

Selezione pubblica, per titoli ed esami, per il reclutamento di una unità di personale da assumere con contratto di lavoro a tempo determinato e pieno di categoria C, posizione economica C1, area tecnica, tecnico-scientifica ed elaborazione dati, della durata di 36 mesi, presso il Dipartimento di Matematica dell'Università degli Studi di Roma "Tor Vergata", nell'ambito del Progetto: "Sistemi operativi e server – Progetto Dipartimento di eccellenza"

D.D. n. 1166 del 15/05/2019 – Scadenza 14/06/2019 – Rif. 1459

PROVA SCRITTA – testo n. 3

Il presente tema d'esame si compone di due parti: un questionario sulla legislazione universitaria e un problema sulla gestione di un sistema operativo.

Richieste, raccomandazioni e regole. Il/la candidato/a restituisca il questionario a pagina 2, debitamente compilato. Inoltre, a lui/lei si richiede di scrivere sul/i foglio/i ricevuto/i inizialmente tutte le istruzioni, che traducono in linguaggio di programmazione la procedura descritta a partire dalla pagina 3. Il linguaggio (*PHP, C, bash scripting*, etc.) è a scelta del/la candidato/a, ma verrà esplicitamente valutata la fattibilità della soluzione adottata dal/la candidato/a medesimo/a.

Si noti che, al termine della descrizione dell'algoritmo, sono allegate alcune importanti informazioni: la struttura tipica di qualche specifico *file* di sistema; le pagine del manuale *online* che sono accessibili da una finestra di terminale in ambiente *GNU/Linux*) e che descrivono alcuni comandi di sistema; etc. Tali contenuti sono inclusi in opportuni riquadri e, laddove è necessario, ad essi ci si riferisce nel testo seguente.

È possibile (anche se non strettamente necessario) aggiungere commenti tra un'istruzione e l'altra, al fine di introdurre delle spiegazioni, qualora il/la candidato/a lo ritenga opportuno. La commissione raccomanda di adottare soluzioni semplici da implementare e da gestire; inoltre, verrà valutata anche la chiarezza con la quale esse sono esposte.

Durante la prova, è consentito l'accesso a manuali e a un *personal computer*, purché esso sia escluso dalla rete. È invece vietato l'utilizzo di telefoni cellulari, *tablet* e altri oggetti che consentono la comunicazione con l'esterno dell'aula d'esame.

La durata della prova scritta è di due ore, al termine delle quali il/la candidato/a dovrà necessariamente consegnare il proprio elaborato.

Questionario riguardante la legislazione universitaria

- [1] Al Credito Formativo Universitario corrispondono:
- ☐ 25 ore di impegno complessivo per studente;
 - ☐ 60 ore di impegno complessivo per studente;
 - ☐ 50 ore di impegno complessivo per studente.
- [2] Quale delle seguenti alternative è corretta?
- ☐ il Rettore presiede il Senato Accademico;
 - ☐ il Rettore presiede il Collegio di disciplina;
 - ☐ il Rettore presiede il Consiglio di Dipartimento.
- [3] Gli obblighi di frequenza agli insegnamenti dei corsi di laurea e di laurea magistrale sono disciplinati da:
- ☐ Statuto di Ateneo;
 - ☐ Regolamento Didattico del corso di studio;
 - ☐ Regolamento Didattico di Ateneo.
- [4] Secondo la disciplina in tema di lavoro universitario, nel comparto università è prevista la possibilità per i dipendenti di ottenere un'aspettativa?
- ☐ Sì, ma solo se lo consente il Regolamento d'Ateneo concernente l'amministrazione del personale;
 - ☐ No, mai;
 - ☐ Sì, a determinate condizioni.

Problema

Scrivere un programma che costituisca un *firewall* selettivo, al fine di escludere l'accesso a un *server* di tipo *GNU/Linux* limitatamente a quegli indirizzi *IP*, da cui sono stati effettuati tentativi di intrusione sul *server* medesimo.

