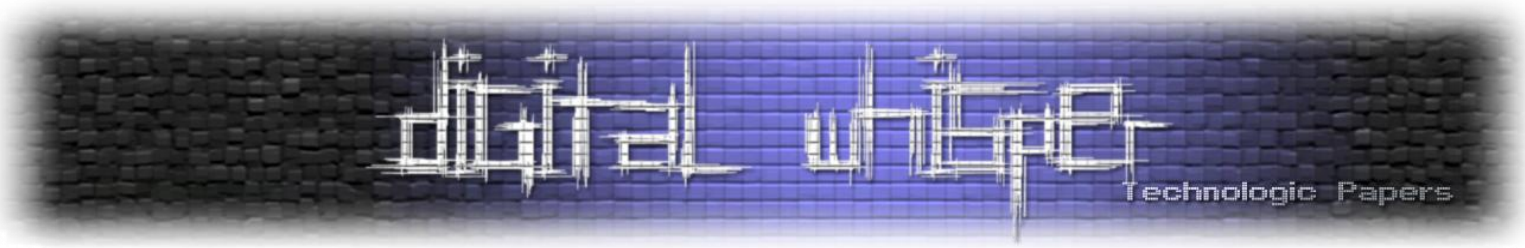


IP Tables

מאת אפיק קסטיאל (cp77fk4r)

הקדמה

IPTables מהווה כלי רב עוצמה המשתמש בעיקר כתוכנית Firewall מבוססת אירועים ומאופיינת כ-Stateful Firewalling (טכנולוגיה המבוססת על בדיקת תוכן/כותרות ה-Packets), במערכות הלינוקס למינהן. הרעיון מאחורי ה-IPTables מאוד פשוט, בעזרתה נוכל לקבוע מה יעלה בגורל כל Packet שיגיע למחשבינו, בשל גמישותה נוכל להבדיל כמעט בין כל Packet ו-Packet, בין אם מדובר בזיהוי מקורו/יעדו, סוגו/תפקידו, או כמעט כל מאפיין שאפשר לחשוב עליו.

הסבר כללי

הכוונה ב-"לקבוע מה יעלה בגורל כל Packet" היא בעצם אישור כניסה ל-Packet, התעלמות מה-Packet, התעלמות ושליחת הודעת שגיאה לשולח ה-Packet וכו', העברת ה-Packet ליעד אחר וכו'.

הרעיון המרכזי הוא שבעזרת מכלול חוקים, כלליים או נקודתיים, נוכל לקבוע איך תצורת תעבורת הרשת מהמחשב שלנו ואל המחשב שלנו תתנהג, וכך למנוע מחבילות מידע "זדוניות" או מכל מני פורענויות לחדור למחשב שלנו.

כיום, בקרנלים (עוד מ-2.4.X) מובנית טבלה שנקראת Filtering Table, דרך הטבלה הזאת המערכת קובעת מה יעלה בגורל Packet המגיע למערכת על ידי מערכת חוקים הכתובים בה.

מבנה המערכת

אפליקצית ה-IPTables הגיעה לעולם כחלק מפרוייקט NetFilter של Rusty Russell מ-Core Team המופץ תחת GPL, היא כלי אשר נועד לנהל בצורה נוחה את ה- Filtering Table. אפליקציה זו מבוססת על מספר טבלאות/שרשראות-חוקים עקריות אשר בעזרתן נוכל לנהל את תצורת תעבורת ה-Packets במערכת שלנו.

הטבלאות הן:

- **FILTER** – טבלאת ברירת המחדל, הטבלה הכי בסיסית אם לא תקבע שום טבלה שתוגדר כאחראית לטיפול באירוע, ה-Packet יגיע לכאן.
  בטבלה הזאת קיימות שלוש שרשראות:

  - **INPUT**- שרשרת המוגדרת לטפל ב-Packets אשר נכנסים למערכת.
  - **OUTPUT** - שרשרת המוגדרת לטפל ב-Packets אשר יוצאים מהמערכת.
  - **FORWARD** - שרשרת המוגדרת לטיפול ב-Packets המיועדים לניתוב.

