

# Αναφορά Εργασίας 2: Linux Firewall - Ερωτήσεις κατανόησης στη χρήση iptables

Γρηγόρης Καπαδούκας (ΑΜ: 1072484)

27 Οκτωβρίου 2022

## 1 Ερωτήσεις Κατανόησης:

### 1.1

Κανονικά δεν είναι αναγκαία η απευθείας ενσωμάτωση του TCP/IP στο λειτουργικό σύστημα της μηχανής, καθώς θα μπορούσε να υπάρξει usermode υλοποίηση που να έτρεχε εκτός του πυρήνα (kernel space)

Παρόλα αυτά όμως τα filtering firewalls αξιοποιούν την ύπαρξη της υποστήριξη του TCP/IP σε μονολιθικούς πυρήνες, όπως του Linux επειδή καθιστούν το φιλτράρισμα πιο εύκολο και αποδοτικό, εφ' όσον κώδικας που τρέχει στο kernel space έχει τη δυνατότητα να χειριστεί τα πακέτα απευθείας αντί να τα μεταφέρει πρώτα στο userspace, καθιστώντας την υλοποίηση πολύ γρηγορότερη.

### 1.2

Στη περίπτωση ενός τείχους προστασίας διακομιστή μεσολάβησης όλο το traffic του LAN περνάει από το τείχος αυτό, που έχει ως αποτέλεσμα να ελέγχεται το traffic όλου του LAN.

Στη περίπτωση ενός τείχους προστασίας φιλτραρίσματος πακέτων έχουμε φιλτράρισμα του traffic του ίδιου του υπολογιστή, από τον ίδιο τον υπολογιστή, συνήθως χρησιμοποιώντας το TCP/IP stack που υποστηρίζεται από τον πυρήνα του υπολογιστή.

Οι δύο αυτές αρχιτεκτονικές μπορούν να συνδυαστούν, για παράδειγμα κανόνες που είναι κοινοί για όλο το δίκτυο μπορούν να εφαρμοστούν από τείχος προστασίας διακομιστή μεσολάβησης και οι κανόνες που είναι ξεχωριστοί για κάθε σύστημα να εφαρμόζονται από τείχος προστασίας φιλτραρίσματος πακέτων πάνω στο σύστημα αυτό.

### 1.3

Οι τέσσερις κύριοι πίνακες που διατηρούνται από τον πυρήνα Linux για την επεξεργασία εισερχόμενων και εξερχόμενων πακέτων είναι οι

- MANGLE
- FILTER
- NAT
- RAW

Στην πραγματικότητα υπάρχει και ένας πέμπτος πίνακας, ο "SECURITY", ο οποίος επιτρέπει στους χρήστες να φτιάχνουν εσωτερικούς SELinux security context marks σε πακέτα.

### 1.4

Τα πακέτα που προωθούνται στην αλυσίδα κανόνων INPUT έχουν πάντα προορισμό την IP διεύθυνση της κάρτας δικτύου στην οποία λειτουργεί το τείχος προστασίας. Άρα γίνεται να ανιχνευτεί ένα πακέτο που θα προωθηθεί στην αλυσίδα κανόνων INPUT από το destination IP πεδίο του πακέτου.

Τα πακέτα που προωθούνται στην αλυσίδα κανόνων OUTPUT έχουν πάντα ως διεύθυνση πηγής την IP διεύθυνση της κάρτας δικτύου στην οποία λειτουργεί το τείχος προστασίας. Άρα γίνεται να ανιχνευτεί ένα πακέτο που θα προωθηθεί στην αλυσίδα κανόνων OUTPUT από το source IP πεδίο του πακέτου.

Τα πακέτα που προωθούνται στην αλυσίδα κανόνων FORWARD έχουν σκοπό να δρομολογηθούν περεταίρω στο δίκτυο μέχρι να βρουν τον προορισμό τους. Άρα για να ανιχνευτούν ελέγχεται και το source IP πεδίο και το destination IP πεδίο του πακέτου και δεν πρέπει κανένα από τα δύο να είναι ίσο με την IP διεύθυνση της κάρτας δικτύου στην οποία λειτουργεί το

τείχος προστασίας.

Σε γενικές γραμμές αυτό που μελετάται για να αντιληφθεί εάν ένα πακέτο εμπίπτει ή όχι σε κάποια συνθήκη μιας εντολής των παραπάνω αλυσίδων είναι το packet header (που περιέχει τις πληροφορίες όπως source και destination IP).

## 1.5

Αν ένα πακέτο φτάσει το τέλος μιας αλυσίδας και δεν έχει βρεθεί κανόνας που να το πληρεί, τότε ο πυρήνας του Linux ελέγχει τη λεγόμενη πολιτική αλυσίδας που προβλέπει τι γίνεται με πακέτα σε αυτές τις περιπτώσεις. Η πιο συνηθισμένη πολιτική αλυσίδας σε συστήματα που έχουν σκοπό την υψηλή ασφάλεια είναι να γίνεται DROP το πακέτο.