Descrizione dettagliata dell'algoritmo

- [1] Si ispezioni il *file* `auth.log` all'interno della cartella `/var/log/` (il cui accesso è riservato all'amministratore del sistema operativo). Un esempio che illustra la struttura tipica del suddetto *file* è riportato nel riquadro di figura 1.
- [2] Si considerino tutte e sole le righe del *file* in cui compare la scritta
`Failed password for invalid user`
 e si proceda come descritto nei seguenti punti [2A]–[2C].
- [2A] Da ciascuna riga in cui compare la suddetta scritta, si legga l'indirizzo *IP* e lo si memorizzi opportunamente (a scelta del/la candidato/a su di un *file*, in un *array*, in una *lista*, in un'opportuna variabile temporanea, etc.).
- [2B] Per ogni indirizzo *IP* tra quelli memorizzati, si conti il numero delle sue occorrenze in quelle righe del *file* `/var/log/auth.log`, che sono state selezionate così come richiesto al punto [2].
- [2C] Per ogni indirizzo *IP* il cui numero di occorrenze è maggiore di^[*] N , si deve predisporre un blocco silente^[⊙] sulla porta *ssh* del sistema, utilizzando il comando `iptables`. Nei riquadri delle figure 2–6 sono riportate le spiegazioni riguardanti l'utilizzo di `iptables`, così come esse sono descritte nel manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*).
- [3] Per quanto riguarda i tentativi di intrusione in un *account* effettivamente esistente sul *server*, si proceda in modo analogo a quanto de-

[*] N può essere una variabile, una costante fissata all'interno del programma o, più semplicemente, un numero. In quest'ultimo caso, per fissare le idee, sia $N = 8$.

[⊙] Nel gergo comune riguardante i *firewall*, un blocco su di una porta *ssh* si dice silente, quando un tentativo di collegamento *ssh* dall'esterno non riceve alcuna risposta.

scritto in precedenza. In dettaglio, si considerino tutte e sole le righe del *file* in cui compare la scritta

Failed password for

ma **non** compare la scritta

invalid user

e si proceda come descritto nei seguenti punti [3A]–[3C].

- [3A] Da ciascuna riga con le suddette caratteristiche, si legga l'indirizzo *IP* e lo si memorizzi opportunamente (a scelta del/la candidato/a su di un *file*, in un *array*, in una *lista*, in un'opportuna variabile temporanea, etc.).
- [3B] Per ogni indirizzo *IP* tra quelli memorizzati, si conti il numero delle sue occorrenze in quelle righe del *file* */var/log/auth.log*, che sono state selezionate così come richiesto al punto [3].
- [3C] Per ogni indirizzo *IP* il cui numero di occorrenze è maggiore di^[⊗] *M*, si deve predisporre un blocco silente^[⊙] sulla porta *ssh* del sistema, utilizzando il comando *iptables*. Nei riquadri delle figure 2–6 sono riportate le spiegazioni riguardanti l'utilizzo di *iptables*, così come esse sono descritte nel manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*).

Nota. Saranno apprezzate quelle soluzioni, dove sono incluse in un unico ciclo le istruzioni che traducono in pratica i punti [2]–[2C] e [3]–[3C], in modo da ridurre la duplicazione di istruzioni identiche o, quantomeno, simili. L'unificazione in un solo ciclo non è strettamente necessaria, ma è gradita perché semplificherebbe la gestione del programma, in previsione di eventuali future modifiche.

Allegati. Seguono due allegati, distribuiti su 6 pagine.

