VERIFICA DI PASSWORD:

Realizzato da:
Iodice Elvira, Mercogliano Roberta, Perropane Viviana, Scala Liberina

Sicurezza dei sistemi informatici: le password

- L'identificazione e l'autenticazione sono il processo di riconoscimento e verifica degli utenti.
  - L'identificazione è l'asserzione della propria identità.
  - L'autenticazione è la dimostrazione della propria identità.
  - L'autorizzazione accorda determinati privilegi ad una certa identità.

Sicurezza dei sistemi informatici: le password

- Per autenticare un utente ci sono tre categorie di elementi da cui un sistema di autenticazione può dipendere:
  - qualcosa che l' utente sa:
    password, Personal Identification Number, passphrase
  - qualcosa che l' utente ha:
    token, smart card, certificato, generatore di password one-time
  - qualcosa che l' utente è:
    impronte digitali, retina, voce, altri dati biometrici

Sicurezza dei sistemi informatici: le password

- Le password sono lo strumento basilare per l'autenticazione.
- Evitano intrusioni indesiderate nel proprio sistema.
- La scelta di buone password è un elemento fondamentale della configurazione del sistema.

Sicurezza dei sistemi informatici: le password

- Alcuni criteri da seguire per creare delle password:
  - Non utilizzare variazioni del proprio nome o username.
  - Non utilizzare parole di senso compiuto anche se precedute o seguite da numeri o simboli.
  - Non utilizzare nomi di alcun genere.
  - Non utilizzare una serie di lettere o numeri contigui sulla tastiera come "qwertyu".

Sicurezza dei sistemi informatici: le password

- Un consiglio per scegliere buone password è considerare la prima lettera di ogni parola di una frase.
  - Hb2Ci1m *Ho bruciato 200 CD in 1 mese*
- In questo modo le password sembrano incomprensibili ma sono facili da ricordare.
- La modifica periodica è sempre una buona idea.

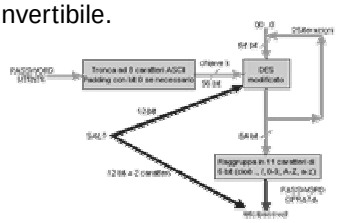
File delle password

- Nelle vecchie versioni di Unix le informazioni relative agli account venivano registrate nel file /etc/passwd.
- Il file era visibile a tutti.
- Solo root poteva modificarlo.
- Le righe erano del tipo:

  username:password:id_utente:id_gruppo:descrizione: home directory:shell

Cifratura delle password: Crypt()

- Le password vengono cifrate utilizzando la funzione *crypt(key,salt)* non invertibile.



Shadow Password

- Un cracker che ha in possesso il file delle password ha molte più possibilità di intromettersi.
- Una difesa supplementare consiste nell'utilizzare il sistema shadow:
  - si sostituisce una 'x' alla password cifrata nel file "passwd".
  - il file "shadow" contiene i cifrati ma è accessibile solo da root.

Attacchi alle password

- Un attacco alle password è un tentativo di violazione dei meccanismi di autenticazione.
- Un cracker che scopre una password può accedere a tutte le risorse ed esercitare tutti i diritti di quell'utente su quel sistema.
- Gli attacchi alle password sono di due tipi:
  - Attacchi di dizionario
  - Attacchi di forza bruta

Attacchi alle password: attacchi di forza bruta

- Provano tutte le combinazioni di un insieme di caratteri.
- Efficaci.
- Poco efficienti.
  - Per una password di 8 caratteri e un insieme di 52 caratteri disponibili (26 lettere dell'alfabeto, 10 cifre e 16 simboli) occorre esaminare 8^{52} combinazioni, cioè 302.231.454903.657.000.000.000 tentativi

Attacchi alle password: attacchi di dizionario

- Provano come password tutte le parole predefinite in un dizionario.
- Più veloci rispetto agli attacchi di forza bruta.
- Il successo dipende dalla bontà del dizionario.