## 1.6

Η εντολή που θα χρησιμοποιήσουμε είναι η εξής:

```
iptables -A INPUT -s 0/0 -p tcp --syn -j DROP
```

Αρχικά με το -A κάνουμε append τον νέο κανόνα στην αλυσίδα INPUT, με το -s ορίζουμε τα source IPs των πακέτων για τα οποία θα εφαρμοστεί ο κανόνας (0/0 είναι όλες οι διευθύνσεις), με το -p tcp ορίζουμε ότι το πρωτόκολλο που θέλουμε να φιλτράρουμε είναι το TCP, με το --syn ορίζουμε ισοδύναμο κανόνα με το --tcp-flag SYN,RST,ACK,FIN SYN το οποίο διακρίνει τα TCP flags και ορίζουμε να φιλτράρει μόνο τα TCP πακέτα με SYN flag = 1. Τέλος με το -j ορίζουμε το jump και με DROP ορίζουμε ότι τα πακέτα που πληρούν τις προϋποθέσεις που τέθηκαν παραπάνω (οποιαδήποτε IP διεύθυνση, TCP πρωτόκολλο και SYN flag), να τα κάνει DROP.

## 1.7

Η εντολή της iptables για να αρχικοποιηθούν (flush) όλες οι αλυσίδες που ορίζονται από τον χρήστη σε έναν πίνακα είναι η εξής:

```
iptables -X <όνομα πίνακα>
```

Για να αρχικοποιηθούν όλοι οι κανόνες σε έναν πίνακα εκτελούμε την εξής εντολή:

`iptables -F`

Η οποία διαγράφει όλους τους κανόνες από όλες τις αλυσίδες, άρα όλους τους κανόνες στον πίνακα.

## 1.8

Το ICMP type 255 κανονικά δεν είναι valid ICMP type, αλλά σε αυτή τη περίπτωση εννοεί όλα τα πακέτα ICMP. Η γραμμή αυτή που θα έχει τη συμβολοσειρά «icmp type 255» στο τέλος θα προήλθε από μια εντολή iptables με όρισμα "-p icmp --icmp-type any".

## 1.9

Ο τύπος ICMP που σχετίζεται με το echo-request (ping) είναι το ICMP type 8, ενώ ο τύπος ICMP που σχετίζεται με το echo-reply (pong) είναι το ICMP type 0.

## 1.10

Ο πίνακας RAW χρησιμοποιείται στην περίπτωση που θέλουμε να ορίσουμε κανόνες για connection tracking, αλλά ταυτόχρονα μερικοί από τους κανόνες μας να μην είναι exposed σε αυτούς. Άρα όταν υπάρχει RAW κανόνας έχει προτεραιότητα σε σχέση με όλους τους υπόλοιπους και τα πακέτα που υπόκεινται στους κανόνες του RAW πίνακα δεν δέχονται connection tracking.

## 1.11

Η εντολή iptables που ζητείται είναι η εξής:

```
iptables -A INPUT ! -s 192.168.0.0/16 -p tcp --syn --dport 22 -j DROP
```

Η εντολή που μας ζητείται είναι ισοδύναμη με το να απορρίπτουμε εξωτερικά αιτήματα σύνδεσης για τη θύρα 22 (στην οποία λειτουργεί ο sshd). Με αυτόν τον τρόπο μόνο εσωτερικά αιτήματα σύνδεσης στη θύρα 22 αποδέχονται και όλα τα άλλα πακέτα αιτήματος σύνδεσης από απομακρυσμένους πελάτες απορρίπτονται, όπως ζητάει η εκφώνηση.

Με την παραπάνω εντολή προσθέτουμε στον πίνακα FILTER στην αλυσίδα INPUT κανόνα που για κάθε εξωτερική IP (δηλαδή όλες τις IP που δεν ανήκουν στο 192.168.0.0/16 που είναι το εσωτερικό δίκτυο) απορρίπτει TCP πακέτα με SYN flag = 1 που έχουν destination port ίσο με 22. Ο λόγος που ορίζουμε πρωτόκολλο TCP είναι επειδή το sshd λειτουργεί με αυτό, και ο λόγος που ορίζουμε μόνο τα SYN πακέτα είναι επειδή η εκφώνηση αναφέρεται μόνο σε αιτήματα σύνδεσης, άρα SYN όχι πακέτα γενικότερα.

