

## Zip of death

(compressione ricorsiva di file vuoti)

Augusto Scatolini ([webmaster@comunecampagnano.it](mailto:webmaster@comunecampagnano.it))

Ver. 1.0 Marzo 2011



Zip of death, letteralmente “**zip della morte**” è un vecchio attacco informatico che risale al 2001 molto interessante da punto di vista didattico, del funzionamento e della costruzione. Chiunque può costruirsi un “zip of death”, sostanzialmente è un piccolo file zippato con un nome attraente che viene allegato ad una mail o è scaricabile da qualche sito internet.

La particolarità consiste nel fatto che quando si tenta di aprire (decomprimere) questo file possono accadere tre cose in ordine di gravità:

1. viene interdetto (bloccato) l'antivirus e quindi il sistema diventa vulnerabile
2. il sistema va in crash per esaurimento della memoria (denial of service)
3. il disco viene saturato con qualche PetaByte di file che contengono NIENTE

Sempre per il principio per cui per difendersi bisogna conoscere il nemico e in base ai principi dell'Hacking Etico, in questa guida mostrerò come si costruisce una zip bomb e su quali principi si basa la costruzione stessa.

Spero che questa guida, scritta a puro scopo didattico, possa aiutare gli utenti a responsabilizzarsi circa i pericoli e i possibili attacchi informatici, attuare tutte le misure possibili di sicurezza e se possibile passare a sistemi GNU/Linux.

## **Compressione ricorsiva di file vuoti**

Tutte le guide che si trovano in rete sulla bomba zip risultano incomplete o insufficientemente chiare (forse volutamente) e sostanzialmente introducono il concetto di compressione ricorsiva.

La bomba zip più famosa, la 42.zip, (ancora reperibile in rete) è un file di soli 42 KB, per l'esattezza 42.374 byte.

Questo file zippato  
contiene 16 file zippati  
che contengono ognuno 16 file zippati  
che contengono ognuno 16 file zippati  
che contengono ognuno 16 file zippati  
che contengono ognuno 16 file zippati  
che contengono ognuno 1 solo file che misura 4,3 GigaByte

è facile verificare (teoricamente) che decomprimendo il file 42.zip si otterranno più di un milione di file da 4,3 GB l'uno per un totale di 4,5 PetaByte.

1 PB equivale a 1.000 TeraByte, (il TeraByte equivale a 1.000 GigaByte).

oggi i dischi più capienti sul mercato superano di poco 1 TB.

A questo punto chiunque si dovrebbe porre due domande:

1. Come è possibile che un file compresso di soli 42 KB contenga file per 4,5 PetaByte?
2. Come fa l'attaccante a maneggiare 4,5 PB di dati se non esistono ancora dischi così capienti?

La soluzione nelle prossime pagine

## Copiare file in modalità binaria e ricorsiva

Il primo passo per costruire un bomba zip è creare un file ASCII vuoto che chiameremo a.txt

apriamo questo file con notepad (blocco note) e scriviamo alcuni caratteri vuoti tenendo premuto il tasto Alt e digitando 225 sul tastierino numerico, quando si rilascia il tasto Alt vedrete che il cursore si sposta a destra di una posizione. Abbiamo scritto un carattere vuoto.

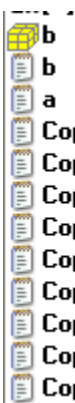
Ora dobbiamo creare una decina copie di questo file

ora dobbiamo fondere questi 10 file in modalità binaria e non in modalità ASCII

per fondere questi 10 file in modalità binaria digitare su una finestra DOS il seguente comando:  
copy /b \*.txt b.txt

Supponendo che la misura del file vuoto a.txt sia di 10 byte il file b.txt misurerà 100 Byte

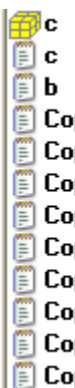
ora zippiamo il file b.txt e vedremo che il file compresso avrà una misura maggiore del file da comprimere.