- **NAT** - טבלאת הניתוב, הטבלה האחראית לניתוב ה-Packets וקיימות בה שלוש שרשראות:
  - **PREROUTING** - שרשרת המוגדרת לטפל ב-Packets לפני הניתוב.
  - **POSTROUTING** - שרשרת המוגדרת לטפל ב-Packets לאחר הניתוב.
  - **OUTPUT** - שרשרת המוגדרת לטפל ב-Packets היוצאים.

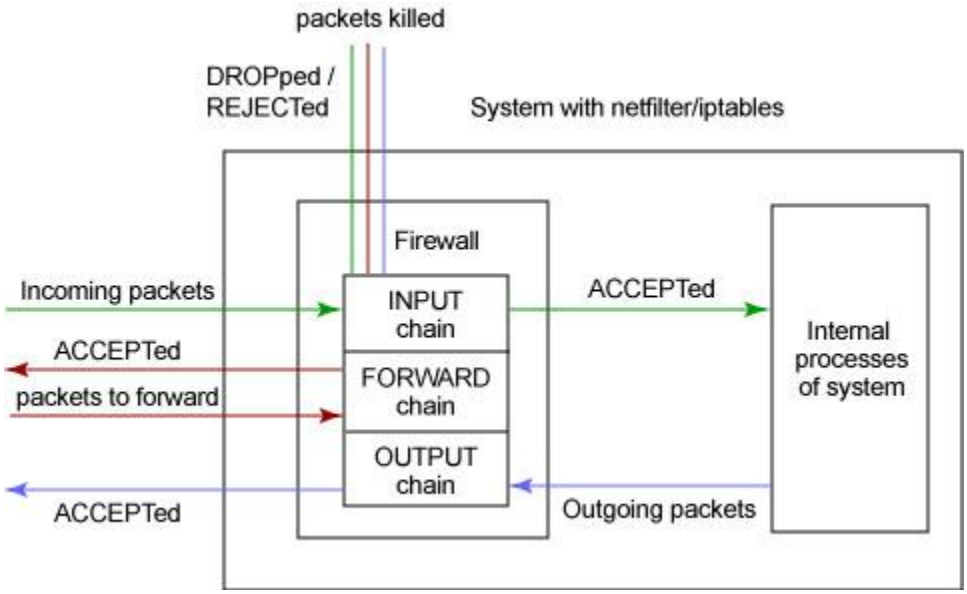
- **MANGLE** - טבלה לטיפול מתקדם ב-Packets. בקרנלים מ-2.4.18 קיימות חמש שרשראות:
  - **PREROUTING** - שרשרת המוגדרת לטפל ב-Packets לפני הניתוב.
  - **POSTROUTING** - שרשרת המוגדרת לטפל ב-Packets לאחר הניתוב.
  - **INPUT** - שרשרת המוגדרת לטפל ב-Packets אשר נכנסים למערכת.
  - **OUTPUT** - שרשרת המוגדרת לטפל ב-Packets היוצאים.
  - **FORWARD** - שרשרת המוגדרת לטיפול ב-Packets המיועדים לניתוב.

כמו שראינו, כל טבלה מכילה מספר שרשראות, וכל שרשרת יכולה להכיל מספר חוקים.

למרות שהחוקים דומים בכל הטבלאות, והעקרונות אף הם דומים למדי, במאמר זה אגע רק בשרשרת בעלת השימוש הנפוץ ביותר בטבלת ה-**FILTER** - שרשרת ה-**INPUT.**

תפקוד

כאשר מגיע Packet למערכת ה-IPTables (מבפנים או מבחוץ), מערכת ה-IPTables מזהה לאיזה אירוע מתאים ה-Packet, Packet יוצא ישלח לשרשרת ה-OUTPUT, Packet נכנס ישלח לשרשרת ה-INPUT.



