Installazione di TCT

- Pacchetto disponibile sulla pagina:
<http://www.porcupine.org/forensics/>
- Decompressione del pacchetto.
- Esecuzione del comando make.
- Esecuibili creati nella cartella /bin.

25

Installazione: esempio

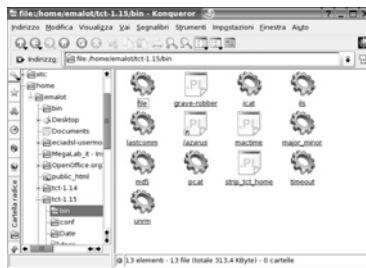


```
linux: # cd ..
linux:/ # cd home
linux/home # cd tct-1.15
linux/home/tct-1.15 # ls
DIRS
.: Makefile conf lib
Source CD_PATCHES doc patchlevel
CHANGES README.FIRST etc quick-start
COPYRIGHT TODO extras reconfig
Src TODO before-next-release help-recovering-file src
INSTALL additional-resources help-when-broken-into
LICENSE bibliography issues
linux/home/tct-1.15 # make
```

Esecuzione del comando make dalla directory principale. I codici del TCT verranno compilati.

26

Installazione: esempio



I file eseguibili vengono creati con successo nella directory /bin.

27

La Suite di Applicazioni

- Grave-Robber
- MacTime
- C Tools
- Unrm e Lazarus

28

Grave-Robber

- Raccolta di grosse quantità di informazioni:
 - Accessi.
 - Ultime operazioni effettuate dagli utenti.
 - Processi.
 - Informazioni su dati critici.



29

Esecuzione del Grave-Robber

- Digitare *grave-robber*, oppure *bin/grave-robber*.
- Consigliabile eseguirlo da root.
- Analisi di default eseguita sull'intero sistema.

30

Esecuzione del Grave-Robber

```
Shell - Konsole
Sessione Modifica Visualizza Segnalibri Impostazioni Aiuto
linux:~ # cd ..
linux:~ # cd home
linux:~/home # cd enalot
linux:~/home/enalot # cd tet-1.15
linux:~/home/enalot/tet-1.15 # perl ./bin/grave-robber -c ~/home/enalot -o LINUX
```

Grave-Robber: esecuzione.

31

Ordine di Volatilità

- Raccolta delle informazioni eseguita secondo l'ordine di volatilità.
- Dati in memoria privilegiati rispetto a quelli sul disco rigido.



32

Procedura e Configurazione

- Raccolta dei dati generalmente lunga.
- Analisi eseguita di default su parti del file system considerati importanti.
- Directory `/etc`, `/bin`, `/usr/local/bin`
- Priorità modificabili editando il file `look@first`, (directory `/conf`).

33

Tipologie di Dati

- Materiale raccolto molto vario:
 - Informazioni sui processi.
 - Informazioni sulle periferiche (in particolare dischi e partizioni).
 - File di importanza rilevante (file di configurazione, file di log, altri file critici).
 - Elenco degli accessi al sistema.

34

Problemi tecnici e forensi

1. Rischio di alterazione dei dati.
2. Procedura eseguita sulla copia originale.
3. Necessità di eseguire il minimo numero di operazioni sui dati.
4. Impossibilità di eseguire un'analisi davvero sicura.



35

Possibili soluzioni

- Analisi offline tramite l'uso dell'opzione `-f`.
- Attraversamento del file system disabilitato.
- Analisi eseguita su una copia dei dati.



36

Documenti MD5

- Strumento importante per la ricerca di materiale 'interessante' dal punto di vista investigativo.
- Il National Institute of Science of Technology possiede un database di checksum MD5 di vario interesse.
- Effettuando un raffronto tra i file MD5 ottenuti in una analisi e quelli del database si possono scovare file di importanza elevata all'interno del sistema.

43

body e body.s

- Contengono:
 - Il database dei file esaminati da Grave-Robber.
 - Informazioni come i permessi, i dati temporali, le user-ID e le group ID, la grandezza dei file.
 - In *body.s* c'è il database di tutti i file SUID. C'è anche in *body*, in *body.s* la lettura n'è agevolata.

44

body

Esempio di file body.

45

Coroner.log e Error.log

- Contengono:
 - La data e l'ora di tutti i programmi eseguiti dal Grave-Robber.
 - Tutti gli errori segnalati durante l'esecuzione.
 - Si trovano entrambe nella directory *main*.