Attacchi di dizionario

- Di solito le password sono memorizzate in forma cifrata.
  - Se si hanno a disposizione i cifrati delle password questi verranno confrontati con le parole del dizionario alle quali è stata applicata la stessa funzione di cifratura.
    - John the Ripper, Crack.
  - Se sono consentiti ripetuti tentativi di accesso è possibile fornire come password una delle parole del dizionario senza applicare alcuna funzione di cifratura.
    - Webcracker, Zip cracker

Attacchi di dizionario

- Il dizionario dovrebbe contenere:
  - Parole del linguaggio corrente
  - Termini tecnici riguardanti il computer e la rete
  - Eventi culturali di massa
  - Date
- Si può pensare di aggiungere numeri e caratteri speciali all'inizio e/o alla fine di ogni parola.

Attacchi di dizionario

- L'efficienza di un attacco dipende da molteplici fattori:
  - numero di password da confrontare
  - algoritmo di cifratura, sua implementazione
  - tempo di esecuzione di una istruzione
  - numero di processori disponibili
- Per un file delle password con 2000 utenti e un dizionario di 500 Kbyte sono richiesti in media tre ore di esecuzione, variabili in base alla potenza dell' hardware.

Attacchi di dizionario: Webcracker

- Daniel Flam, 1999.
- Testa la vulnerabilità di siti web con accesso ristretto.
- È uno strumento di sicurezza: se Webcracker scopre una combinazione *id password,* qualunque cracker può farlo.
- Richiede di specificare:
  - URL
  - lista di user ID
  - opzionalmente, un file di possibili password

Attacchi di dizionario: Webcracker

- Utilizza file di combinazioni *ID/password:*
  - Mickey Mouse
- Converte ID e password in maiuscolo e in minuscolo.
- Utilizza le variabili di sostituzione
  - %USERID -corrente user ID-
  - %REVUID -user ID invertito-

Attacchi di dizionario: Webcracker

- Le variabili di sostituzione consentono di creare una lista di password ricavate dallo user ID.
  - %USERID1
  - %USERID%REVUID
  - 99%USERID99
- Se il corrente user ID è Mickey verranno provate combinazioni del tipo:
  - *Mickey /Mickey1,MickeyyekciM ,Mickey/99Mickey99*

Attacchi di dizionario: Zipcrack

- Paul Kocher, 1992.
- Cracker di password di archivi compressi con Pkzip v1.1.
- Costo:
  - $ 50 per uso personale
  - $ 500 per uso commerciale o governativo

Attacchi di dizionario: Pkcrack

- Risale al 1993 da autore sconosciuto.
- Cracker di password di archivi .zip compressi con Pkzip v2.04.
- Il pacchetto è gratuito.

Attacchi di dizionario: FZC (Fast Zip Cracker)

- Fernando Papa Budzyn.
- Ultima versione 1.05, 1998.
- Maggiori probabilità di successo.
- Free.

Attacchi di dizionario: UZPC (Ultra Zip Password Cracker)

- Ivan Golubev.
- Più leggero e veloce di altri cracker di password di archivi zippati.
- Attacco template: ibrido tra dizionario e forza bruta.
- Ottimizzato per:
  - Processore Pentium
  - S.O. Win 95/98/NT
- Richiede 100 Kbyte di spazio su disco.
- Free.

Attacchi di dizionario: John the Ripper

- Richiede un file delle password e opzionalmente una wordlist su cui verranno applicate delle regole di trasformazione.
- Supporta diversi algoritmi di cifratura:
  - DES, DES doppio
  - DES esteso di BSDI
  - MD5 di free BSD
  - Blowfish di open BSD
- Free.

Attacchi di dizionario: Crack

- Crack v5.0, Alec Muffett, 1996.
- Verifica eventuali insicurezze nel file delle password.
- Richiede l'accesso al file delle password.

Crack

- Consente di ampliare il suo dizionario.
- Consente di ripartire il carico di lavoro tra più hosts.
- Permette di verificare file di password di più hosts.
- Permette di avvertire con una e-mail gli utenti con password debole.