(התמונה נלקחה מ-DeveloperWorks של IBM)

לאחר שזוהתה השרשרת בטבלה אליה ה-Packet מתאים, המערכת עוברת חוק חוק (בסדר מהעליון לתחתון) בשרשרת חוקים ובודקת האם קיים חוק העונה ל-Packet הספציפי, לאחר שהיא מוצאת חוק שמאפייני ה-Packet עונים לו, תפסיק המערכת בחיפוש אחר החוקים ותבצע את מה שהחוק אומר. לפיכך חשוב לשים לב טוב טוב לסדר בו כותבים את החוקים.

אם חוק אחד אומר **לקבל כל Packet** המשתמש בפרוטוקול TCP, ואחריו יש חוק האומר **שאין לקבל שום Packet**, ה-Packet **יכנס למערכת ללא בעיה**. לעומת זאת, אם קיים מצב ובו יש חוק האומר למערכת **לא לקבל שום Packet** ואחריו קיים חוק האומר **לקבל כל Packet מכתובת IP מסויימת**- שום Packet **לא יתקבל**. חשוב מאוד לשים לב לסדר כתיבת החוקים ולזכור כי לאחר שיימצא החוק **הראשון** המתאים ל-Packet, רק הוא יתבצע!

עבור כל Packet שמגיע נוכל לבחור כיצד המערכת תגיב מתוך חמשת התגובות הבאות:

- **ACCEPT**- המערכת תאפשר ל-Packet שהתקבל לעבור.
- **DROP**- המערכת לא תתייחס ל-Packet שהתקבל.
- **REJECT**- המערכת לא תתייחס ל-Packet שהתקבל אך תשלח שגיאה למקור ה-Packet.
- **QUEUE**- המערכת תעביר את חבילת המידע לשימוש ב-Userspace.
- **RETURN**- המערכת תחזור אחורה לחוק שממנו הופעל החוק הנוכחי.

מתחילים

בכדי להבין כיצד לכתוב או לקרוא את החוקים הנמצאים בטבלאות, נשתמש במקרים נפוצים (יותר ופחות) ונראה איך ניתן להגיב אליהם וכיצד ליישם את התגובה בעזרת ה-IPTables.

טבלת ברירת המחדל שלנו היא filter ולכן, אם לא נציין שום טבלה, המידע יכנס או ישלף מהטבלה הזאת.

תוכן הטבלאות

דבר ראשון, כתבו בקונסול:

```
iptables -t filter -L INPUT
```

הפקודה הנ"ל אומרת ל-IPTable להציג את כל החוקים שקיימים בשרשרת INPUT בטבלת filter.
שימו לב שאם תכתבו:

```
iptables -L INPUT
```

תקבלו את אותה התוצאה, היות ו-filter היא ברירת המחדל.

אם לא נגעתם ב-iptables עד עכשיו, מרבית הסיכויים הם שתראו את הפלט הבא:

```
Chain INPUT (policy ACCEPT)
target            port opt source   destination
```

הטבלה ריקה, ואלה ראשי העמודות:

- **TARGET** - מה יעלה בגורל ה-Packet (לאן הוא ישלח)
- **PORT** – באיזה פורט משתמש הפקט.
- **OPT** – אפשרויות שונות בהן נגע בהמשך.
- **SOURCE** – מאיפה נשלח ה-Packet.
- **DESTINATION** – לאן הוא נשלח.

במרבית המקרים המערכת תפלוט לכם כי אין לכם גישה ל-IPTables, במקרים כאלה תאלצו להתחבר לחשבון בעל הרשאות מערכת, או לבצע את הכל תחת sudo.

כתיבת החוקים

כל חוק שנכתוב, אם לא נאמר בפירוש לאיזה שורה אנחנו רוצים לכתוב אותו, יכתב בסוף השרשרת.