46

Coroner.log e Error.log



I file di log contenuti nella directory main.

47

removed_but_running

- Contiene:
 - Tutti i file cancellati ancora aperti o in esecuzione durante l'esecuzione di Grave-Robber.
 - Viene riempita dai programmi *ils* e *icat*.

48

trust

- Contiene:
 - Tutti i file di sistema che possono fornire dati circa *trust relationship*.
 - Queste ultime possono essere intese come privilegi concessi dall'amministratore di un dominio a una macchina che fa parte di un altro dominio.

55

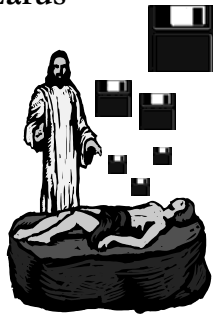
MD5_all

- In questo file vi è un sommario di tutti i file creati durante l'analisi e dei relativi checksum MD5.

56

Unrm e Lazarus

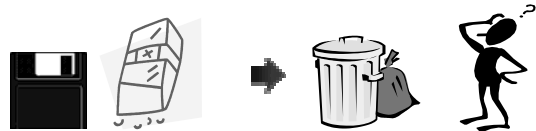
- Programmi correlati tra di loro.
- Recupero e ripristino di file.
- Unrm recupera dati grezzi dalle porzioni di spazio non più utilizzate.
- Lazarus setaccia e analizza queste porzioni di dati.



57

La Cancellazione dei dati

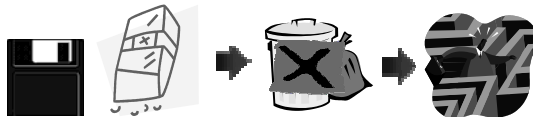
- Quando un file viene cancellato diviene irrimediabilmente irrecuperabile?



58

La Cancellazione dei dati

- Assolutamente no! Ogni riferimento al file viene cancellato dal file system. Ma il file è ancora presente sull'unità di memoria. Per recuperarlo basta trovarlo.



59

Lazarus - Passi

- Esecuzione:
 - 1) Lettura di una porzione di dati (di default 1k).
 - 2) Determinazione del tipo di dati letti – binari o testo.
 - 3) a) Nel caso i dati siano di tipo testuale, vengono controllati in modo da ottenere il testo.

60

Lazarus - Passi

- 3) b) Se sono binari, viene eseguito il comando *file*. Se la classificazione fallisce, viene effettuato un controllo dei primi byte per verificare se sono in formato ELF.
- 4) I file vengono classificati. Se il loro tipo coincide con quello del blocco precedente, la porzione appena esaminata viene concatenata a quest'ultimo.
- 5) Altrimenti si assume che faccia parte dell'ultimo blocco riconosciuto.

61

Lazarus - Passi

- 6) L'output viene generato in due forme: i dati veri e propri e un file contenente una mappa.
 - I dati grezzi sono esaminati blocco a blocco.
 - E' possibile effettuare l'analisi byte a byte.
 - In questo caso il processo dura molto più a lungo.

62

Prerequisiti del Recupero Dati

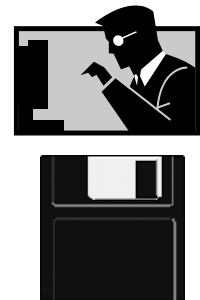
- Un secondo sistema che può riconoscere il disco.
- Un secondo disco su cui scrivere i risultati dell'analisi.



63

Rischi

- Produrre l'output sulla stessa unità di memoria esaminata è rischioso.
- I file ricercati vengono facilmente sovrascritti se si eseguono operazioni di scrittura su quella stessa unità.



64

Prerequisiti del Recupero Dati

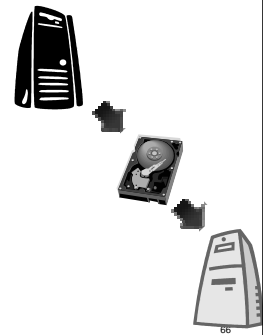
- Una quantità di spazio libero pari almeno al doppio di quella esaminata.
- Un bel po' di tempo libero.



65

Procedura di Recupero Dati

- Montare il disco su un'altra macchina in read only.
- Lanciare *Unrm*.
- Lanciare *Lazarus*.