Crack

- Requisiti richiesti:
  - S.O. Unix-like Linux, FreeBSD, Ultrix, NetBSd, OSF
  - Compilatore C
  - Spazio sufficiente su disco
  - Molto tempo di CPU
  - Permesso da parte dell'amministratore di sistema
  - Privilegi di root

Crack su sistemi Linux

- Crack opera nel seguente modo:
  - per ogni "parola" del dizionario e per ogni "salting" nel file delle password:
    - applica crypt("parola","salting").
    - confronta il risultato col cifrato nel file delle password.
    - se coincidono è stata individuata una password debole.
  - Nel pacchetto Crack è inclusa la libreria *libdes* di Eric Young che contiene un'implementazione elegante e veloce della funzione standard crypt().

Crack e Cracklib

- Funzione preventiva è il controllo della complessità della password con un vocabolario nel momento in cui viene inserita.
- Occorre procurarsi la libreria *Cracklib* di Alec Muffett fornendo un vocabolario non troppo ampio.
- L'ideale è usare un piccolo vocabolario con Cracklib e periodicamente lanciare Crack con un vocabolario più ampio.

Crack: il dizionario

- Crack raggruppa il dizionario in tre gruppi:
  - più probabili - gruppo 1
  - meno probabili - gruppo 2
  - minimamente probabili - gruppo 3
- Si specificano i file di parole che compongono il gruppo.

Crack: il dizionario

- Dizionari aggiuntivi:
  - gecos: contiene le parole del file delle password.
  - gcperm: permutazioni e combinazioni di parole del file delle password.
- Si sconsiglia di utilizzare come password parole presenti nel file delle password o parole ricavate da permutazioni di queste.

Crack: i dizionari

- E' possibile ampliare il dizionario aggiungendo nuovi file di parole.
- File di parole aggiunti da noi:
  - Dizionario di nomi italiani
  - Dizionario di date nel seguente formato:
    - gg/mm/aa
    - gg-mm-aa
    - mm/gg/aa
    - ggmesaa
    - ddmonyy

Crack: le regole

- E' possibile generare nuove parole a partire da quelle dei dizionari applicando delle regole di trasformazione.
- Crack leggerà il gruppo di dizionario e creerà da esso un insieme di possibili password applicando le regole specificate .

Crack: le regole

- Esistono molte regole predefinite.
- E' possibile specificare nuove combinazioni di regole.
- E' possibile implementare nuove regole.

Crack: le regole

- Abbreviazioni per classi di caratteri usati nelle regole:
  - vocale: ?v
  - consonante: ?c
  - minuscolo : ?l
  - maiuscolo: ?u
  - numeri: ?d
  - alfabetici: ?a
  - alfanumerici: ?x
  - spazi bianchi: ?w
  - punteggiatura: ?p
  - simboli: ?s

Crack: le regole

- Lista di regole:
  - **restart : ***
    - Resetta il buffer a uno stato iniziale.
  - **prepend : ^X**
    - Inserisce il carattere X all'inizio della parola nel buffer.
  - **append : $X**
    - Inserisce il carattere X alla fine della parola nel buffer.

Crack: le regole

- **dfirst : [**
  - Cancella il primo carattere dalla parola nel buffer.
- **dlast :]**
  - Cancella l'ultimo carattere dalla parola nel buffer.
- **reverse : r**
  - Prende la parola nel buffer e ne fa il reverse.
- **duplicate : d**
  - Prende la parola nel buffer e ne appende una copia di se stessa.

Crack: le regole

- **reflect : f**
  - Prende la parola nel buffer e ne appende il reverse.
- **uppercase : u**
  - Prende la parola nel buffer e rende ogni lettera maiuscola.
- **lowercase : l**
  - Prende la parola nel buffer e rende ogni lettera minuscola.

Crack: le regole

- **capitalise : c**
  - Prende la parola nel buffer, rende maiuscolo il primo carattere e minuscolo il resto.
- **ncapital : C**
  - Prende la parola nel buffer, rende minuscolo il primo carattere e maiuscolo il resto.
- **pluralise : p**
  - Prende la parola nel buffer e ne fa il plurale in inglese.

Crack: le regole

- **togcase : t**
  - Prende una parola dal buffer e trasforma le minuscole in maiuscolo e viceversa.
- **lt : <N**
  - Rifiuta la parola a meno che la sua lunghezza non sia minore di n caratteri.
- **gt : >N**
  - Rifiuta la parola a meno che la sua lunghezza non sia maggiore di n caratteri.

