

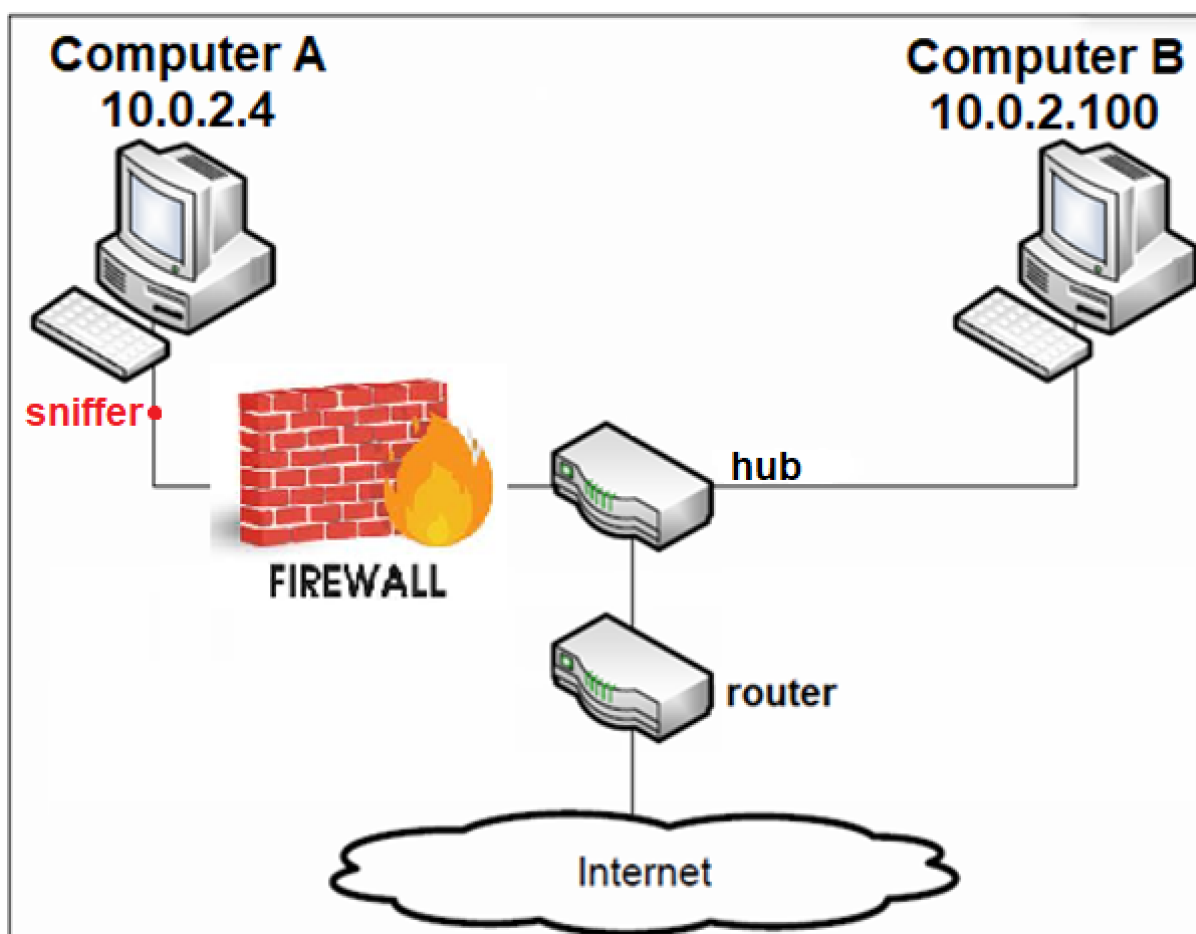
Linux Firewall Exploration Lab

כתובות IP לכל מחשב

Name	IP	MAC
Computer A	10.0.2.4	-
Computer B	10.0.2.100	-

Firewall

התמונה מטה מייצגת את מבנה הרשת



Task 1: Using Firewall

מבוא:

תיאור

במשימה זו נגדיר חוקים שונים לטבלאות ה-FIREWALL

מטרה

לבטל אפשרות ליצירת חיבור TELNET בין שני מחשבים, וחסידת אפשרות גישה לאתר ספציפי דרך מחשב מסוים.

