IPTABLES CHEAT SHEET

Τρίτη 10-1-2024

Περιορισμός Εξερχόμενης Πρόσβασης σε Συγκεκριμένη Θύρα

Εντολή: `sudo iptables -A OUTPUT -p tcp --dport 80 -j REJECT`

Εξήγηση: Αυτή η εντολή απορρίπτει όλες τις εξερχόμενες συνδέσεις προς την θύρα 80 (τυπικά χρησιμοποιείται για HTTP). Αυτό σημαίνει ότι το σύστημά σας δεν θα μπορεί να πραγματοποιήσει συνδέσεις σε web servers στην θύρα 80.

Για να δοκιμάσετε τον κανόνα ανοίξτε ένα τερματικό και εκτελέστε:

```
bash
sudo apt-get update
sudo apt-get install curl
```

Τώρα, δοκιμάστε να στείλετε μια αίτηση HTTP σε οποιοδήποτε δημόσιο ιστότοπο. Για παράδειγμα:

```
bash
curl http://www.example.com
```

Δημιουργία Σύνθετων Κανόνων Με Χρήση Πολλαπλών Κριτηρίων

Εντολή: `sudo iptables -A FORWARD -i eth0 -o eth1 -p tcp --syn --dport 22 -m conntrack --ctstate NEW -j ACCEPT`

Εξήγηση: Αυτή η εντολή προσθέτει έναν κανόνα στην FORWARD chain για να επιτρέπει νέες εισερχόμενες συνδέσεις SSH (θύρα 22) από τη διεπαφή eth0 προς τη διεπαφή eth1. Χρησιμοποιεί πολλαπλά κριτήρια, όπως τον τύπο πακέτου (SYN, που είναι χαρακτηριστικό των νέων συνδέσεων TCP), τη θύρα προορισμού, και την κατάσταση της σύνδεσης (NEW).

Αποκλεισμός Εισερχόμενων Συνδέσεων από Συγκεκριμένη IP Διεύθυνση

Στο Ubuntu:

- `sudo iptables -A INPUT -s [IP_Διεύθυνση_Kali] -j DROP`
- Αντικαταστήστε το `[IP_Διεύθυνση_Kali]` με την πραγματική IP διεύθυνση του μηχανήματος Kali Linux.

Έλεγχος από το Kali Linux:

- Δοκιμάστε να κάνετε ping το μηχάνημα Ubuntu: `ping [IP_Διεύθυνση_Ubuntu]`
- Αν ο κανόνας iptables λειτουργεί σωστά, το ping δεν θα λάβει απάντηση.

Πρόσβαση Μόνο Εισερχόμενων SSH Συνδέσεων

Στο Ubuntu:

- `sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT`
- `sudo iptables -A INPUT -j DROP`
- Αυτό θα επιτρέψει μόνο τις εισερχόμενες συνδέσεις SSH και θα απορρίπτει όλες τις άλλες.

Έλεγχος από το Kali Linux:

- Δοκιμάστε να συνδεθείτε μέσω SSH :`ssh [username]@[IP_Διεύθυνση_Ubuntu]`
- Εάν η σύνδεση επιτραπεί, τότε ο κανόνας λειτουργεί σωστά.

Αποκλεισμός Όλων των Εισερχόμενων ICMP (Ping) Αιτημάτων

Στο Ubuntu:

- `sudo iptables -A INPUT -p icmp -j DROP`
- Αυτό θα αποκλείσει όλα τα εισερχόμενα ICMP πακέτα, συμπεριλαμβανομένων των αιτημάτων ping.

Έλεγχος από το Kali Linux:

- Δοκιμάστε να κάνετε ping το μηχάνημα Ubuntu : `ping [IP_Διεύθυνση_Ubuntu]`
- Εάν το ping δεν λαμβάνει απάντηση, τότε ο κανόνας λειτουργεί σωστά.

