CyberChallengeIT 2019
Università degli studi di Pisa

Funzioni Hash e Steganografia

JohnTheRipper

JohnTheRipper è un tool a linea di comando utile per crackare hash di diversi algoritmi (MD5, SHA1…)

Potete installarlo con aptitude

```
apt-get install aptitude
sudo aptitude install john
```

Con ‘john --help’ potete vedere tutte le sue opzioni.

JohnTheRipper ha 2 principali modalità di attacco:

JohnTheRipper – Wordlist Attack

L'idea è abbastanza intuitiva. Fornite a john un dizionario (wordlist) di parole che voi vi aspettiate possano aver generato le hash che state cercando di crackare.
Le parole devono essere in un file di testo, una per riga.
È consigliabile ordinarle alfabeticamente.
Esistono molti dizionari pronti all'uso, il più famoso è rockyou.txt .

È possibile provare anche delle varianti delle parole, dette «mangled», storpiate.
Esempio di parola storpiata: topolino → t0p0l1n0

JohnTheRipper – Wordlist Attack

Non c'è bisogno di inserire tutte le possibili varianti storpiate delle parole all'interno della wordlist. Tramite un set di regole (per john, ovviamente, RULES) potete decidere quali storpiature applicare al vostro dizionario.

Potete decidere di fare più set di regole, inserendole alla fine del file '/etc/john/john.conf'.

John ha dei set predefiniti di regole, la più potente dei quali è la JUMBO rule.

JohnTheRipper – Wordlist Attack

```
[List.Rules:Example]
cAz"[0-9]"
```

Questa regola rende maiuscola la prima lettera delle parola all'interno della wordlist, e «postfigge» (append) un numero alla stringa così generata. In questo modo, se nella wordlist ci fosse soltanto la parola «ciao», john testerebbe:

- ciao
- Ciao0
- Ciao1
- Ciao2
- Ciao3
- Ciao4
- Ciao5
- Ciao6
- Ciao7
- Ciao8
- Ciao9

JohnTheRipper – Incremental Attack

Questa è la modalità bruteforce, proverà ogni singola combinazione del set di caratteri fornito. Per usare questa modalità bisogna definire dei parametri (lunghezza massima della chiave, set di caratteri consentiti ecc…). Queste definizioni si trovano ne file di configurazione, sotto la voce "[Incremental:XXXX]" dove al posto delle XXXX dovete inserire una stringa identificativa, che sarà il nome col quale invocherete la modalità incrementale.

Esistono diverse modalità pre-impostate:

- "ASCII" (tutti i 95 caratteri stampabili ASCII)
- "Alnum" (tutti i 62 caratteri alfanumerici)
- "Alpha" (tutte le 52 lettere, maiuscole e minuscole)
- "LowerNum" (lettere minuscole e cifre, 36 caratteri totali)
- "UpperNum" (lettere maiuscole e cifre, 36 caratteri totali)
- "LowerSpace" (lettere minuscole più lo spazio, 27 caratteri totali)
- "Lower" (lettere minuscole, 26 caratteri totali)
- "Upper" (lettere maiuscole, 26 caratteri totali)
- "Digits" (solo cifre, 10 caratteri totali)

JohnTheRipper - sources

Cracking modes:
https://www.openwall.com/john/doc/MODES.shtml

Sintassi per la composizione delle regole:
https://www.openwall.com/john/doc/RULES.shtml

Esempi di utilizzo di JohnTheRipper:
https://www.openwall.com/john/doc/EXAMPLES.shtml

JohnTheRipper cheatsheet:
https://countuponsecurity.files.wordpress.com/2016/09/jtr-cheat-sheet.pdf

Introduzione alla creazione di regole personalizzate:
https://www.gracefulsecurity.com/custom-rules-for-john-the-ripper/

Esempio di 3 rule set (potete copiare ed incollare nel vostro john.conf):
https://www.gracefulsecurity.com/custom-rules-for-john-the-ripper-examples/

JohnTheRipper – script di “estrazione hash”

Nell’archivio scaricabile da github c’è la cartella run.

Al suo interno potete trovare numerosi script in python utili per estrarre le hash di password che proteggono file in diversi formati.

Questi script appaiono nel formato

<estensione>2john.py

Tra i più utili troviamo

zip2john.py
pdf2john.py
rar2john.py
7z2john.py

JohnTheRipper – Esempio cracking di Zip

Scaricate il file zip «may3» da telegram.