תוצאה מצופה

כאשר ננסה ליצור חיבור TELNET או ננסה לגשת לאתר החסום נכשל ולא נקבל חיבור מוצלח.

Prevent A from doing telnet to Machine B

תחילה בדקנו שטבלת החוקים של חומת האש ריקה

```
[Tue May 02 20:25:37] Computer A:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

ניתן לראות שאין חוקים מוגדרים בטבלת מחשב A
INPUT – טבלה האחראית על הכנסת מידע למחשב
FORWARD – טבלה האחראית על העברת המידע למחשב אחר
OUTPUT – טבלה האחראית על הוצאת המידע מהמחשב

כעת נרצה לבדוק שאכן ניתן ליצור חיבור TELNET בין מחשב A למחשב B

```
[Tue May 02 20:26:07] Computer A:~$ telnet 10.0.2.100
Trying 10.0.2.100...
Connected to 10.0.2.100.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Mar  5 15:59:58 IST 2023 from 10.0.2.4 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

ניתן לראות שהחיבור נוצר בהצלחה

כעת נרשום את הפקודה הבאה במחשב A כדי לחסום את האפשרות ליצור חיבור TELNET עם מחשב B

```
[Tue May 02 20:40:13] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 23 -s 10.0.2.4 -d 10.0.2.100 -j DROP
```

A – הוספת חוק חדש
OUTPUT – הטבלה אליה אנחנו מוסיפים את החוק
ק – פרוטוקול התקשורת של TELNET אותו אנחנו רוצים לחסום
--dport – פורט היעד אליו נשלח הפאקט
-s – מחשב הSOURCE ממנו נשלח הפאקט
-d – מחשב הdestination אליו מיועדת הפאקט
-j – הפעולה אותה נרצה לבצע על הפאקט (jump)
DROP – הפעולה שמתבצעת על הפאקט היא זריקה

כעת נבדוק שאכן הפקודה נוספה לטבלת הOUTPUT

```
[Tue May 02 20:42:35] Computer A:~$ sudo iptables -L Chain INPUT (policy ACCEPT) target      prot opt source      destination Chain FORWARD (policy ACCEPT) target      prot opt source      destination Chain OUTPUT (policy ACCEPT) target      prot opt source      destination DROP        tcp  --  10.0.2.4      10.0.2.100 tcp dpt:telnet
```

ניתן לראות בטבלה הOUTPUT את החוק החדש שאומר לזרוק פאקטות אשר מועברות מ-IP 10.0.2.4 מחשב A ל-IP 10.0.2.100 מחשב B בתקשורת TCP.TELNET

כעת ננסה ליצור חיבור ממחשב A למחשב B בTELNET

```
[Tue May 02 20:40:34] Computer A:~$ telnet 10.0.2.100 Trying 10.0.2.100...
```

ניתן לראות שמחשב A מנסה ליצור חיבור אך ללא הצלחה, בדקנו גם בWIRESHARK וראינו שהפאקט לא הועברה לכן החיבור לא יצליח לעולם.

Prevent B from doing telnet to Machine A

כעת ביצענו את אותן הפעולות גם במחשב B רק שהפכנו בפקודה את הIP של המקור ושל היעד וכך חסמנו את אופציית יצירת חיבור TELNET בין מחשב B למחשב A

```
[Tue May 02 20:41:32] Computer B:~$ sudo iptables -A OUTPUT -p tcp --dport 23 -s 10.0.2.100 -d 10.0.2.4 -j DROP
```

לאחר מכן ביצענו את אותן הבדיקות וראינו שאכן החוק נוסף לטבלה, ושלא ניתן ליצור חיבור Telnet בין המחשבים.

```
[Tue May 02 20:56:55] Computer B:~$ telnet 10.0.2.4
Trying 10.0.2.4...
```

התוצאות היו זהות.