Περιορισμός Συχνότητας Συνδέσεων

Στο Ubuntu:

1. Εντολή iptables:

```
   bash
1. sudo iptables -A INPUT -p tcp --dport 80 -m limit --limit
   1/minute --limit-burst 5 -j ACCEPT
2. sudo iptables -A INPUT -p tcp --dport 80 -j DROP
```

3. Εξήγηση: Αυτή η σειρά εντολών καθορίζει ότι τα εισερχόμενα πακέτα στη θύρα 80 (HTTP) θα επιτρέπονται με ένα ρυθμό μέχρι 1 πακέτο ανά λεπτό με μια αρχική "έκρηξη" (burst) έως 5 πακέτων. Αυτό σημαίνει ότι μετά τα πρώτα 5 πακέτα, τα επόμενα θα επιτρέπονται μόνο με ρυθμό 1 ανά λεπτό.

Script 1: Έλεγχος Περιορισμού Συχνότητας Συνδέσεων

```
bash
#!/bin/bash

# Διεύθυνση IP του συστήματος Ubuntu
UBUNTU_IP="[IP_Διεύθυνση_Ubuntu]"

echo "Δοκιμή συχνότητας συνδέσεων στην θύρα 80"

# Στέλνουμε αιτήματα προς τη θύρα 80 και παρατηρούμε αν και πότε απορρίπτονται
for i in {1..10}
do
    curl http://$UBUNTU_IP
    echo "Αίτημα $i στάλθηκε"
    sleep 10 # Καθυστέρηση για να ελέγξουμε τον περιορισμό συχνότητας
done
```

Περίπλοκος Συνδυασμός Κανόνων με Χρήση Πολιτικής

Στο Ubuntu:

1. Εντολή iptables:

```
bash
sudo iptables -N MYCHAIN
sudo iptables -A MYCHAIN -s [IP_Διεύθυνση_Kali] -j ACCEPT
sudo iptables -A MYCHAIN -j LOG --log-prefix "Blocked: "
sudo iptables -A MYCHAIN -j DROP
sudo iptables -I INPUT -p tcp --dport 22 -j MYCHAIN
```

6. Εξήγηση: Αυτές οι εντολές δημιουργούν μια νέα αλυσίδα (chain) με όνομα `MYCHAIN` και προσθέτουν κανόνες σε αυτή. Οι εισερχόμενες συνδέσεις στη θύρα 22 (SSH) εντοπίζονται και επεξεργάζονται με βάση τους κανόνες της `MYCHAIN`. Αρχικά, επιτρέπεται η εισερχόμενη κίνηση από την διεύθυνση IP του Kali Linux. Στη συνέχεια, καταγράφεται οποιαδήποτε άλλη προσπάθεια σύνδεσης και τέλος απορρίπτεται.

Script 2: Έλεγχος Πολύπλοκης Αλυσίδας Κανόνων με Συνδυασμό Πολιτικής

```
bash
#!/bin/bash

# Διεύθυνση IP του συστήματος Ubuntu
UBUNTU_IP="[IP_Διεύθυνση_Ubuntu]"

echo "Δοκιμή πολύπλοκης αλυσίδας κανόνων για SSH"

# Προσπάθεια SSH σύνδεσης
ssh [username]@$UBUNTU_IP

# Η επιτυχία ή αποτυχία της σύνδεσης θα δείξει αν οι κανόνες
λειτουργούν σωστά
```

Περιορισμός Εξερχόμενης Κίνησης με Βάση την Ωρα

Στο Ubuntu:

1. Εντολή iptables:

```
bash
sudo iptables -A OUTPUT -p tcp --dport 80 -m time --timestart 08:00 -
-timestop 17:00 -j REJECT
```

Εξήγηση: Αυτή η εντολή απορρίπτει όλες τις εξερχόμενες συνδέσεις προς την θύρα 80 (HTTP) κατά τις ώρες 08:00 έως 17:00. Χρησιμοποιείται για να περιορίσει την πρόσβαση σε διαδικτυακές υπηρεσίες κατά τις ώρες εργασίας.

Script 3: Έλεγχος Περιορισμού Εξερχόμενης Κίνησης με Βάση την Ώρα

```
bash
#!/bin/bash

# Διεύθυνση IP του συστήματος Ubuntu
UBUNTU_IP="[IP_Διεύθυνση_Ubuntu]"

echo "Δοκιμή περιορισμού εξερχόμενης κίνησης με βάση την ώρα"

# Στέλνουμε αίτημα προς τη θύρα 80 και παρατηρούμε αν απορρίπτεται
curl http://$UBUNTU_IP

# Εκτυπώνει την τρέχουσα ώρα για να βεβαιωθούμε ότι το αποτέλεσμα
συμφωνεί με τον κανόνα χρονικού περιορισμού
echo "Τρέχουσα ώρα: $(date)"
```

Για να χρησιμοποιήσετε αυτά τα scripts:

1. Αντικαταστήστε το `[IP_Διεύθυνση_Ubuntu]` με την πραγματική IP διεύθυνση του Ubuntu συστήματός σας.
2. Αντικαταστήστε `[username]` με το όνομα χρήστη που θα χρησιμοποιείτε για τη σύνδεση SSH.