66

Unrm: Utilizzo

- Schema di utilizzo:
`./ unrm -f<fstype> device [start-stop] <opzioni>`
- Argomenti:
fstype : specifica il tipo del file system. I valori di default sono *ufs* per Unix e *ext2fs* per Linux.
device : nome della partizione sulla quale si desidera cercare. *[start-stop]* : porzione dei memoria che si desidera analizzare. Questo campo può essere omesso.

67

Lazarus: Utilizzo

- Schema di utilizzo:
`./ lazarus -D<directory> <opzioni>`
- Argomenti:
directory : nome della directory in cui salvare i blocchi di dati ottenuti.

68

Output di Lazarus 1

- L'Output di Lazarus viene fornito su una sorta di mappa presentata come una pagina HTML.
- E' possibile accedere al contenuto di ogni file cliccando sul link corrispondente.
- Ogni tipo di file è identificato da una lettera maiuscola e da un colore.

69

Output di Lazarus 2

Lettera e Colore	Tipo di File
A	Archivio
C	Codice C
E	Elf
f	Sniffers
H	Html
I	Immagine
L	File di Log

70

Output di Lazarus 3

Lettera e Colore	Tipo di File
M	Mail
O	Null
P	Programmi
Q	Mailq
R	Rimosso
S	Lisp
T	Testo

71

Output di Lazarus 4

Lettera e Colore	Tipo di File
U	Non codificato
W	File password
X	Eseguibile
Z	Compresso
.	Binario
!	Sonoro

72

Analisi dell'Output

- L'analisi dell'output non è automatica.
- Dipende molto dall'abilità dell'analista e da come sono organizzati i file sul sistema.
- Impossibile effettuare una stima del tempo necessario a un'analisi.



73

Suggerimenti

- Ricerca nei file di chiavi specifiche. Utilizzo del tool **grep** o **egrep**.
- Uso del tool **less** per la visualizzazione di dati in formato binario.
- Utilizzo del tool **strings** per ricercare output leggibile.
- Utilizzo di un browser grafico per dati contenenti immagini.
- Tutti i tool utilizzati devono poter manipolare dati binari

74

Mactime

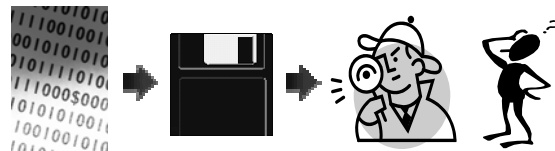
- Recupero dei dati temporali.
- Usato di solito in congiunzione con Graverobber.
- Indispensabile per l'analisi dei sistemi, specie in ambito forense.



75

Informazioni contenute in un file

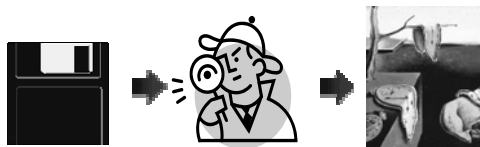
- Per ottenere tutte le informazioni possibili dai file di un sistema corrotto è sufficiente recuperarli, riconoscerli e aprirli?



76

Informazioni contenute in un file

- Assolutamente no! Manipolando un file se ne alterano le informazioni temporali, ovvero la data di creazione, ultimo accesso e ultima modifica!



77

MACtimes

- Le informazioni temporali di un file sono chiamate "MACtimes".
- MACtimes = Modification, Access, Inode change timestamps.
- I MACtimes sono estremamente volatili.
- Sono molto importanti in ambito forense.



78

Mactime: Utilizzo

- Schema di utilizzo:

```
perl mactime -d <directory> <data: dd/mm/yyyy> <opzioni>
```

- Opzioni:

- d : directory da analizzare
- B : scrive le informazioni sul filesystem in un file *body*
- b : produce output in HTML anziché in ASCII
- R : analizza ricorsivamente le sottodirectory
- y : le date in output sono in formato *yyyy/mm/dd*

79

Mactime : esempio

```
Shell n.2 - Konsole
Sessione Modifica Visualizza Segnalibri Impostazioni Aiuto
linux:~ # cd ..
linux:~ # cd home
linux:/home # cd enalot
linux:/home/enalot # cd tct-1.15
linux:/home/enalot/tct-1.15 # perl ./bin/mactime -d /home/enalot -h 9/27/2005
```

Esempio stringa di esecuzione per mactime.