כמו כן, יכולנו לרשום את הפקודה הבאה ממחשב A ולא לעבור למחשב B:

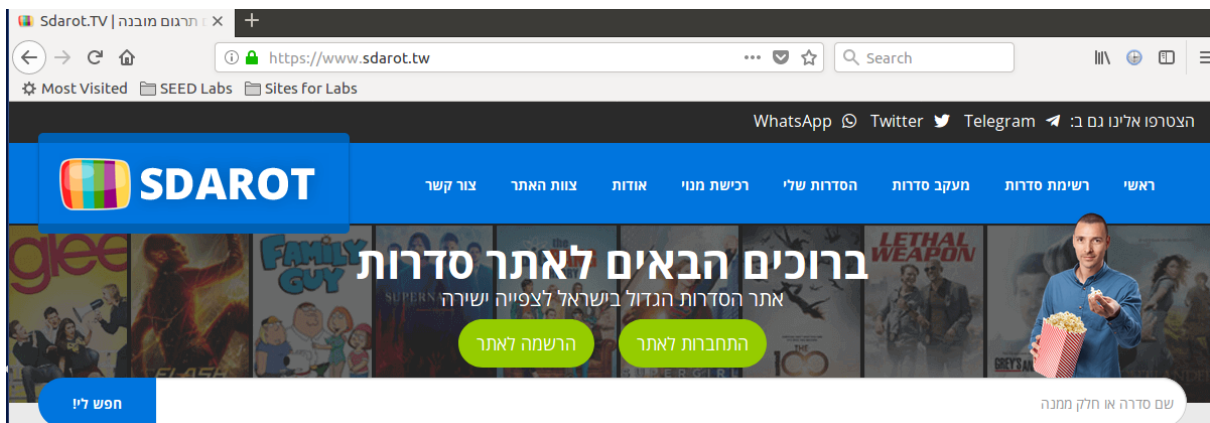
```
sudo iptables -A INPUT -p tcp --dport 23 -s 10.0.2.100 -d 10.0.2.4 -j DROP
```

Prevent A from visiting an external web site. You can choose any web site that you like to block, but keep in mind, some web servers have multiple IP addresses.

כעת ביצענו מחיקה לכל החוקים בטבלאות על ידי הרצת הפקודה `sudo iptables -F` ונרצה להגדיר חוק חדש אשר יחסום גישה לאתר `www.sdarot.tw` תחילה נבצע `host` ל `www.sdarot.tw` כדי לראות מה הכתובות של האתר

```
[Thu May 04 16:30:42] Computer A:~$ sudo host www.sdarot.tw
www.sdarot.tw has address 37.221.65.66
www.sdarot.tw has address 79.133.51.206
www.sdarot.tw has address 185.224.81.69
www.sdarot.tw has IPv6 address 2001:678:6d4:6010::6d4
www.sdarot.tw has IPv6 address 2a01:7e0:0:151:1fff:ffff:ffff:405a
```

קיבלנו 3 כתובות IPv4 שאותם נרצה לחסום כדי למנוע כניסה לאתר כעת ננסה להיכנס לקישור ונראה שהאתר תקין



ניתן לראות שהקישור מפנה לאתר בהצלחה

נוסיף את החוקים הבאים לטבלת ה `OUTPUT` לצורך חסימת גישה לאתר

```
[Thu May 04 16:36:50] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -s 10.0.2.4 -d 79.133.51.206 -j DROP
[Thu May 04 16:39:09] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -s 10.0.2.4 -d 185.224.81.69 -j DROP
[Thu May 04 16:39:26] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -s 10.0.2.4 -d 37.221.65.66 -j DROP
```