b	zip	119
b	txt	100
a	txt	10
Copia (2) di a	txt	10
Copia (3) di a	txt	10
Copia (4) di a	txt	10
Copia (5) di a	txt	10
Copia (6) di a	txt	10
Copia (7) di a	txt	10
Copia (8) di a	txt	10
Copia (9) di a	txt	10
Copia di a	txt	10

Ora spostiamo il file b.txt in una nuova cartella e facciamo una decine di copie del file b.txt con il comando copy /b \*.txt c.txt

il file c.txt è ovviamente 10 volte la misura del file b.txt ma il suo equivalente file compresso è di soli 125 Byte



c	zip	125
c	txt	1.000
b	txt	100
Copia (2) di b	txt	100
Copia (3) di b	txt	100
Copia (4) di b	txt	100
Copia (5) di b	txt	100
Copia (6) di b	txt	100
Copia (7) di b	txt	100
Copia (8) di b	txt	100
Copia (9) di b	txt	100
Copia di b	txt	100

continuiamo con stessa sequenza fino al file h.txt e al suo corrispondente h.zip passando per  
d.txt,  
e.txt,  
f.txt e  
g.txt

d	zip	152	e	zip	326
d	txt	10.000	e	txt	100.000
c	txt	1.000	Copia (2) di d	txt	10.000
Copia (2) di c	txt	1.000	Copia (3) di d	txt	10.000
Copia (3) di c	txt	1.000	Copia (4) di d	txt	10.000
Copia (4) di c	txt	1.000	Copia (5) di d	txt	10.000
Copia (5) di c	txt	1.000	Copia (6) di d	txt	10.000
Copia (6) di c	txt	1.000	Copia (7) di d	txt	10.000
Copia (7) di c	txt	1.000	Copia (8) di d	txt	10.000
Copia (8) di c	txt	1.000	Copia (9) di d	txt	10.000
Copia (9) di c	txt	1.000	Copia di d	txt	10.000
Copia di c	txt	1.000	d	txt	10.000
f	zip	2.069	g	zip	19.539
f	txt	1.000.000	g	txt	10.000.000
Copia (2) di e	txt	100.000	Copia (2) di f	txt	1.000.000
Copia (3) di e	txt	100.000	Copia (3) di f	txt	1.000.000
Copia (4) di e	txt	100.000	Copia (4) di f	txt	1.000.000
Copia (5) di e	txt	100.000	Copia (5) di f	txt	1.000.000
Copia (6) di e	txt	100.000	Copia (6) di f	txt	1.000.000
Copia (7) di e	txt	100.000	Copia (7) di f	txt	1.000.000
Copia (8) di e	txt	100.000	Copia (8) di f	txt	1.000.000
Copia (9) di e	txt	100.000	Copia (9) di f	txt	1.000.000
Copia di e	txt	100.000	Copia di f	txt	1.000.000
e	txt	100.000	f	txt	1.000.000
h	zip	194.241	i	zip	1.941.302
h	txt	100.000.000	i	txt	1.000.000.000
Copia (2) di g	txt	10.000.000	Copia (2) di h	txt	100.000.000
Copia (3) di g	txt	10.000.000	Copia (3) di h	txt	100.000.000
Copia (4) di g	txt	10.000.000	Copia (4) di h	txt	100.000.000
Copia (5) di g	txt	10.000.000	Copia (5) di h	txt	100.000.000
Copia (6) di g	txt	10.000.000	Copia (6) di h	txt	100.000.000
Copia (7) di g	txt	10.000.000	Copia (7) di h	txt	100.000.000
Copia (8) di g	txt	10.000.000	Copia (8) di h	txt	100.000.000
Copia (9) di g	txt	10.000.000	Copia (9) di h	txt	100.000.000
Copia di g	txt	10.000.000	Copia di h	txt	100.000.000
g	txt	10.000.000	h	txt	100.000.000