## 1.12

Η παρακολούθηση σύνδεσης των iptables είναι ένα feature που προσθέτει κατανόηση stream πακέτων. Είναι βασισμένο στην ιδέα της κατάστασης (state) ενός πακέτου. Αν ένα πακέτο είναι το πρώτο που ανιχνεύει το τείχος προστασίας, θεωρείται ότι είναι στην κατάσταση NEW (πχ SYN packet σε TCP σύνδεση) ή αν αντιληφθεί ότι το πακέτο είναι κομμάτι μιας ήδη αρχικοποιημένης ροής (stream) πακέτων τότε θεωρεί το state ESTABLISHED (στο παράδειγμα παραπάνω προφανώς θεωρούμε ροή πακέτων μέχρι να ανιχνευτεί RST ή FIN πακέτο).

## 1.13

Οι καταστάσεις που αναγνωρίζονται από τη κατάσταση της σύνδεσης (connection tracking) του iptables είναι οι εξής:

- NEW
- ESTABLISHED
- RELATED
- INVALID

## 1.14

Το παράδειγμα χρήσης iptables που μελετάμε θα αναλυθεί στην παρακάτω ενότητα της εργασίας

## 2 Εργασία:

Το τείχος προστασίας που φτιάχτηκε να επιτελεί τα χαρακτηριστικά που ζητούνται είναι το εξής:

```
#Αρχή myfirewall.sh
```

```
#Καθαρισμός προηγούμενων κανόνων  
iptables -X  
iptables -F
```

```
#Ερώτημα 1  
iptables -A OUTPUT -j ACCEPT
```

```
#Ερώτημα 2  
iptables -A INPUT -p tcp --dport 22 -m iprange --src-range 150.140.139.194-  
150.140.139.255 -j ACCEPT
```

```
#Ερώτημα 3  
iptables -A INPUT -s 192.168.0.0/16 -p tcp --dport 22 -j ACCEPT
```

```
#Ερώτημα 4  
iptables -A INPUT -s 192.168.122.1 -p tcp --dport 80 -j ACCEPT
```

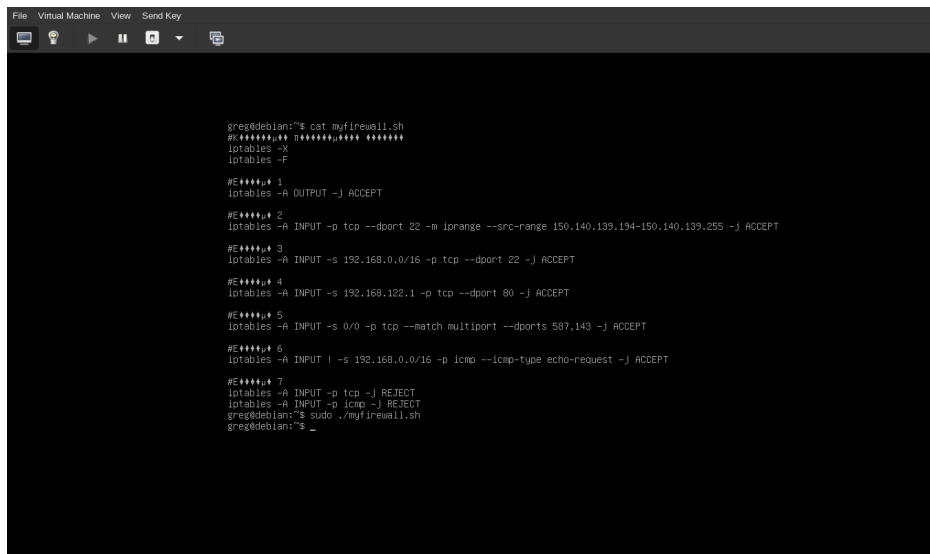
```
#Ερώτημα 5  
iptables -A INPUT -s 0/0 -p tcp --match multiport --dports 587,143 -j AC-  
CEPT
```

```
#Ερώτημα 6  
iptables -A INPUT ! -s 192.168.0.0/16 -p icmp --icmp-type echo-request -j  
ACCEPT
```

```
#Ερώτημα 7  
iptables -A INPUT -p tcp -j REJECT  
iptables -A INPUT -p icmp -j REJECT
```

```
#Τέλος myfirewall.sh
```

Παρακάτω δίνονται screenshots εκτέλεσης του script από εικονική μηχανή debian11 headless:



```
greg@debian:~$ cat myfirewall.sh
#*****
iptables -X
iptables -F

#***** 1
iptables -A OUTPUT -j ACCEPT

#***** 2
iptables -A INPUT -p tcp --dport 22 -m iprange --src-range 150.140.139.194-150.140.139.255 -j ACCEPT

#***** 3
iptables -A INPUT -s 192.168.0.0/16 -p tcp --dport 22 -j ACCEPT

#***** 4
iptables -A INPUT -s 192.168.102.1 -p tcp --dport 80 -j ACCEPT

#***** 5
iptables -A INPUT -s 0/0 -p tcp --match multiport --dports 507,143 -j ACCEPT

#***** 6
iptables -A INPUT -s 192.168.0.0/16 -p icmp --icmp-type echo-request -j ACCEPT

#***** 7
iptables -A INPUT -p tcp -j REJECT
iptables -A INPUT -p icmp -j REJECT
greg@debian:~$ sudo ./myfirewall.sh
greg@debian:~$ _
```

## 2.1

Για να πετύχουμε να μην έχουμε κανέναν περιορισμό των πακέτων εξόδου χρησιμοποιούμε την εντολή "iptables -A OUTPUT -j ACCEPT". Αυτό ορίζει όλα τα πακέτα στο OUTPUT chain να κάνουν jump στο ACCEPT.