Da terminale:

```
zip2john may3.zip > hash_zip.txt
```

Scrive nel file hash_zip l'hash della password per sbloccare lo zip.

```
john hash_zip.txt > solution.txt
```

Esegue il cracking in modalità default, prima con la wordlist inclusa (molto piccola), poi in modalità incrementale ASCII. La stringa che genera quell'hash è salvata in solution.txt

Gli stessi passaggi possono essere fatti con altri tipi di file utilizzando altri script che trovate nella cartella run.

JohnTheRipper – Shadow file

Il file /etc/shadow contiene informazioni riguardanti gli utenti del sistema, uno per riga, la quale è composta da 9 campi separati dal carattere «duepunti» ':'
I campi indicano, nell'ordine:

1. Il nome utente.

2. La password crittografata, oppure un asterisco per indicare che non è possibile effettuare direttamente il login come quell'utente.

3. La data dell'ultima modifica della password, in giorni trascorsi dal 1º gennaio 1970.

4. Il minimo di giorni che devono trascorrere prima di poter modificare la password.

5. Il massimo di giorni che devono intercorrere tra una modifica e l'altra di una password.

6. I giorni di preavviso della scadenza della password.

7. I giorni dopo i quali una password scaduta comporta la disattivazione dell'account.

8. La data di scadenza della password, come numero di giorni dal 1º gennaio 1970.

9. Un campo riservato per usi futuri.

JohnTheRipper – Esempio cracking di password unix

Supponete di avere accesso ad un computer condiviso con altri utenti.
Potete utilizzare john per trovare le loro password!

Da terminale (aggiungiamo un utente fasullo):

```
useradd -r fantoccio
passwd fantoccio
```

Unix vi chiederà di inserire una password, vi suggerisco di inserirne una facile, come «hello» oppure «1612», per non perdere troppo tempo in questa prova.

```
john /etc/shadow > usersPwd.txt
```

Nel file usersPwd.txt potrete trovare le password che john è riuscito a crackare.

Il seguente comando mostra le hash presenti in «file.txt» già crackate:

```
john file.txt --show
```

Hashcat

Hashcat è un famoso software utilizzato per crackare diversi tipi di hash, supporta esecuzione su GPU ed è disponibile su diversi sistemi operativi.
Installazione: `sudo aptitude install hashcat`
Le opzioni più importanti per hashcat sono:

-a → attack mode, 3 per bruteforce

-m → hash type, 0 per md5

--force → per forzare l'esecuzione su alcune piattaforme

-w → per settare le performance 1 basse, 4 massime

-o → specifica il file di output dove memorizzare i risultati.

-i → attacco incrementale, seguiti da parametri:
--increment-min <min> --increment-max <max>

Hashcat - maschere

Oltre alle opzioni viste, hashcat ha un'altra feature molto utile, anche se spesso poco utilizzabile: le maschere.

Sapere la struttura di una password, può essere molto utile. Per esempio, provate ad hashare con md5 la password jack1970

Ed invocate hashcat con il comando

```
hashcat -m 0 -a 3 -w 4 --force e4340d4fdb5705d65b8284802b683727
```

Quanto ci mette? Troppo…

Adesso provate ad invocare lo stesso comando, ma inserendo alla fine la seguente stringa:

```
?l?l?l?l?d?d?d?d
```

Il cracking ci impiega sensibilmente meno! In 3 minuti ha trovato la soluzione!

Hashcat - charset

Le maschere funzionano con set di caratteri:

?l → abcdefghijklmnopqrstuvwxyz
?u → ABCDEFGHIJKLMNOPQRSTUVWXYZ
?d → 0123456789
?h → 0123456789abcdef
?H → 0123456789ABCDEF
?s → !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
?a → ?l?u?d?s
?b → 0x00 - 0xff

Potete anche definire i vostri custom character set (da 1 a 4). Se adesso provate a fare l'hash della stringa «Jack1970», potete utilizzare il seguente comando:

```
hashcat <varie opzioni ed hash> -4 ?u?l ?4?l?l?l?d?d?d?d
```

Per vedere tutte le password crackate da hashcat:

```
cat .hashcat/hashcat.potfile
```

Fcrackzip

Fcrackzip è un tool molto semplice ed intuitivo per crackare archivi zip protetti da password. sudoLink: https://github.com/hyc/fcrackzip

Le varie opzioni di Fcrackzip sono:

-b	→	use brute force algorithm
-D	→	use a dictionary
-c	→	use characters from charset (aA1!:)
-V	→	sanity-check the algortihm
-v	→	be more verbose
-p	→	use string as initial password/file
-l	→	check password with length min to max
-u	→	use unzip to weed out wrong passwords
file...		the zipfiles to crack

Potete testare Fcrackzip con il seguente comando

```
fcrackzip -b -c '1' -l 1-6 -u may3.zip
```

Corkami/collisions

Corkami è un insieme di script in python che permette di modificare leggermente due file in modo che generino entrambi lo stesso hash MD5 (collisione).