חוק ראשון- אנחנו רוצים לנתק את כל תעבורת האינטרנט הנכנסת למחשב, כיתבו:

```
Iptables –A INPUT –j DROP
```

והנה פירוט של מה שעשינו, הצבענו על השרשרת INPUT בעזרת:

```
-A INPUT
```

וקבענו את גורל ה-Packets שמגיעים לשם, בעזרת:

```
-j DROP
```

*לא קבענו שום סינון ולכן החוק יחול על כל ה-Packets.
שימו לב, בעזרת המתג **j –** נקובעים מה יעלה בגורל ה-Packets שמקיימים את אותו החוק, במקרה שלנו קבענו DROP, יכולנו לקבוע כל גורל אחר.
כדי לבחון איך החוק נשמר, נכתוב שוב:

```
iptables -L INPUT
```

ונקבל את הפלט הבא:

```
Chain INPUT (policy ACCEPT)
target            port  opt   source            destination
DROP              0     --    anywhere          anywhere
```

שימו לב:
קבענו רק מה יעלה בגורל ה-Packet ולא קבענו שום מסננים, לכן תחת TARGET הוכנס ה-DROP ותחת כל השאר- הוכנס anywhere (פורט 0 אומר "כל הפורטים").

איך נבדוק שהחוק באמת פועל? שלחו פינג לגוגל ותראו מה קורה, כתבו:

```
 ping www.google.com
```

לאחר שיעבור ה-timed_out, נקבל:

```
ping: unknown host www.google.com
```

הבה נמחק חוק זה ונמשיך הלאה, כיתבו:

```
Iptables –D INPUT 1
```

ושוב, כיתבו:

```
iptables -L INPUT
```

ונראה שוב- טבלה ריקה. שימו לב שבעזרת:

-D INPUT 1

אמרנו ל-IPTables שאנחנו רוצים למחוק את חוק מספר 1 משרשרת INPUT. פשוט ביותר.
עכשיו נשלח שוב פינג לגוגל רק בשביל לבדוק שהכל פועל כשורה:

```
ping www.google.com
```

ונוכל לראות לפי הפלט שאנחנו אכן מקבלים echo מגוגל:

```
PING www.l.google.com (74.125.39.106) 56(84) bytes of data.
64 bytes from fx-in-f106.google.com (74.125.39.106): icmp seq=1 ttl=241 time=73.4 ms
64 bytes from fx-in-f106.google.com (74.125.39.106): icmp_seq=2 ttl=241 time=74.6 ms
64 bytes from fx-in-f106.google.com (74.125.39.106): icmp_seq=3 ttl=241 time=72.3 ms
64 bytes from fx-in-f106.google.com (74.125.39.106): icmp_seq=3 ttl=241 time=74.5 ms

64 bytes from fx-in-f106.google.com (74.125.39.106): icmp seq=1 ttl=241 time=76.0 ms
--- www.l.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4003ms
Rtt min/avg/max/mdev = 72.324/74.211/76.076/1.298 ms
```

אגב, מומלץ לעבוד עם Packet Sniffer על מנת להבין טוב יותר בדוגמאות הבאות.
כעת נראה כיצד ניתן לאפשר רק ל-Packet בעל אופי מסויים להכנס למחשב, נניח, רק Packets מפורט 80 יוכלו להכנס למערכת שלנו, וכל השאר ילכו לפח. כך נאפשר רק לגלוש באינטרנט, אבל לא להשתמש בשאר התוכנות, כמו ICQ, IRC, FTP וכו', כיתבו:

```
iptables -A INPUT -p tcp --sport 80 -j ACCEPT
```

החוק די מובן: "Packets על-גבי פרוטוקול TCP מ-80 יכנסו למערכת". בעזרת:

```
-p tcp --sport 80
```

קבענו שהמאפיין שיבדק על ידי החוק שלנו הוא source-port של ה-Packet, ואם הוא יהיה שווה ל-80 על גבי פרוטקול ה-TCP נקבל את ה-Packet.