si può notare che il file h.txt misura 100 MegaByte, il suo multiplo i.txt misura 1 Gbyte ma il suo equivalente file compresso i.zip misura meno di 2MegaByte












se mettiamo tutti questi dati su un foglio elettronico possiamo notare che il fattore di incremento del file zippato non è assolutamente uguale a quello del file binario

	n. file	byte/file	byte totali	assumendo 1024 = 1000	byte totali zippati	assumendo 1024 = 1000
<b>a</b>	10	10	100	100 byte	119	119 byte
<b>b</b>	10	100	1.000	1 KB	125	125 byte
<b>c</b>	10	1.000	10.000	10 KB	152	152 byte
<b>d</b>	10	10.000	100.000	100 KB	326	326 byte
<b>e</b>	10	100.000	1.000.000	1 MB	2.069	2,1 KB
<b>f</b>	10	1.000.000	10.000.000	10 MB	19.069	19 KB
<b>g</b>	10	10.000.000	100.000.000	100 MB	194.241	194 KB
<b>h</b>	10	100.000.000	1.000.000.000	1 GB	1.941.302	1,94 MB






















## Zippare file in maniera ricorsiva

In questa seconda fase invece di decuplicare l'ultimo file binario i.txt si duplica il suo equivalente zippato i.zip, poi si zippano tutti i file zippati prodotti

si noterà che la compressione di 10 file che misurano circa 2 MB produrrà un file di soli 27 KB

 L	zip	27.658
 Copia (2) di i	zip	1.941.302
 Copia (3) di i	zip	1.941.302
 Copia (4) di i	zip	1.941.302
 Copia (5) di i	zip	1.941.302
 Copia (6) di i	zip	1.941.302
 Copia (7) di i	zip	1.941.302
 Copia (8) di i	zip	1.941.302
 Copia (9) di i	zip	1.941.302
 Copia di i	zip	1.941.302
 i	zip	1.941.302