3. Αποθηκεύστε κάθε script σε ένα αρχείο, για παράδειγμα `test_script_1.sh`, `test_script_2.sh`, και `test_script_3.sh`.
4. Δώστε δικαιώματα εκτέλεσης σε κάθε script με την εντολή `chmod +x test_script_x.sh`.
5. Εκτελέστε τα scripts ./ στο Kali Linux για να ελέγξετε τους κανόνες iptables του Ubuntu.

ΠΟΛΙΤΙΚΗ 1 - WORKSTATION

```
# 1: set default DROP policy
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]

# 2: accept any related or established connection
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# 3: allow all traffic on the loopback interface
-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT

# 4: allow outbound DHCP requests
-A OUTPUT -p udp --dport 67:68 --sport 67:68 -j ACCEPT

# 5: allow outbound DNS lookups
-A OUTPUT -p udp -m udp --dport 53 -j ACCEPT

# 6: allow outbound ping requests
-A OUTPUT -p icmp -j ACCEPT

# 7: allow outbound NTP requests
-A OUTPUT -p udp --dport 123 --sport 123 -j ACCEPT

# 8: allow outbound http/https requests
-A OUTPUT -p tcp -m tcp --dport 80 -m state --state NEW -j ACCEPT
-A OUTPUT -p tcp -m tcp --dport 443 -m state --state NEW -j ACCEPT

# 9: allow SMTP
-A OUTPUT -p tcp --dport 25 -m state --state NEW -j ACCEPT

# 10: allow incoming IMAP/IMAPS
-A OUTPUT -p tcp --dport 143 -m state --state NEW -j ACCEPT
-A OUTPUT -p tcp --dport 993 -m state --state NEW -j ACCEPT

# 11: access SSH server
-A OUTPUT -p tcp -m tcp --dport 22 -j ACCEPT

# commit changes
COMMIT
```

Commented [ΔΚ1]: 1.by default, drop all packets for all chains
2.accept, either incoming or outgoing, any packet where its state is either ESTABLISHED (connection is known and tracked) or RELATED (initiated from an already established connection)
3.the loopback interface is *127.0.0.1 or localhost. It's a local one, so accept all connexions to it*
4.allow dhcp requests, from tcp source port 67 or 68 to destination port 67 or 68
5.allow DNS requests to port 53
6.if you want to ping an address, you need this rule
7.if you're using the time protocol to set your clock, you need this rule
8.the core one: only allow *http or https protocol from your box, those initiating the connection (--state NEW)*
9.if you're using a mail software like *Thunderbird to read and send emails, this adds a rule for SMTP server*
10.the same for IMAP/IMAPS servers
11.If you have to access a remote SSH server (for example your AWS ionstance), add this rule

Commented [ΔΚ2R1]: 9. Today, however, SMTP should use port 587 for secure SMTP over TLS.