נסו להכנס עם הדפדפן לאיזה אתר, האם הכל עובד? הפעילו את קליינט ה-IRC האהוב עליכם, ו-גם עובד! מוזר, לא? בואו נראה מה עשינו, אמרנו למערכת שתאפשר ל-Packets מפורט 80 להכנס, אך מה בדבר שאר ה-Packets? הם לא מקיימים שום חוק בשום פילטר ולכן הם עוברים. מה שאנחנו צריכים לעשות זה להוסיף את השורה הבאה:

```
Iptables –A INPUT –j DROP
```

את השורה הזאת אנחנו מכירים, היא חוסמת את כל ה-Packets הנכנסים למערכת שלנו.
רק רגע.. זה לא מתנגש עם החוק הקודם? זה מתנגש בהחלט, אבל כמו שאמרתי בתחילת המאמר- המערכת תחפש את החוק **הראשון** שמאפייני הפאקים מקיימים ואחרי שהיא תמצא אחד כזה, היא תפסיק לחפש ותבצע את הפעולה, מה שאומר שאם מגיעים למערכת שלנו Packets מפורט 80, הם יענו על החוק הראשון ולכן הם יגיעו אלינו. לעומת זאת, Packets שלא מגיעים מפורט 80 לא יענו על החוק

הראשון, ולכן המערכת תבדוק האם החוק השני חל עליהם- אנחנו כבר יודעים שהחוק השני חל על **כל ה-Packets הנכנסים** ולכן מאפייני ה-Packet אכן יענו עליו- מה שיוביל לכך ש המערכת תבצע את מה שהחוק אומר- להתעלם ממנו. כיתבו:

```
iptables -L INPUT
```

אתם אמורים לקבל:

```
Chain INPUT (policy ACCEPT)
target      port        opt         source      destination
ACCEPT      tcp         anywhere    anywhere    tcp spt:www
DROP        0           --          anywhere    anywhere
```

נניח ואנחנו רוצים להוסיף עוד חוק שיקבע שמהמחשב שלנו יהיה אפשר להתחבר גם לשרתי IRC, רובם כיום משתמשים בפורט 6667. אנחנו מכירים את הפקודה לאפשר ל-Packet מפורט מסויים להכנס למערכת, אנחנו רק צריכים לשנות אותה ל-6667:

```
iptables -A INPUT -p tcp --sport 6667 -j ACCEPT
```

אבל אנחנו גם יודעים שכל פעם שאנחנו מכניסים חוק חדש לטבלה, אותו החוק הוא יכנס לסוף הטבלה, כך שאם טבלת ה-INPUT שלנו נראת ככה:

```
target      port        opt         source              destination
ACCEPT      tcp         anywhere    anywhere            tcp spt:www
DROP        0           --          anywhere            anywhere
```

אנחנו יודעים שהחוק שנקבע לאפשור Packets בפורט 6667 לא יהיה בר תוקף היות והוא יכנס לאחר החוק שאומר לזרוק את כל ה-Packets. לפיכך, אנחנו צריכים להכניס אותו לפני אותו החוק, בשורה הראשונה או השניה, אך בשום אופן לא בשורה השלישית.

כיצד כותבים חוק לשורה ספציפית? פשוט מאוד:

```
iptable –I INPUT 1 –p tcp --sport 667 –j ACCEPT
```

החלפנו את: **-A INPUT** שאומר "Append to chain" ב: **-I INPUT 1** שאומר "Insert in chain as rulenum" – וקבענו ש-rulenum יהיה שווה ל-1, משמעות הדבר שהשורה תכנס כשורה ראשונה ותוריד את שאר השורות הקיימות למטה. כעת, אם נסתכל בטבלה שלנו, נראה שהיא בנוייה באופן הבא:

```
target      port  opt   source      destination
ACCEPT      tcp         anywhere    anywhere    tcp spt:ircd
ACCEPT      tcp         anywhere    anywhere    tcp spt:www
DROP        0     --    anywhere    anywhere
```

מה שאומר שרק Packets שיכנסו מה-ircd (6667) וה-www (פורט 80) יכנס למערכת, ולכל השאר המערכת תבצע DROP.
כעת, אם נרצה לשכתב שורה קיימת בטבלה, נחליף את המתג I במתג R, והרעיון אותו רעיון- כותבים את מספר השורה שאותה רוצים לשכתב.