כתבנו חוק לכל כתובת IP

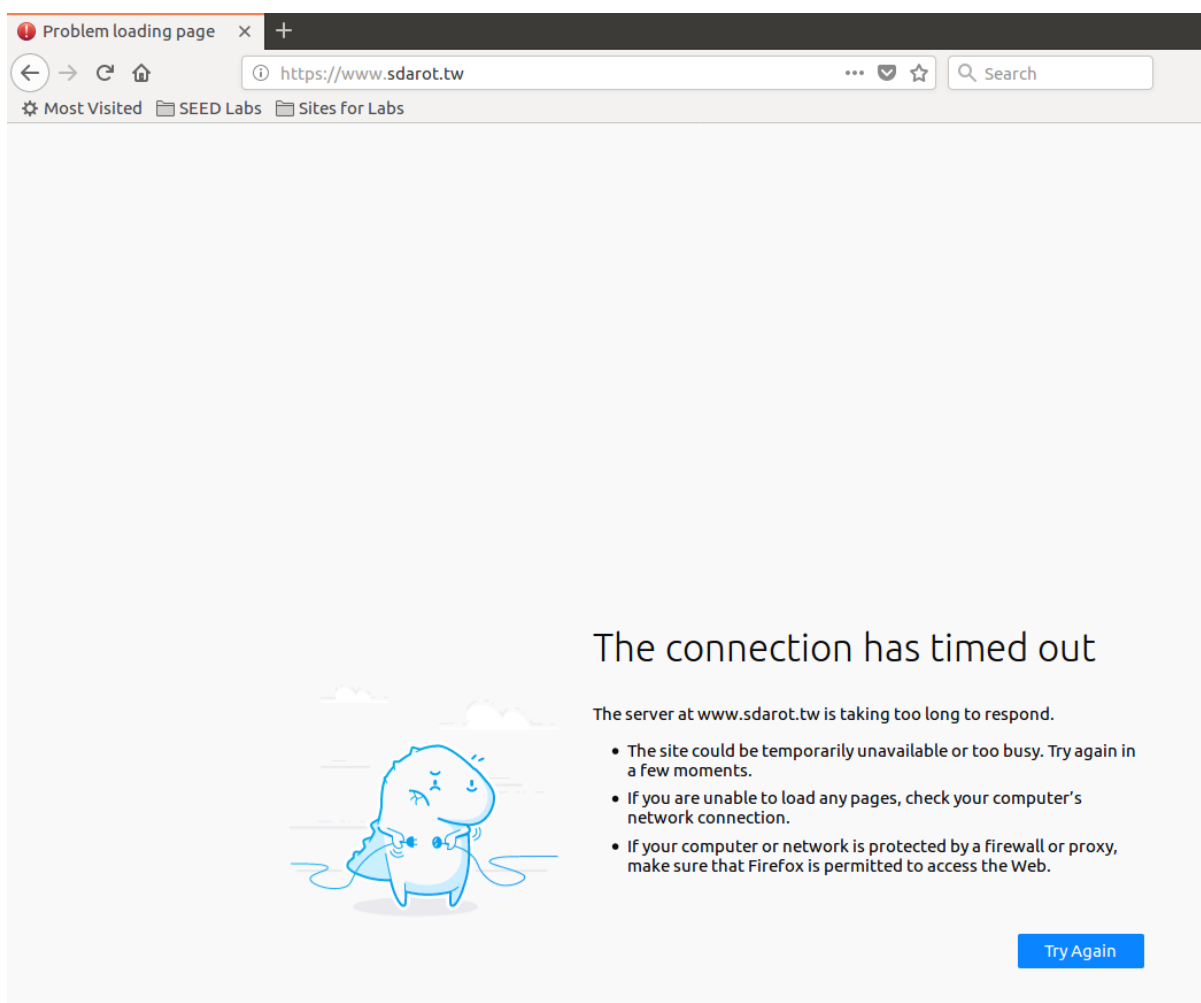
נרצה לוודא שהחוקים אכן נוספו לטבלת הOUTPUT

```
[Thu May 04 16:45:16] Computer A:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
DROP       tcp  --  10.0.2.4               localhost             tcp dpt:https
DROP       tcp  --  10.0.2.4               abelohost-69.81.224.185.dedicated-ip.ab
elons.com  tcp  dpt:https
DROP       tcp  --  10.0.2.4               79.133.51.206        tcp dpt:https
```

ניתן לראות שהחוקים נוספו בהצלחה לטבלה



לא ניתן לגשת לאתר המבוקש, רואים זאת לפי כך שלא קיבלנו תוצאה זזה למצב לפני כתיבת החוק בFIREWALL וקיבלנו connection has times out מאחר ולא הצלחנו ליצור קשר עם השרת של האתר.

ביצענו בנוסף פינג לאתר כדי לראות שהגישה בפרוטוקול ICMP לא נחסמה גם

```
[Thu May 04 16:39:43] Computer A:~$ ping www.sdarot.tw
PING www.sdarot.tw (37.221.65.66) 56(84) bytes of data.
64 bytes from localhost (37.221.65.66): icmp_seq=1 ttl=54 time=167 ms
64 bytes from localhost (37.221.65.66): icmp_seq=2 ttl=54 time=98.1 ms
64 bytes from localhost (37.221.65.66): icmp_seq=3 ttl=54 time=98.1 ms
^C
--- www.sdarot.tw ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 98.117/121.254/167.454/32.668 ms
```

וראינו שניתן לגשת לאתר בפרוטוקול שהוא לא TCP כפי שהגדרנו בחוקים.