ΠΟΛΙΤΙΚΗ 2 – Web Server [1]

```
#!/usr/bin/env bash

#
# Common iptables rules for a front-end web server
#
# iptable rules to allow outgoing DNS lookups, outgoing icmp (ping)
requests,
# outgoing connections to configured package servers, outgoing
connections to
# all ips on port 22, all incoming connections to port 22, 80 and 443
and
# everything on localhost.
#
# Source: https://gist.github.com/thomasfr/9712418
#

IPT="/sbin/iptables"

# Server IP
# SERVER_IP="$(ip addr show eth0 | grep 'inet ' | cut -f2 | awk '{
print $2}')"

# Your DNS servers you use: cat /etc/resolv.conf
DNS_SERVER="8.8.4.4 8.8.8.8"

# Allow connections to this package servers
PACKAGE_SERVER="ftp.us.debian.org security.debian.org"

echo "flush iptable rules"
$IPT -F
$IPT -X
$IPT -t nat -F
$IPT -t nat -X
$IPT -t mangle -F
$IPT -t mangle -X

echo "Set default policy to 'DROP'"
$IPT -P INPUT   DROP
$IPT -P FORWARD DROP
$IPT -P OUTPUT  DROP

## This should be one of the first rules.
## so dns lookups are already allowed for your other rules
for ip in $DNS_SERVER
do
    echo "Allowing DNS lookups (tcp, udp port 53) to server '$ip'"
    $IPT -A OUTPUT -p udp -d "$ip" --dport 53 -m state --state
NEW,ESTABLISHED -j ACCEPT
    $IPT -A INPUT  -p udp -s "$ip" --sport 53 -m state --state
ESTABLISHED     -j ACCEPT
    $IPT -A OUTPUT -p tcp -d "$ip" --dport 53 -m state --state
NEW,ESTABLISHED -j ACCEPT
    $IPT -A INPUT  -p tcp -s "$ip" --sport 53 -m state --state
ESTABLISHED     -j ACCEPT
done

echo "allow all and everything on localhost"
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT
```

```
for ip in $PACKAGE_SERVER
do
    echo "Allow connection to '$ip' on port 21"
    $IPT -A OUTPUT -p tcp -d "$ip" --dport 21  -m state --state
NEW,ESTABLISHED -j ACCEPT
    $IPT -A INPUT  -p tcp -s "$ip" --sport 21  -m state --state
ESTABLISHED     -j ACCEPT

    echo "Allow connection to '$ip' on port 80"
    $IPT -A OUTPUT -p tcp -d "$ip" --dport 80  -m state --state
NEW,ESTABLISHED -j ACCEPT
    $IPT -A INPUT  -p tcp -s "$ip" --sport 80  -m state --state
ESTABLISHED     -j ACCEPT

    echo "Allow connection to '$ip' on port 443"
    $IPT -A OUTPUT -p tcp -d "$ip" --dport 443 -m state --state
NEW,ESTABLISHED -j ACCEPT
    $IPT -A INPUT  -p tcp -s "$ip" --sport 443 -m state --state
ESTABLISHED     -j ACCEPT
done


######################################################################
################################
## Global iptable rules. Not IP specific

echo "Allowing new and established incoming connections to port 21,
80, 443"
$IPT -A INPUT  -p tcp -m multiport --dports 21,80,443 -m state --
state NEW,ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -p tcp -m multiport --sports 21,80,443 -m state --
state ESTABLISHED     -j ACCEPT

echo "Allow all outgoing connections to port 22"
$IPT -A OUTPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j
ACCEPT
$IPT -A INPUT  -p tcp --sport 22 -m state --state ESTABLISHED     -j
ACCEPT

echo "Allow outgoing icmp connections (pings,...)"
$IPT -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED,RELATED -j
ACCEPT
$IPT -A INPUT  -p icmp -m state --state ESTABLISHED,RELATED     -j
ACCEPT

echo "Allow outgoing connections to port 123 (ntp syncs)"
$IPT -A OUTPUT -p udp --dport 123 -m state --state NEW,ESTABLISHED -j
ACCEPT
$IPT -A INPUT  -p udp --sport 123 -m state --state ESTABLISHED     -j
ACCEPT

# Log before dropping
$IPT -A INPUT  -j LOG  -m limit --limit 12/min --log-level 4 --log-
prefix 'IP INPUT drop: '
$IPT -A INPUT  -j DROP

$IPT -A OUTPUT -j LOG  -m limit --limit 12/min --log-level 4 --log-
prefix 'IP OUTPUT drop: '
$IPT -A OUTPUT -j DROP
```

```
exit 0
```

- Που είναι το λάθος -> port 80
- Στην περίπτωση του Web Server σε αντίθεση με την περίπτωση του workstation,

- `$IPT -A INPUT  -p tcp -m multiport --dports 21,80,443 -m state --state NEW,ESTABLISHED -j ACCEPT`
- `$IPT -A OUTPUT -p tcp -m multiport --sports 21,80,443 -m state --state ESTABLISHED     -j ACCEPT`

[1] https://jonlabelle.com/snippets/view/shell/iptables-firewall-rules-for-web-server

Koutras Dimitris – dkoutras@unipi.gr

10-01-2024