Crack: le regole

- **match : /X** or **/?C**
  - Rifiuta la parola a meno che non contenga il carattere X, o un carattere membro della classe C.
- **not : !X** or **!?C**
  - Rifiuta la parola se contiene il carattere X, o un carattere membro della classe C.
- **mfirst : (X** or **(?C**
  - Rifiuta la parola a meno che il primo carattere non sia X, o un membro della classe C.

Crack: le regole

- **mlast :)X** or **)?C**
  - Rifiuta la parola a meno che l'ultimo carattere non sia X, o un membro della classe C.
- **equals : =NX** or **=N?C**
  - Rifiuta la parola a meno che il carattere N non sia X, o un membro della classe C.
- **atleast : %NX** or **%N?C**
  - Rifiuta la parola a meno che non contenga almeno N istanze del carattere X, o dei membri della classe C.

Crack: le regole

- **substitute : sXY** or **s?CY**
  - Sostituisce tutte le istanze di X, o dei membri della classe C, con il carattere Y.
- **extract : xNM**
  - Estrae la sottostringa di lunghezza M, iniziando dalla posizione N, dalla parola, e scarta il resto.
- **insert : iNX**
  - Inserisce il carattere X alla posizione N, shiftando tutte le altre lettere verso destra.

Crack: le regole

- **overstrike : oNX**
  - Sovrascrive il carattere N con il carattere X.
- **purge : @X** or **@?C**
  - Rimuove tutte le istanze (o caratteri della classe C) dalla parola.
- **snip : 'N**
  - Tronca la parola alla lunghezza N.

Crack: Regole

- Esempio:
  - **/ese3/oso0u**
    - /e scarta le parole che non contengono il carattere "e".
    - se3 sostituisce il carattere "e" con "3".
    - /o scarta le parole che non contengono il carattere "o".
    - so0 sostituisce il carattere "o" con "0".
    - u trasforma tutta la parola in maiuscolo.
  - **Esempio** ⇨ **3S3MPI0**

Crack: Regole

- E' possibile specificare nuove combinazioni di regole:
  - **>5/asa@/sss$l**
    - >5 se la parola è >5 caratteri.
    - /a controlla se la parola contiene almeno una "a".
    - sa@ sostituisce le "a" con "@".
    - /s controlla se la parola contiene almeno una "s".
    - ss$ sostituisce la "s" con "$".
    - l trasforma tutta la parola in minuscolo.
  - **cassata** ⇨ **c@$$@t@**

Crack: Regole

- Regola implementata da noi:
  - **Kappa k:** sostituisce la prima occorrenza di "ch" o "cch" con "k" o "kk".
- Esempio:
  - kkkk sostituisce quattro occorrenze di "ch".
    - "Chicchirichi" viene scritta come "kikkiriki".

Crack: compressione dei dizionari

- I dizionari sono compressi usando la funzione DAWG (Directed Acyclic Word Graph).
- Rimuove le ridondanze.
- Comprime la wordlist in input del 50% circa.

Crack: compressione dei dizionari

- L'algoritmo è il seguente:
  - 1. Ordina la wordlist.
  - 2. Per ogni parola della wordlist:
    - 2.2. conta il numero di caratteri iniziali che la parola condivide con quella precedente.
    - 2.3. codifica questo numero come un carattere ASCII stampabile [0-9a-zA-Z] per valori compresi tra 0..61 (se il valore è maggiore di 61 stop).
    - 2.4. stampa questo carattere e la parte rimanente della parola.

Crack: compressione dei dizionari

- Esempio:

emulare
emulatore
emulazione
emulo
emulsionare

- Con DAWG verranno scritti 29 caratteri invece di 42

0emulare	La parola condivide 0 caratteri con la precedente
5tore	5 lettere in comune con la precedente
5zione	5 lettere in comune con la precedente
4o	4 lettere in comune con la precedente
4sionare	4 lettere in comune con la precedente

Crack: opzioni

- Il comando per eseguire Crack è
  - Crack [–opzioni] filepassword
- Lista completa delle opzioni:
  - **makeonly :** Crea l'eseguibile Crack.
  - **makedict :** Crea i gruppi di dizionari compressi.
  - **fgnd :** Esegue Crack in foreground.