analogamente (ancora più spinto) con i prossimi due passaggi

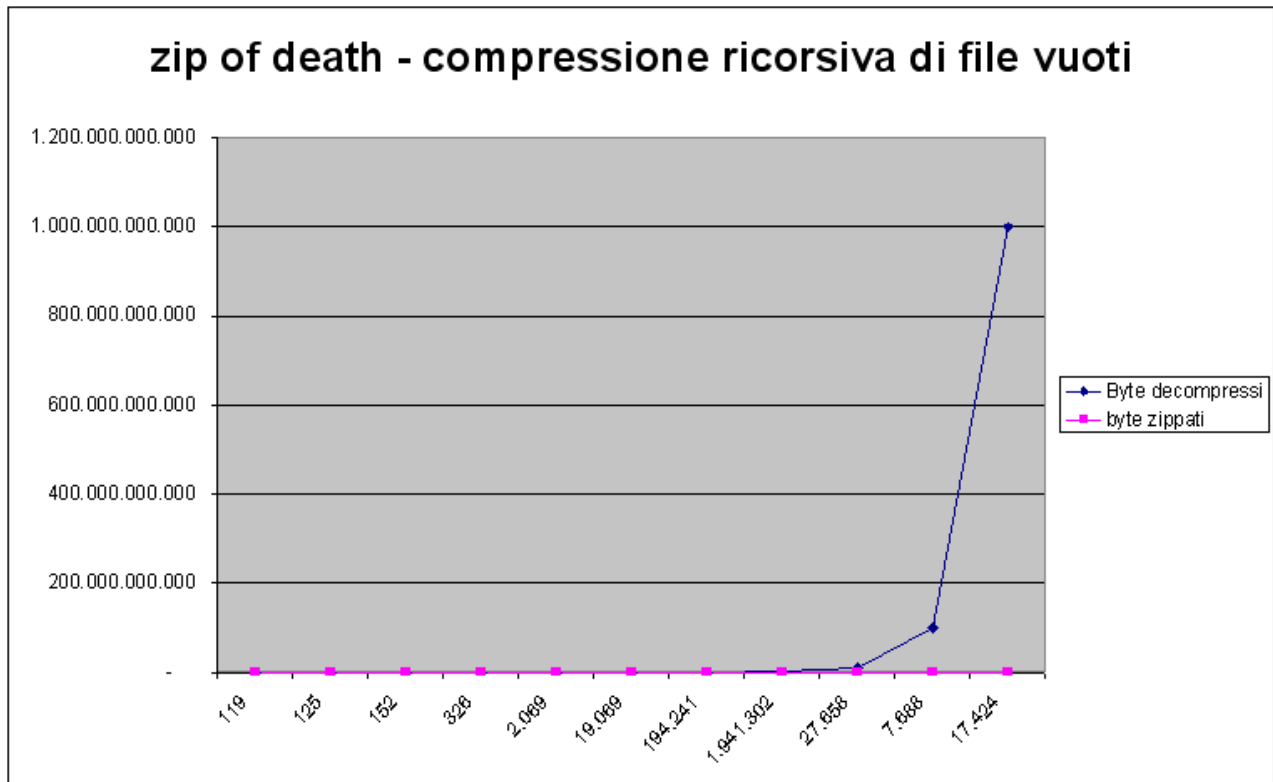
 M	zip	7.688	 m	zip	17.424
 Copia (2) di L	zip	27.658	 Copia (2) di M	ZIP	7.688
 Copia (3) di L	zip	27.658	 Copia (3) di M	ZIP	7.688
 Copia (4) di L	zip	27.658	 Copia (4) di M	ZIP	7.688
 Copia (5) di L	zip	27.658	 Copia (5) di M	ZIP	7.688
 Copia (6) di L	zip	27.658	 Copia (6) di M	ZIP	7.688
 Copia (7) di L	zip	27.658	 Copia (7) di M	ZIP	7.688
 Copia (8) di L	zip	27.658	 Copia (8) di M	ZIP	7.688
 Copia (9) di L	zip	27.658	 Copia (9) di M	ZIP	7.688
 Copia di L	zip	27.658	 Copia di M	ZIP	7.688
 L	zip	27.658			

l'ultimo file creato m.zip (ma si potrebbe continuare) di soli 17.424 Byte (17 KB) in realtà contiene tutti i file zippati precedenti che complessivamente occuperebbero (se aperti) 1 PetaByte

Ecco come è possibile maneggiare grandi quantità di Byte, primo perché sono zippati e secondo per il particolarissimo rapporto di compressione che si applica a questi tipi di file (ricordiamoci che sono centinaia di migliaia di file che contengono NIENTE)

nella prossima pagina la tabella riassuntiva dei dati con relativo grafico

	n. file	byte/file	byte totali	assumendo 1024 = 1000	byte totali zippati	assumendo 1024 = 1000
<b>a</b>	10	10	100	100 byte	119	119 byte
<b>b</b>	10	100	1.000	1 KB	125	125 byte
<b>c</b>	10	1.000	10.000	10 KB	152	152 byte
<b>d</b>	10	10.000	100.000	100 KB	326	326 byte
<b>e</b>	10	100.000	1.000.000	1 MB	2.069	2,1 KB
<b>f</b>	10	1.000.000	10.000.000	10 MB	19.069	19 KB
<b>g</b>	10	10.000.000	100.000.000	100 MB	194.241	194 KB
<b>h</b>	10	100.000.000	1.000.000.000	1 GB	1.941.302	1,94 MB
<b>i</b>	10	1.941.302	10.000.000.000	10 GB	27.658	27 KB
<b>L</b>	10	27.658	100.000.000.000	100 GB	7.688	7,7 KB
<b>m</b>	10	7.688	1.000.000.000.000	1 PB	17.424	17,4 KB



FINE

Questo documento è rilasciato con licenza Copyleft  
(tutti i rovesci sono riservati)

altre miniguide su  
<http://www.comunecampagnano.it/gnu/miniguide.htm>  
oppure direttamente su  
<http://miniguide.tk>