^[⊗] *M* può essere una variabile, una costante fissata all'interno del programma o, più semplicemente, un numero. In quest'ultimo caso, per fissare le idee, sia $M = 10$.

```

Jun 2 00:09:01 Marvin CRON[909]: pam_unix(cron:session): session opened for user root by (uid=0)
Jun 2 00:09:01 Marvin CRON[909]: pam_unix(cron:session): session closed for user root
Jun 2 00:09:20 Marvin sshd[922]: Invalid user ncs from 93.114.77.11 port 57766
Jun 2 00:09:20 Marvin sshd[922]: input_userauth_request: invalid user ncs [preauth]
Jun 2 00:09:20 Marvin sshd[922]: pam_unix(sshd:auth): check pass; user unknown
Jun 2 00:09:20 Marvin sshd[922]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=93.114.77.11
Jun 2 00:09:22 Marvin sshd[922]: Failed password for invalid user ncs from 93.114.77.11 port 57766 ssh2
Jun 2 00:09:22 Marvin sshd[922]: Received disconnect from 93.114.77.11 port 57766:11: Bye Bye [preauth]
Jun 2 00:09:22 Marvin sshd[922]: Disconnected from 93.114.77.11 port 57766 [preauth]
Jun 2 00:12:50 Marvin sshd[966]: Invalid user hscroot from 93.114.77.11 port 54078
Jun 2 00:12:50 Marvin sshd[966]: input_userauth_request: invalid user hscroot [preauth]
Jun 2 00:12:50 Marvin sshd[966]: pam_unix(sshd:auth): check pass; user unknown
Jun 2 00:12:50 Marvin sshd[966]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=93.114.77.11
Jun 2 00:12:52 Marvin sshd[966]: Failed password for invalid user hscroot from 93.114.77.11 port 54078 ssh2
Jun 2 00:12:53 Marvin sshd[966]: Received disconnect from 93.114.77.11 port 54078:11: Bye Bye [preauth]
Jun 2 00:12:53 Marvin sshd[966]: Disconnected from 93.114.77.11 port 54078 [preauth]
.
.
.
Jun 2 00:17:01 Marvin CRON[1022]: pam_unix(cron:session): session opened for user root by (uid=0)
Jun 2 00:17:01 Marvin CRON[1022]: pam_unix(cron:session): session closed for user root
Jun 2 00:17:16 Marvin sshd[1027]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=182.52.224.33 user=root
Jun 2 00:17:18 Marvin sshd[1027]: Failed password for root from 182.52.224.33 port 44886 ssh2
Jun 2 00:17:18 Marvin sshd[1027]: Received disconnect from 182.52.224.33 port 44886:11: Normal Shutdown, Thank you for playing [preauth]
Jun 2 00:17:18 Marvin sshd[1027]: Disconnected from 182.52.224.33 port 44886 [preauth]
.
.
.
Jun 3 18:27:40 Marvin sshd[12695]: Accepted password for ugo-loc from 78.13.173.32 port 57798 ssh2
Jun 3 18:27:40 Marvin sshd[12695]: pam_unix(sshd:session): session opened for user ugo-loc by (uid=0)
Jun 3 18:27:40 Marvin systemd-logind[417]: New session 1496 of user ugo-loc.
Jun 3 19:55:45 Marvin sshd[12704]: Disconnected from 78.13.173.32 port 57798
Jun 3 19:55:45 Marvin sshd[12695]: pam_unix(sshd:session): session closed for user ugo-loc
.
.
.
Jun 5 16:37:34 Marvin sshd[27682]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=160.80.11.161 user=root
Jun 5 16:37:36 Marvin sshd[27682]: Failed password for root from 160.80.11.161 port 39170 ssh2
Jun 5 16:37:50 Marvin sshd[27682]: Failed password for root from 160.80.11.161 port 39170 ssh2
Jun 5 16:38:01 Marvin sshd[27682]: Connection closed by 160.80.11.161 port 39170 [preauth]
Jun 5 16:38:01 Marvin sshd[27682]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=160.80.11.161 user=root
.
.
.

```

Figure 1. Esempio di *file* /var/log/auth.log – Ovviamente, quando compaiono sequenze verticali di punti del tipo `:`, si deve intendere che le righe mancanti sono analoghe a quelle precedenti.