נניח ואנחנו לא רוצים לאפשר 6667 אלא לאפשר גישת SSH למחשב שלנו. SSH רץ לרוב על פורט 22, כך שעלינו רק לשנות בשורה את 6667 ל-22, השורה של 6667 (ה-ircd) היא השורה הראשונה, אז נכתוב את הפקודה באופן הבא:

```
iptables -R INPUT 1 -p tcp --sport 22 -j ACCEPT
```

כעת, אם נסתכל על הטבלה, נראה שהיא עודכנה:

```
target            port   opt    source              destination
ACCEPT      tcp           anywhere     anywhere     tcp spt:ssh
ACCEPT      tcp           anywhere     anywhere     tcp spt:www
DROP        0      --     anywhere     anywhere
```

ב-spt, שונה הערך מ-ircd, ל-ssh, מה שיאפשר לכל משתמש שירצה לגשת לפורט 22.
נעבור לעוד אפשרות- מי אמר שאנחנו רוצים לאפשר לכל אחד לגשת לפורט 22? אולי נרצה להגדיר רק לכתובת IP מסויימת או לטווח כתובות IP? (מאוד לא בריא לאפשר לכל אחד לגשת ל-ssh אפילו אם נדרשת סיסמה בכדי להכנס). כשמשתמשים באפשרות הזאת, יש לזכור כי כתובות IP הן לא סטטיות ומשתנות בכל פעם שמתבצע חיוג ל-ISP. נכון שכיום גובר השימוש ב-Static IP או בחיבורי DHCP, אבל עדיין, צריך לזכור את זה כאשר משתמשים באימות מבוסס IP.

מחקו את כל החוקים שכתבנו עד עכשיו בשרשרת, על ידי:

```
Iptables -F INPUT
```

(אם לא נכתוב את שם השרשרת באופן שמצויין כאן, פקודה זו תמחק את כל החוקים הקיימים בכל השרשראות, אז שימו לב!)

כיתבו את החוק הבא:

```
iptables -A INPUT -s XXX.XXX.XXX.XXX -p tcp --sport 22 -j ACCEPT
```

אנחנו כבר מכירים כמעט את כל המתגים המרכיבים את החוק הזה, הדבר היחיד שהוספנו הוא:

```
-s XXX.XXX.XXX.XXX
```

כפי שבוודאי הבנתם, מדובר במאפיין ה-Source IP – כתובת ה-IP (או הדומיין) שממנה נשלח אלינו ה-Packet. כמובן שחוק זה לבדו לא יעזור לנו היות ובעצם לא נמנעה שום גישה לשאר האנשים, לכן יש להוסיף גם את החוק הבא:

```
iptables -A INPUT -p tcp --sport 22 -j DROP
```

אם נרצה לאפשר ל-SUBNET ספציפי, נוכל לכתוב את החוק באופן זה:

```
iptables -A INPUT -s XXX.XXX.XXX.XXX/YYY -p tcp --sport 22 -j ACCEPT
```

אפשרות זו כבר מוכרת לנו - זרוק לפח כל נסיון גישה לפורט 22 על גבי פרוטוקול ה-tcp.
חשוב לציין כי ניתן לבצע את הפעולה הזאת גם על ידי חוק אחד בלבד:

```
iptables -A INPUT -s ! XXX.XXX.XXX.XXX -p tcp --sport 22 -j DROP
```

נראה שהוא לבד יכול לבצע את הסינון, דבר ראשון, הוספנו סימן קריאה לפני כתובת ה-IP, מה שאומר כאן כמו כמעט בכל שפות התיכנות- "NOT", ודבר שני, שינינו את גורל ה-Packet ל-"DTOP". חוק זה אומר "זרוק לפח כל Packet שמאפיין ה-Source-IP שלו **לא שווה** ל:XXX.XXX.XXX.XXX שמנסה לגשת לפורט 22 על גבי הפרוטוקול TCP." לפיכך, רק ה-IP שקבענו יוכל לגשת לפורט 22 (כי הוא לא מתאים למאפיין שבפילטר) ושאר האנשים- לא.