Crack: opzioni

- **debug :** Visualizza l'esecuzione dello script Crack.
- **keep :** Previene la cancellazione del file temporaneo usato per memorizzare l'input.
- **nice N :** Esegue il cracker delle password con bassa priorità.

Crack: opzioni

- **recover :** Fa ripartire l'elaborazione se questa non era terminata normalmente.
- **mail :** Invia una e-mail all'utente a cui è stata scoperta la password.

Crack: opzioni

- **from N :** Inizia il cracking delle password dalla regola numero N.
- **network :** Ripartisce il carico di lavoro tra più host.

Crack: opzioni

- **fmt format :** Specifica il formato del file delle password da convertire nel formato SPF.
  - Per default format vale "trad" ma può essere specificato anche "bsd" se si usano sistemi FreeBSD e NetBSD.

Funzioni di cifratura: interfaccia ELCID

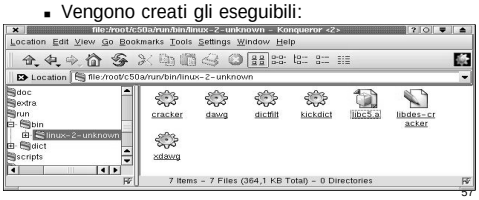
- External Library Crypt Interface Definition.
- crypt()
  - Unix, Linux, FreeBSD, NetBSD
  - Utilizza elcid.c.
  - Già configurato.
- MD5
  - FreeBSD, NetBSD
  - Utilizza elcid.c,bsd.
  - Da sostituire ad elcid.c.

Funzioni di cifratura: interfaccia ELCID

- crypt16()
  - Ultrix, Digital Unix Machine
  - Utilizza libc-crypt.
  - Si scarica da un sito ftp GNU.
  - Si spacchetta tra i sorgenti di Crack creando una directory crypt.
  - Si pone la costante PLAINTEXTSIZE definita in elcid.c a 16.

Esempio: Esecuzione di Crack

- Per compilare il programma Crack sull'host altafini:
  - altafini>c50a# Crack –makeonly
  - Vengono creati gli eseguibili:



Esempio: Esecuzione di Crack

- Per creare i gruppi di dizionari compressi:
  - altafini>c50a# Crack –makedict

Esempio: Esecuzione di Crack

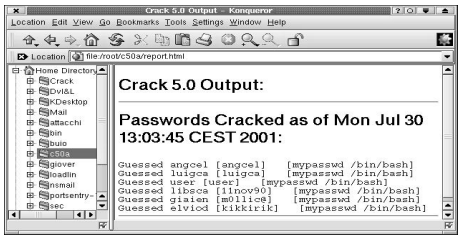
- Per fondere i file *etc/passwd* e *etc/shadow* usare lo script *shadow.sv*
- L'output è rediretto nel file mypasswd.
  - altafini>c50a# scripts/shadow.sv>mypasswd
- Per eseguire Crack:
  - altafini>c50a# Crack mypasswd

Esempio: Esecuzione di Crack

- Per visualizzare l'output eseguire ripetutamente lo script *Reporter*.
- L'output può essere rediretto in un file html.
  - altafini>c50a# Reporter –quiet -html>Report.html

Esempio: Esecuzione di Crack

- Visualizzazione del file Report.html:



Esempio: Esecuzione di Crack

- Per terminare l'esecuzione di Crack:
  - altafini>c50a# scripts/plaster
- Per rimuovere i file di scratch:
  - altafini>c50a# make tidy

Esempio: Crack -recover

- Per recuperare una sessione terminata accidentalmente conservare il file *run/Daltafini.N*, contenente le password da scoprire e i risultati dell'esecuzione, in un file temporaneo:
  - altafini>c50a# mv run/Daltafini.4466 run/tempfile

Esempio: Crack -recover

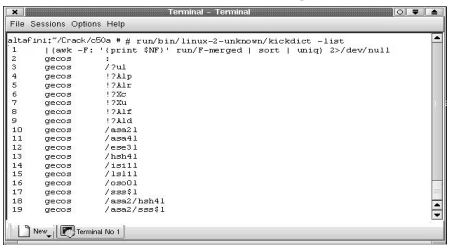
- Eseguire Crack con l'opzione *-recover* utilizzando come file delle password il file tempfile nel formato standard delle password.
  - altafini>c50a# Crack -recover -fmt spf run/tempfile
- L'esecuzione riprenderà dal punto in cui è stata interrotta.