IPTABLES(8)	iptables 1.4.21	IPTABLES(8)
NAME	iptables/ip6tables âM-^@M-^T administration tool for IPv4/IPv6 packet filtering and NAT	
SYNOPSIS	<pre> iptables [-t table] {-A -C -D} chain rule-specification ip6tables [-t table] {-A -C -D} chain rule-specification iptables [-t table] -I chain [rulenum] rule-specification iptables [-t table] -R chain rulenum rule-specification iptables [-t table] -D chain rulenum iptables [-t table] -S [chain [rulenum]] iptables [-t table] {-F -L -Z} [chain [rulenum]] [options...] iptables [-t table] -N chain iptables [-t table] -X [chain] iptables [-t table] -P chain target iptables [-t table] -E old-chain-name new-chain-name rule-specification = [matches...] [target] match = -m matchname [per-match-options] target = -j targetname [per-target-options]</pre>	
DESCRIPTION	<p>Iptables and ip6tables are used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.</p> <p>Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a 'target', which may be a jump to a user-defined chain in the same table.</p>	
TARGETS	<p>A firewall rule specifies criteria for a packet and a target. If the packet does not match, the next rule in the chain is examined; if it does match, then the next rule is specified by the value of the target, which can be the name of a user-defined chain, one of the targets described in iptables-extensions(8), or one of the special values ACCEPT, DROP or RETURN.</p> <p>ACCEPT means to let the packet through. DROP means to drop the packet on the floor. RETURN means stop traversing this chain and resume at the next rule in the previous (calling) chain. If the end of a built-in chain is reached or a rule in a built-in chain with target RETURN is matched, the target specified by the chain policy determines the fate of the packet.</p>	
TABLES	<p>There are currently five independent tables (which tables are present at any time depends on the kernel configuration options and which modules are present).</p> <p>-t, --table table This option specifies the packet matching table which the command should operate on. If the kernel is configured with automatic module loading, an attempt will be made to load the appropriate module for that table if it is not already there.</p> <p>The tables are as follows:</p> <p>filter: This is the default table (if no -t option is passed). It contains the built-in chains INPUT (for packets destined to local sockets), FORWARD (for packets being routed through the box), and OUTPUT (for locally-generated packets).</p> <p>nat: This table is consulted when a packet that creates a new connection is encountered. It consists of three built-ins: PREROUTING (for altering packets as soon as they come in), OUTPUT (for altering locally-generated packets before routing), and POSTROUTING (for altering packets as they are about to go out). IPv6 NAT support is available since kernel 3.7.</p> <p>mangle: This table is used for specialized packet alteration. Until kernel 2.4.17 it had two built-in chains: PREROUTING (for altering incoming packets before routing) and OUTPUT (for</p>	

Figure 2. Parte del manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*) riguardo al comando `iptables` – Pagina 1 di 5.