## 2.2

Για να επιτρέψουμε την πρόσβαση ssh μόνο από τις διευθύνσεις του εργαστηρίου δικτύων εκτελούμε την εξής εντολή:

"iptables -A INPUT -p tcp --dport 22 -m iprange --src-range 150.140.139.194-150.140.139.255 -j ACCEPT".

Με αυτόν τον τρόπο πετυχαίνουμε να επιτρέπεται η σύνδεση από τις IP διευθύνσεις του εργαστηρίου, και αργότερα σιγουρευόμαστε ότι επιτρέπεται η σύνδεση μόνο από αυτές στο τελευταίο ερώτημα όπου θα γίνει απάντηση με TCP RST σε οποιαδήποτε άλλη IP, αφού για αυτές η θύρα 22 πρέπει να θεωρείται αποκλεισμένη για αυτές.

Άρα εκμεταλευόμαστε το γεγονός ότι οι κανόνες της αλυσίδας ελέγχονται από πάνω προς τα κάτω στον πίνακα μέχρι να βρεθεί κανόνας που να ταιριάζει.

## 2.3

Για να επιτρέψουμε την πρόσβαση στο ssh από το εσωτερικό δίκτυο εκτελούμε την εξής εντολή:

```
iptables -A INPUT -s 192.168.0.0/16 -p tcp --dport 22 -j ACCEPT
```

Αυτή η εντολή επιτρέπει στο δίκτυο 192.168.0.0/16, δηλαδή το LAN με όλα τα υποδίκτυά του να έχουν πρόσβαση στη θύρα 22. Άρα πετύχαμε τον σκοπό μας.

## 2.4

Για να δώσουμε πρόσβαση στην μια IP διεύθυνση στο HTTPD server έχουμε εκτελέσει την εξής εντολή:

```
iptables -A INPUT -s 192.168.122.1 -p tcp --dport 80 -j ACCEPT
```

Με αυτόν τον τρόπο δίνουμε πρόσβαση στην IP διεύθυνση 192.168.122.1 (σε αυτή τη περίπτωση ο υπολογιστής μου) να έχει πρόσβαση στην θύρα 80 στην οποία τρέχει ο HTTPD.

Για να επιτύχουμε την μοναδικότητα της πρόσβασης της IP διεύθυνσης στη θύρα αυτή αξιοποιούμε το τελευταίο ερώτημα, μέσω του οποίου η θύρα 80 για όλες τις άλλες IP διευθύνσεις θεωρείται αποκλεισμένη.

Άρα αξιοποιούμε όπως παραπάνω το γεγονός ότι οι κανόνες διαβάζονται από πάνω μέχρι κάτω μέχρι να βρεθεί κάποιος που να ταιριάζει.

## 2.5

Για να δώσουμε πρόσβαση σε SMTP over TLS και IMAP (θύρες 587 και 143 αντίστοιχα) εκτελούμε την εξής εντολή:

```
iptables -A INPUT -s 0/0 -p tcp --match multiport --dports 587,143 -j ACCEPT
```

Με αυτόν τον τρόπο όλες οι IP διευθύνσεις (λόγω του 0/0) έχουν πρόσβαση στις δύο θύρες 587 και 143, όπως επιθυμούμε.



## 2.6

Για να αποδεχτούμε τα αιτήματα ICMP Echo από το εξωτερικό δίκτυο εκτελούμε την εξής εντολή:

```
iptables -A INPUT ! -s 192.168.0.0/16 -p icmp --icmp-type echo-request  
-j ACCEPT
```

Άρα όλες οι διευθύνσεις, εκτός από αυτές του τοπικού δικτύου, άρα όλες οι διευθύνσεις εξωτερικού δικτύου μπορούν να στείλουν ICMP πακέτα τύπου echo-request και αυτά να γίνουν αποδεκτές από τον υπολογιστή.

## 2.7

Για να επιτύχουμε την απάντηση με TCP RST ή ICMP μη προσβάσιμο για τα εισερχόμενα αιτήματα για όλες τις αποκλεισμένες θύρες εκτελούμε την εξής εντολή:

```
iptables -A INPUT -p icmp -j REJECT
```