באופן כללי אין נכון או לא נכון בכתיבת חוקים, אם החוק מבצע את תפקידו כהלכה הוא נכון, אך כל משתמש אוהב ורואה לנכון לבצע מספר פעולות בדרכים שונות.

מאפייני ה-Packet שנגענו בהם עד כה הם:

- **p** – ה-**Protocol** בו משתמש ה-Packet.
- **sport** – ה-**Source Port** ממנו הגיע ה-Packet.
- **s** – ה-**Source IP/Domain** ממנו הגיע ה-Packet.

קיימים מספר מאפיינים נוספים הנוגעים לטבלת ה-**INPUT** שבהם אפשר להשתמש:

- **dport** - מאפיין המצביע על ה-**Destination Port** – הפורט אליו נשלח ה-Packet.
- **i** - ממשק (**In interface**), כרטיס הרשת ממנו הגיע ה-Packet (ppp, wlan, eth וכו').
- **syn** - מאפיין הבודק האם ה-Packet מאופיין כ-**syn-packet**.

בכדי להמחיש דוגמא לשימוש ב-**i** וב-**syn** נשתמש במקרה בו אנחנו לא רוצים שמחשבים חיצוניים יכלו לפתוח חיבור איתנו דרך eth1, אך בכל זאת רוצים לאפשר לעצמנו להתחבר אליהם. משמעות הדבר היא שאם אנחנו נבקש ליצור חיבור- הוא יתאפשר, אך אם מישהו חיצוני יבקש ליצור חיבור הוא ידחה על ידי המערכת. נכתוב את השורה הבאה:

```
iptables -A INPUT -p tcp -i eth1 --syn -j DROP
```

חוק זה הוא פשוט להבנה: כל syn שיגיע מרכיב ה-eth1 יזרק לפח. שימו לב שהחוק שונה מהחוק הראשון שלמדנו:

```
iptables –A INPUT –j DROP
```

החוק הראשון מונע מכל Packet לעבור ולכן תעבורת הרשת הנכנסת למערכת שלנו תחסם. אם נרצה, נכון להוציא מידע מהמחשב נוכל, אך לא נוכל לקבל את התוצאות מפני שחסמנו את כל תעבורת הרשת

פנימה. לעומת זאת, בחוק הנוכחי תעבורת האינטרנט פנימה לא תחסם, אלא רק Packets אשר נועדו להגיד למערכת שהם מעוניינים לפתוח חיבור TCP.

נגענו מספיק בטבלת ה-INPUT, הבה נראה קצת דוגמאות מטבלת ה-OUTPUT. הרעיון הכללי הוא אותו רעיון וברוב הדגלים משתמשים בלא שום שינוי. לדוגמא, חסימת כל תעבורת הרשת החוצה, תתבצע ע"י הפקודה הבאה:

```
iptables –A OUTPUT –j DROP
```

חסימת כל תעבורת הרשת היוצאת, חוץ מפורט 25 (SMTP לשליחת דואר בין מערכות) תתבצע כך:

```
iptables -A OUTPUT -p tcp --dport ! 25 -j DROP
```

שימו לב, שכשמדובר בתעבורה יוצאת, משתמשים ב-dport, שמצביע על Destination-Port, ולא ב-sport שמצביע על Source Port.

חסימת ה-Packets היוצאים מממשק eth1 שלנו, תתבצע עם שינוי דומה:

```
iptables -A OUTPUT -p tcp -o eth1 -j DENY
```

שוב, שימו לב שהשימוש בממשק הרשת בוצע על ידי המתג: **o-** שאומר "Out Interface", ולא על ידי: **i-** שאומר "In Interface".