Esempio: Crack -from

- Per eseguire Crack dalla regola *N* :
  - altafini>c50a# Crack -from 13 mypasswd
- Per visualizzare il numero associato ad ogni regola:
  - altafini>c50a# run/bin/linux-2-unknown/kickdict –list
  - Linux-2-unknown è l'architettura che si sta utilizzando

Esempio: Crack -from

- Visualizzazione delle regole:

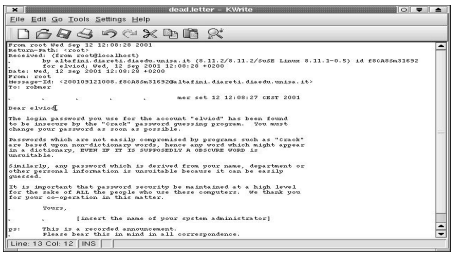


Esempio: Crack -mail

- Per eseguire Crack con l'opzione *–mail*:
  - altafini>c50a# Crack – mail mypasswd
- Ogni utente a cui è stata scoperta la password riceverà una e-mail di avvertimento.

Esempio: Crack -mail

- E-mail di avvertimento:

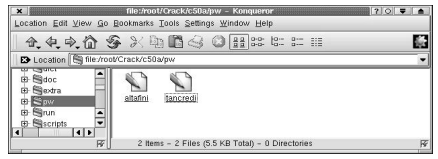


Esempio: Crack eseguito su diversi file delle password.

- Per eseguire Crack sul file delle password dell'host *tancredi* e *altafini* bisogna memorizzare tali file in *pw/tancredi* e *pw/altafini*.
- I file tancredi e altafini sono stati trasformati in un formato che Crack può leggere con lo scripts *shadmrg.sv*.

Esempio: Crack eseguito su diversi file delle password.

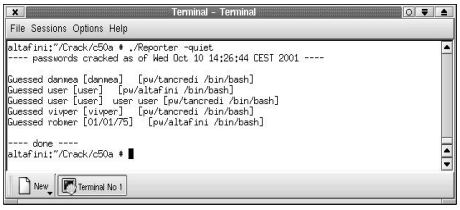
- Directory pw:



- Per eseguire Crack su più file delle password:
  - altafini>c50a# Crack pw/*

Esempio: Crack eseguito su diversi file delle password.

- Visualizzazione del Reporter:



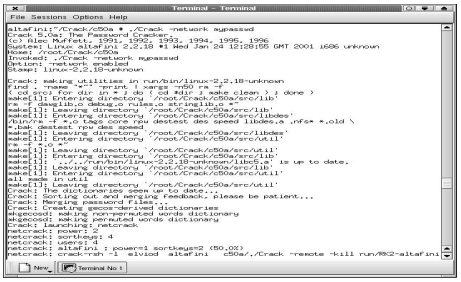
Esempio: Crack -network

- Crack ripartisce il carico di lavoro tra l'utente *vivper* sull'host remoto *tancredi* e l'utente *elviod* sull'host *altafini*.
  - Gli utenti vivper e elviod inseriscono nel file *.rhosts* la riga:
    - altafini.diareti.diaedu.unisa.it root
  - L'utente root sull'host altafini configura il file *c50a/conf/network.conf* inserendo le righe:
    - tancredi:1:n:vivper:c50a/.
    - altafini:1:n:elviod:c50a/.