Potete scaricare Corkami da:
https://github.com/corkami/collisions

E trovate gli script in
collisions-master/scripts

Corkami/collisions funzionamento

Scaricate da internet due immagini qualsiasi in formato png.
Controllatene l'hash MD5. A meno di una straordinaria coincidenza, dovrebbero tornare diversi!

Adesso eseguite il comando:

```
python png.py file1.png file2.png
```

Il programma dovrebbe generare due file «collision1.png» e «collision2.png»

Se controllate le hash di queste due immagini risultanti, vedrete che saranno uguali!

Potete eseguire lo stesso procedimento con file in formato jpg, pdf, ed mp4.

STEGANOGRAFIA

Stegsolve

Stegsolve è un'applicazione java utile per analizzare i «piani» di un'immagine in vari formati: png, jpeg, gif, bitmap…

Per scaricare l'applicazione:

```
wget http://www.caesum.com/handbook/Stegsolve.jar -O stegsolve.jar
```

Per renderla eseguibile:

```
chmod +x stegsolve.jar
```

Per Eseguirla:

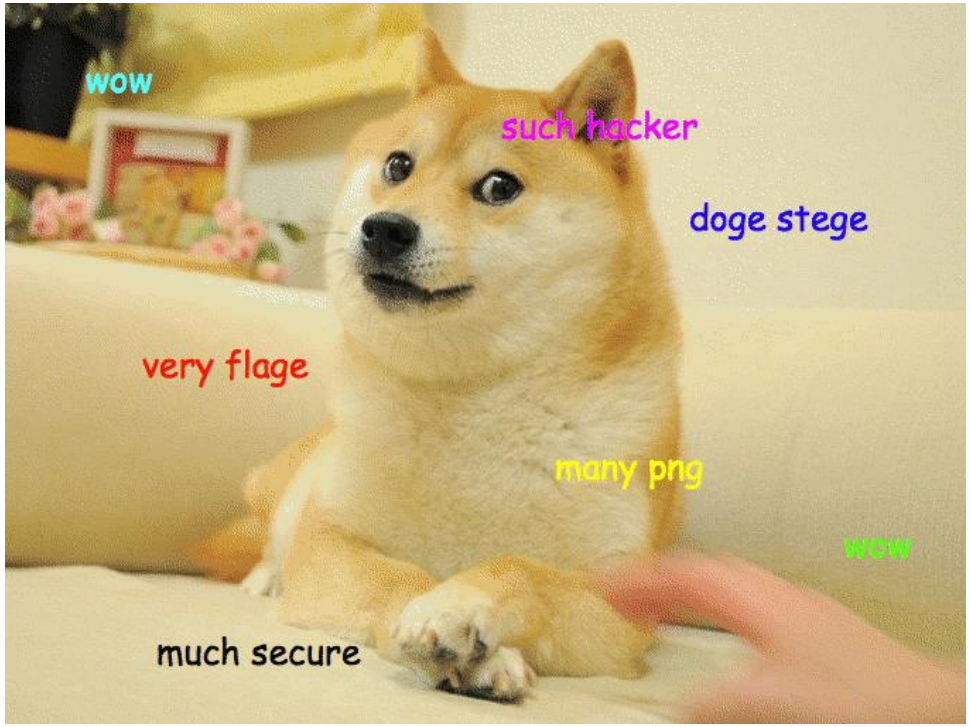
```
java -jar stegsolve.jar
```

GIMP

GIMP fa, tra le altre cose, un lavoro molto simile a stegsolve.
Per installare GIMP:
apt-get install gimp

Utilizzo:

- Nella cartella doge c'è doge.png
- Apritelo in GIMP.
- Navigate in colors→ set colormap
- Provatele tutte finché non trovate la flag.



Steghide

Steghide è un'applicazione a linea di comando tramite la quale si possono nascondere informazioni in immagini e tracce audio dei seguenti formati JPEG,JPG,BMP,WAV,AU.

Installazione:

```
apt-get install steghide
```

Informazioni:

```
steghide --help
```

Come esempio, provate a scaricare un'immagine e create un file di testo «segreto.txt».
Per creare un'immagine steganografica:

```
steghide embed –cf immagine.jpg –ef segreto.txt
```

Il programma vi chiederà anche di inserire una password.
ATTENZIONE! L'IMMAGINE VERRA' MODIFICATA!