כמה דברים שחשוב לדעת

אם נרצה לעקוב אחרי Packets מסויימים, נוכל להגדיר ל-IPTables לשמור ב-LOG Packets ספציפים. לדוגמא, אם נרצה לבצע מעקב אחרי כל לקוחות שירות ה-SSH שהתקנו על השרת, נדע שלשם כך אנחנו צריכים לשמור את המידע שמגיע אלינו בפורט 22 על גבי ה-TCP ונבצע זאת על ידי הפקודה הבאה:

```
Iptables -A INPUT -tcp --sport 22 -j LOG --log-prefix "SSH CONNECTION"
```

אין הרבה חדש בפקודה זו, השורה לא קובעת אם ה-Packet יתקבל או לא יתקבל, אלא פשוט אומרת למערכת לשמור את המידע הזה בקובץ בכדי שבשלב מאוחר יותר נוכל לראות מה בדיוק קרה במהלך ההתחברות.

קבענו ש"גורל ה-Packet" ילך ל-LOG, החלק של ה-"log-prefix--" ומה שמגיע אחריו בגרשיים יכנס כהערה בשורה בכדי שנדע בזמן ניתוח הלוג מה שייך למה.

כברירת המחדל של ה-IPTables, קבצי הלוגים נשמרים במיקום:

```
/var/log/messages
```

אך אם תרצו לשנות את המיקום בו ישמרו הקבצים, תוכלו לגשת ל:

```
kate etc/syslog.conf
```

ותוסיפו את השורה הבאה:

```
kern.warning /var/log/your_file
```

כמובן שתאלצו להפעיל מחדש את מנגנון ה-syslog. אני משתמש ב-Ubuntu, הפקודה היא:

```
/etc/init.d/sysklogd restart
```

ב-Fedora, Redhat ודומיהן, הפקודה קצת שונה:

```
/etc/init.d/syslog restart
```

חשוב לזכור כי אם נכבה את המחשב, כל החוקים שכתבנו ימחקו . כדי לשמור את החוקים שכתבנו נפעל כך: נשמור את ההגדרות שכתבנו בקובץ חיצוני:

```
iptables-save > your-rules-file
```

לאחר מכן נקבע למערכת להפעיל את הפקודה:

```
ptables-restore < your-rules-file
```

בכל פעם שהמערכת עולה, על ידי הוספת הפקודה הזאת, לקובץ: **etc/init.d/networking** בסוף הסקטור **start**. דרך זו אמנם אינה אידיאלית או הכי נוחה, אך היא ללא ספק היעילה ביותר.

סיכום

עד כאן הפרק הראשון בסדרת מאמרים זו. כפי שראיתם, נגענו רק בקצה המערכת ועדיין השגנו ידע המאפשר לנו לקנפג את מערכת ה-Filtering Table בצורה שתוסיף די אבטחה לשרת/מחשב שלנו, בהמשך הסידרה נגע בטבלאות מתקדמות יותר ומתוחכמות מ-INPUT.

נקודה אחרונה: שימו לב שיש מחרוזות שנכתבו באותיות גדולות, כמו למשל: ACCEPT, DENY, INPUT, ויש מחרוזות שנכתבו באותיות קטנות, כמו למשל: sport, tcp. הרעיון הוא נובע מכך, כפי שאתם כנראה יודעים, שמספר מערכות, וביניהן לינוקס, רגישות לאותיות קטנות ולאותיות גדולות (Case Sensitive), כך ש-ACCEPT לא שווה ל-Accept או ל-AccepT. לפיכך יש תמיד לזכור מה כותבים באותיות קטנות או גדולות, זכרון זה מגיע עם תרגול ועם הזמן.

לקריאה נוספת

- **האתר של הפרוייקט Netfilter, כולל עידכונים, כלים ועוד הרבה:** http://www.netfilter.org/
- **מדריך ענקי של Oskar Andreasson על IPTables:** http://iptables-tutorial.frozentux.net/iptables-tutorial.html
- **ממשק ויז'ואלי ל-IPTables התומך בהרבה מאוד רכיבים:** http://www.fwbuilder.org/