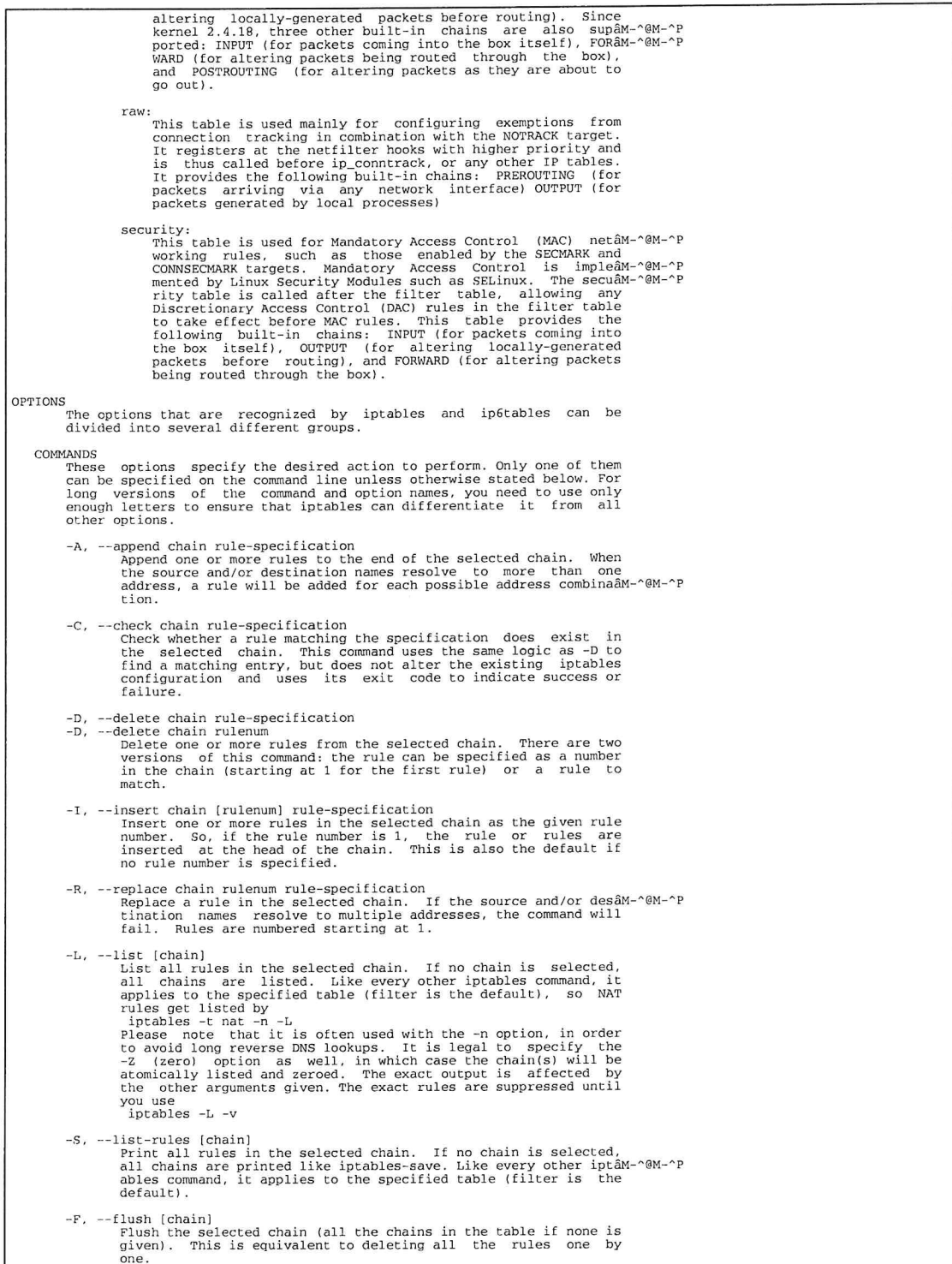


Figure 3. Parte del manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*) riguardo al comando `iptables` – Pagina 2 di 5.

```

-Z, --zero [chain [rulenum]]
    Zero the packet and byte counters in all chains, or only the
    given chain, or only the given rule in a chain. It is legal to
    specify the -L, --list (list) option as well, to see the counâM-^@M-^P
    ters immediately before they are cleared. (See above.)

-N, --new-chain chain
    Create a new user-defined chain by the given name. There must
    be no target of that name already.

-X, --delete-chain [chain]
    Delete the optional user-defined chain specified. There must be
    no references to the chain. If there are, you must delete or
    replace the referring rules before the chain can be deleted.
    The chain must be empty, i.e. not contain any rules. If no
    argument is given, it will attempt to delete every non-builtin
    chain in the table.

-P, --policy chain target
    Set the policy for the chain to the given target. See the secâM-^@M-^P
    tion TARGETS for the legal targets. Only built-in (non-user-
    defined) chains can have policies, and neither built-in nor
    user-defined chains can be policy targets.

-E, --rename-chain old-chain new-chain
    Rename the user specified chain to the user supplied name. This
    is cosmetic, and has no effect on the structure of the table.

-h      Help. Give a (currently very brief) description of the command
    syntax.

PARAMETERS
The following parameters make up a rule specification (as used in the
add, delete, insert, replace and append commands).

-4, --ipv4
    This option has no effect in iptables and iptables-restore. If
    a rule using the -4 option is inserted with (and only with)
    ip6tables-restore, it will be silently ignored. Any other uses
    will throw an error. This option allows IPv4 and IPv6 rules in a
    single rule file for use with both iptables-restore and
    ip6tables-restore.

-6, --ipv6
    If a rule using the -6 option is inserted with (and only with)
    iptables-restore, it will be silently ignored. Any other uses
    will throw an error. This option allows IPv4 and IPv6 rules in a
    single rule file for use with both iptables-restore and
    ip6tables-restore. This option has no effect in ip6tables and
    ip6tables-restore.

[!] -p, --protocol protocol