80

Mactime: esempio

```

Date
-----
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/TRE_COSMOS_TOOLKIT.ppt
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/ocicad4d-usermode-0.11
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/boah_history
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/oft
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/TRE_COSMOS_TOOLKIT.ppt
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/OpisOffLine-orig1.1
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/FontDeLectioe-1
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/ICAWorker1ty
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/operation-scene
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/boah_history
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/Fontc
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/14000011e
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/ctandromer
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/aweeccc
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/ocicad4d-usermode-0.11
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/4de
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/gpib11c.html
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/akel
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/aweeccc
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/OpisOffLine-orig1.1
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/Desktop
Sep 28 05 19:11:28 143872 m... ..user:~#x enalot /home/enalot/comp

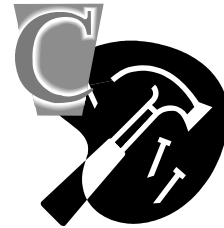
```

Esempio di output di mactime in formato html.

81

The C-Tools

- Suite di piccole applicazioni per l'analisi del sistema.
- Usati di solito in congiunzione con Graverobber.
- [ILS](#)
- [PCAT](#)
- [ICAT](#)
- [FILE](#)
- [MD5](#)



82

ILS

- ILS permette di leggere nella periferica richiesta e lista le informazioni sugli *inode*.
- Di default prende in considerazione solo gli *inode* di file eliminati.

83

ILS: Utilizzo

- Schema di utilizzo:
`./ils -f <tipo filesystem> periferica [inizio-fine] <opzioni>`
- Argomenti
periferica: file che contiene l'immagine del filesystem
-f: specifica tipo di sistema operativo. *ffs* per unix, *ext2fs* per linux.
[inizio - fine]: definiscono il raggio entro il quale effettuare la ricerca.

84

PCAT

- PCAT si aggancia al processo specificato dall'utente e manda la memoria ad esso associata in output.
- Per default, PCAT salta tutti i "buchi" nello spazio degli indirizzi del processo.
- Le informazioni assolute sul mappaggio sono perse.

85

PCAT: Utilizzo

- Schema di utilizzo

```
./ pcat -m<mapfile> process_ID <opzioni>
```

- Argomenti

process_ID : numero che identifica il processo rispetto al sistema operativo.

mapfile : nome del file nel quale scrivere l'output.

86

ICAT

- Questo tool viene usato per copiare file locati in *inode* precisi.
- L'utente dà in input il nome del disco fisso e l'*inode number*.
- ICAT restituisce il file associato all'*inode* sullo standard output.

87

ICAT: Utilizzo

- Schema di utilizzo

```
./ icat -f<fstype> device <opzioni>
```

- Argomenti

device : nome che identifica un disco fisso rispetto al sistema operativo.

fstype : specifica il tipo del file system. I valori di default sono *nfs* per Unix e *ext2fs* per Linux.

88

File

- File viene utilizzato per determinare il tipo del file che gli viene fornito in input.
- Esegue tre tipi di test, in ordine:
 - 1) **Filesystem test**
 - 2) **Magic Number test**
 - 3) **Language test**

89

Test

- Il primo test che ha esito positivo determina un output.
- Il tipo ottenuto tipicamente conterrà una di queste tre parole:
 - **text**
 - **executable**
 - **data**

90

Filesystem Test

- Esegue prima la system call *stat* e poi ne esamina il risultato.
- *File* effettua dei controlli per vedere se è un file vuoto o se è un tipo che conosce il sistema (file definiti in *sys/stat.h*).

91

Magic Number Test

- Cerca file che corrispondono a un formato.
- Alcuni tipi di file sono identificati da un numero.
- *File* cerca questo numero e, se lo trova, restituisce il file corrispondente



92

Language Test

- Questo test esamina le stringhe contenute nei primi blocchi del file.
- Cerca corrispondenza tra una di queste stringhe e una keyword (contenute in *names.h*) può dare in output *ascii text* o *data*.

93

File: Utilizzo

- Schema di utilizzo
`./file <files> -f<nomefile> -m<magicfiles>`
- Argomenti
files : nomi dei file dei quali si vuole sapere il tipo.
nomefile : nome di un file dal quale leggere i nomi dei file da analizzare.
magicfiles : lista alternativa di files che contengono magic numbers.

94

MD5

- Permette di elaborare firme MD5.
- Vengono dati in input i nomi dei file per i quali desideriamo una firma MD5.
- MD5 restituisce in output una singola riga per ogni file contenente la firma MD5 ed il nome del file separati da uno spazio.

95

MD5: Utilizzo

- Schema di utilizzo
`./md5 <files...>`
- Argomenti
files : nomi dei file per i quali si vuole computare la firma MD5.

96