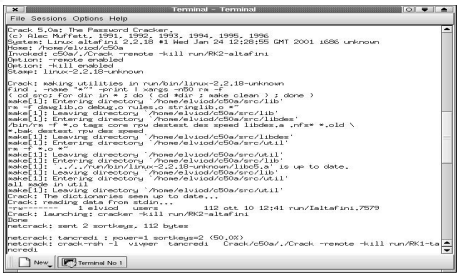
Esempio: Crack -network

- Per eseguire Crack con l'opzione *-network*:
  - altafini>c50a# Crack – network mypasswd
- Crack viene compilato su entrambi gli utenti.
- Crack viene automaticamente invocato attraverso il comando *rsh*
  - Crack -remote -kill run/RK1 -tancredi
  - Crack -remote -kill run/RK2 -altafini

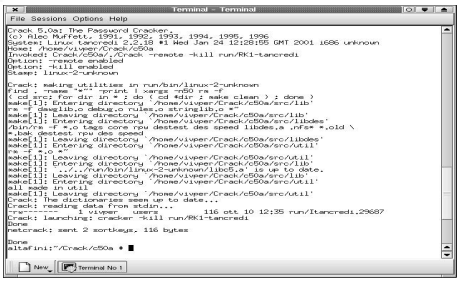
Esempio: Crack -network



Esempio: Crack -network

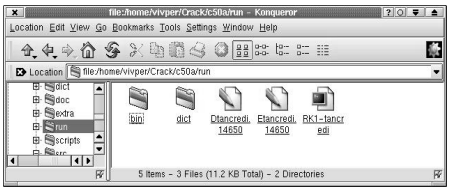


Esempio: Crack -network



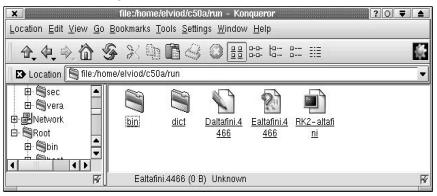
Esempio: Crack -network

- Durante l'esecuzione Crack aggiunge alcuni file nella directory *c50a/run* .
- Directory run su vivper



Esempio: Crack -network

- Directory run su elviod



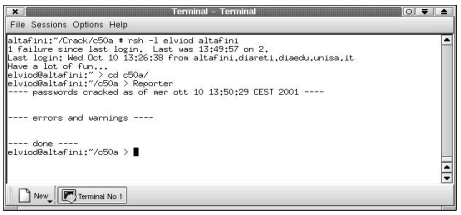
- RK1-tancredi e RK2-altafini sono script che uccidono i processi *cracker* e sono invocati da scripts/plaster eseguito da root.

Esempio: Crack -network

- I processi cracker su elviod e vivper sono mostrati con il comando :
  - *ps -fe* oppure *ps -aux*
- Per visualizzare i risultati dell'esecuzione dall'host *altafini*, root deve lanciare lo script Reporter su elviod e vivper aprendo una shell remota.

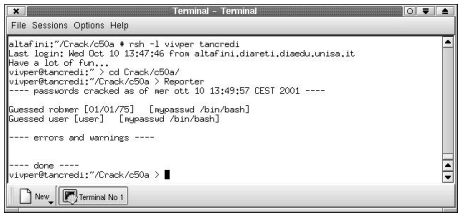
Esempio: Crack -network

- Reporter su elviod



Esempio: Crack -network

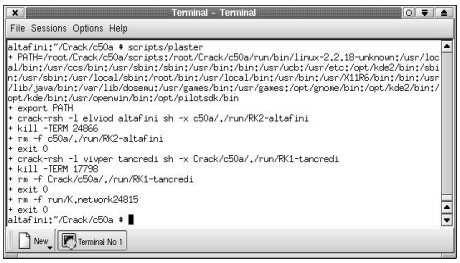
- Reporter su vivper



Esempio: Crack -network

- Per terminare l'esecuzione di Crack l'utente root deve invocare:
  - altafini>c50a# scripts/plaster
- I processi *cracker* in esecuzione su elviod e vivper vengono uccisi.
- I file RK1-tancredi e RK2-altafini vengono cancellati.

Esempio: Crack -network



Esempio: Crack -network

- Per eliminare i file di scratch eseguire il comando *make tidy* in remoto.
- Unico file conservato è *c50a/run/F-merged*, contenente coppie testo in chiaro e testo cifrato delle password scoperte.

Esempio: Crack -network