    The protocol of the rule or of the packet to check. The speciâM-^@M-^P
    fied protocol can be one of tcp, udp, udplite, icmp, icmpv6, esp,
    ah, sctp, mh or the special keyword "all", or it can be a
    numeric value, representing one of these protocols or a differâM-^@M-^P
    ent one. A protocol name from /etc/protocols is also allowed.
    A "!" argument before the protocol inverts the test. The number
    zero is equivalent to all. "all" will match with all protocols
    and is taken as default when this option is omitted. Note that,
    in ip6tables, IPv6 extension headers except esp are not allowed.
    esp and ipv6-nonext can be used with Kernel version 2.6.11 or
    later. The number zero is equivalent to all, which means that
    you cannot test the protocol field for the value 0 directly. To
    match on a HBH header, even if it were the last, you cannot use
    -p 0, but always need -m hbh.

[!] -s, --source address[/mask][...]
    Source specification. Address can be either a network name, a
    hostname, a network IP address (with /mask), or a plain IP
    address. Hostnames will be resolved once only, before the rule
    is submitted to the kernel. Please note that specifying any
    name to be resolved with a remote query such as DNS is a really
    bad idea. The mask can be either an ipv4 network mask (for iptâM-^@M-^P
    ables) or a plain number, specifying the number of 1's at the
    left side of the network mask. Thus, an iptables mask of 24 is
    equivalent to 255.255.255.0. A "!" argument before the address
    specification inverts the sense of the address. The flag --src
    is an alias for this option. Multiple addresses can be speciâM-^@M-^P
    fied, but this will expand to multiple rules (when adding with
    -A), or will cause multiple rules to be deleted (with -D).

[!] -d, --destination address[/mask][...]
    Destination specification. See the description of the -s
    (source) flag for a detailed description of the syntax. The
    flag --dst is an alias for this option.

-m, --match match
    Specifies a match to use, that is, an extension module that
    tests for a specific property. The set of matches make up the
    condition under which a target is invoked. Matches are evaluated
    first to last as specified on the command line and work in
    short-circuit fashion, i.e. if one extension yields false, evalâM-^@M-^P
    uation will stop.