Per estrarre un segreto:

```
steghide extract -sf immagine.jpg
```

Zsteg

Zsteg è uno strumento per scoprire informazioni nascoste in PNG e BMP.

Si può scaricare zsteg da https://github.com/zed-0xff/zsteg
Ed installare:

```
gem install zsteg
```

Per informazioni su zsteg e le sue opzioni:

```
zsteg -h
```

Per eseguire zsteg basta semplicemente invocarlo con il nome del file da analizzare:

```
zsteg flower_rgb3.png
```

Se utilizzate il comando di prima con l'opzione «–a» troverete più risultati. Zsteg –a infatti utilizza tutti i metodi conosciuti a prescindere dai risultati che ottiene.

Binwalk

Binwalk è uno strumento di steganalisi che rileva archivi o altri file «embeddati», inclusi, in altri file.

Lo potete installare con:

```
sudo aptitude install binwalk
```

Con il comando

```
binwalk «nomefile»
```

Il programma elenca i contenuti nascosti nel file

Con il comando

```
binwalk -e «nomefile»
```

Il programma cerca di estrarre i contenuti nascosti.

WavSteg

WavSteg è uno script python che permette di nascondere (e recuperare) file nei bit meno significativi di una traccia audio in formato .wav. Attenzione: richiede python3!

Potete scaricarlo da:
https://github.com/ragibson/Steganography

Con il comando WavSteg.py –help vedrete:

```
-h, --hide          To hide data in a sound file
-r, --recover       To recover data from a sound file
-s, --sound=        Path to a .wav file
-f, --file=         Path to a file to hide in the sound file
-o, --output=       Path to an output file
-n, --LSBs=         How many LSBs to use
-b, --bytes=        How many bytes to recover from the sound file
--help              Display this message
```

WavSteg

Quindi, per nascondere il file «secret.txt» in «sound.wav» con una profondità di 1 bit:

```
python3 WavSteg.py -h -s sound.wav -f secret.txt -o hidden.wav -n 1
```

Mentre, per recuperare il segreto:

```
WavSteg.py -r -s hidden.wav -n 1 -b <sizeof(secret.txt)>
```

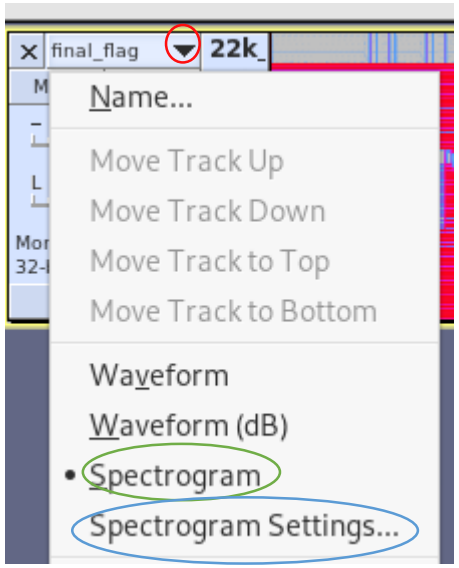
Il destinatario del messaggio segreto deve conoscere la dimensione esatta del file che è stato nascosto, e questa è una limitazione.
Una semplice strategia utile per scambiarsi file di testo è inserire come primo dato nascosto la dimensione del file su un numero predeterminato di byte.
In questo modo il destinatario recupera con una prima chiamata a WavSteg la dimensione del file nascosto, e con una seconda chiamata recupera effettivamente il file.

Audacity

Audacity è un software di analisi di tracce audio, utilizzato in steganografia per la capacità di analizzare lo spettro di frequenze di tali tracce.

Installazione
apt-get install audacity

Quando aprite un file vi appare una riga che vi «mostra» il suono del file.
Cliccando sulla freccia accanto al nome del file
vi si apre un menù a tendina.
L'opzione più utile è lo spettrogramma,
e le sue relative impostazioni.



DeepSound

DeepSound è un software che permette di individuare ed estrarre contenuti nascosti all'interno di tracce audio.
L'unica distribuzione disponibile è per windows:
http://jpinsoft.net/deepsound/

In deepsound basta aprire la traccia audio da cui si vogliano estrarre le informazioni, inserire la password con la quale le informazioni sono state (eventualmente) protette, e trovere in «Documenti» i contenuti nascosti, in chiaro.

In caso di protezione con password, nella cartella «run» di john the ripper c'è uno script molto utile:
`deepsound2john.py`