```

Figure 4. Parte del manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*) riguardo al comando `iptables` – Pagina 3 di 5.

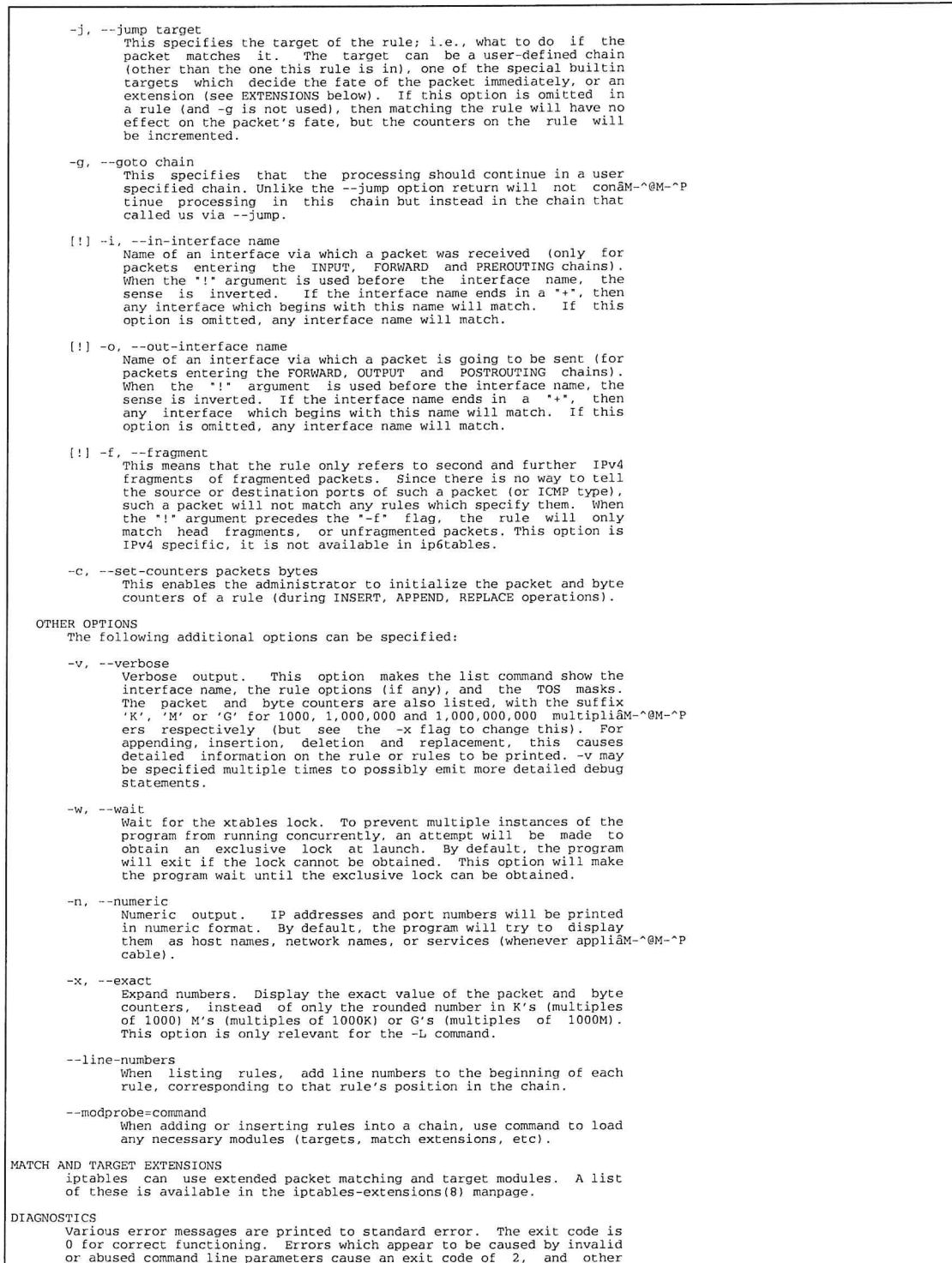


Figure 5. Parte del manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*) riguardo al comando `iptables` – Pagina 4 di 5.

```

errors cause an exit code of 1.

BUGS
  Bugs?  What's this?  ;-)  Well, you might want to have a look at
  http://bugzilla.netfilter.org/

COMPATIBILITY WITH IPCHAINS
  This iptables is very similar to ipchains by Rusty Russell.  The main
  difference is that the chains INPUT and OUTPUT are only traversed for
  packets coming into the local host and originating from the local host
  respectively.  Hence every packet only passes through one of the three
  chains (except loopback traffic, which involves both INPUT and OUTPUT
  chains); previously a forwarded packet would pass through all three.

  The other main difference is that -i refers to the input interface; -o
  refers to the output interface, and both are available for packets
  entering the FORWARD chain.

  The various forms of NAT have been separated out; iptables is a pure
  packet filter when using the default 'filter' table, with optional
  extension modules.  This should simplify much of the previous confusion
  over the combination of IP masquerading and packet filtering seen
  previously.  So the following options are handled differently:
  -j MASQ
  -M -S
  -M -L
  There are several other changes in iptables.

SEE ALSO
  iptables-apply(8),  iptables-save(8),  iptables-restore(8),  iptables-
  extensions(8),

  The packet-filtering-HOWTO details iptables usage for packet filtering,
  the NAT-HOWTO details NAT, the netfilter-extensions-HOWTO details the
  extensions that are not in the standard distribution, and the netfilter-
  hacking-HOWTO details the netfilter internals.
  See http://www.netfilter.org/.

AUTHORS
  Rusty Russell originally wrote iptables, in early consultation with
  Michael Neuling.

  Marc Boucher made Rusty abandon ipnatctl by lobbying for a generic
  packet selection framework in iptables, then wrote the mangle table,
  the owner match, the mark stuff, and ran around doing cool stuff every-
  where.

  James Morris wrote the TOS target, and tos match.

  Jozsef Kadlecsek wrote the REJECT target.

  Harald Welte wrote the ULOG and NFQUEUE target, the new libiptc, as
  well as the TTL, DSCP, ECN matches and targets.

  The Netfilter Core Team is: Marc Boucher, Martin Josefsson, Yasuyuki
  Kozakai, Jozsef Kadlecsek, Patrick McHardy, James Morris, Pablo Neira
  Ayuso, Harald Welte and Rusty Russell.

  Man page originally written by Herve Eychenne <rv@wallfire.org>.

VERSION
  This manual page applies to iptables/ip6tables @PACKAGE_AND_VERSION@.

iptables 1.4.21                                IPTABLES(8)

```

Figure 6. Parte del manuale *online* (accessibile da una qualsiasi finestra di terminale, in ambiente *Linux*) riguardo al comando `iptables` – Pagina 5 di 5.