סיכום המשימה

הצלחנו לבצע את המשימה, ניתן לראות שהצלחנו לחסום גישה בתקשורת TELNET ממחשב A למחשב B ולהפך והראנו זאת על ידי כך שניסיון התחברות נוסף לאחר הכנסת החוקים החדשים לFIREWALL נכשל.

בנוסף הצלחנו לחסום גישה לאתר ספציפי שבחרנו www.sdarot.tw והראינו זאת על ידי כך שלאחר כתיבת החוקים בFIREWALL קיבלנו תוצאה שהחיבור פג תוקף.

גילינו כיצד להגדיר חוקים לחסימת גישות עבור תקשורת ספציפית ועבור אתר ספציפי.

התוצאות התאימו למצופה מאחר ולפני כתיבת החוקים הייתה לנו אפשרות לגשת לחיבורים שרצינו ולאחר חסימתם והוספת החוקים לטבלה IPTABLES לא הצלחנו לבצע חיבור שוב פעם.

לא נתקלנו בבעיות במהלך ביצוע המשימה.

Task 2: Implementing a Simple Firewall

מבוא:

תיאור

במשימה זו נגדיר חוקים שונים לטבלאות ה-FIREWALL בעזרת קוד בשפת C

מטרה

לבטל אפשרות ליצירת חיבור TELNET בין שני מחשבים, חסימת אפשרות גישה לאתר ספציפי דרך מחשב מסוים, חסימת אפשרות לביצוע פינג לאתר ספציפי דרך מחשב מסוים, ביטול אפשרות לביצוע חיבורי SSH ממחשב מסוים.

תוצאה מצופה

עקב הפעלת החוקים במחשב יהיו כישלונות ביצירת חיבורים המוזכרים מעלה.

ביצוע המשימה

- הסבר קצר על NETFILTER
זוהי מסגרת בתוך הגרעין של מערכת לינוקס אשר מאפשר לבצע פעולות על תעבורת הרשת, כגון סינון, שינוי, הסנפה (יירוט).
משמש בעיקר ליישום חומות אש, כתובות NAT, והתעסקות בPACKETS.
 - הסבר קצר על NAT
זוהי טכניקה אשר משמשת לתרגום כתובות IP ברשת הפרטית לכתובת IP חיצונית אחת. טכניקה זו מספקת חיסכון בכתובות IP ושכבת אבטחה בכך שמסתירה את הכתובת הפרטית ממנה יוצא הPACKET.
 - הסבר קצר על LKM
פיסות קוד שניתן לטעון ולפרוק מליבת לינוקס מבלי לדרוש אתחול מחדש או לבצע COMPILE מחדש לליבה.
מאפשר שינוי דינמי של הליבה בזמן ריצה, ומאפשר למשתמשים להוסיף או להסיר תכונות ספציפיות לפי הצורך כגון פונקציונליות נוספת, מנהלי התקנים או ממשקי קריאת מערכת.
פיתוח ב-LKM נעשה בדרך כלל בקוד בשפות C או ++C שמתממשק עם ממשק תכנות היישומים (API) של ליבת לינוקס.
את הקוד מרכיבים לקובץ אובייקט, אשר מקושר לאחר מכן לקרנל כדי ליצור מודול הניתן לטעינה, כדי לטעון את המודול לתוך הקרנל נשתמש בפקודה insmod, וכדי לפרוק נשתמש בפקודה rmmod.
- לפני הכנסת החוקים החדשים נעבור על כל החיבורים אותם נרצה לחסום ונראה כי הם תקינים וללא הוספת החוקים ניתן ליצור את החיבורים האלו.

ביצוע חיבור TELNET ממחשב A למחשב B

```
[Fri May 05 17:01:01] Computer A:~$ telnet 10.0.2.100
Trying 10.0.2.100...
Connected to 10.0.2.100.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: Connection closed by foreign host.
[Fri May 05 17:02:49] Computer A:~$
```

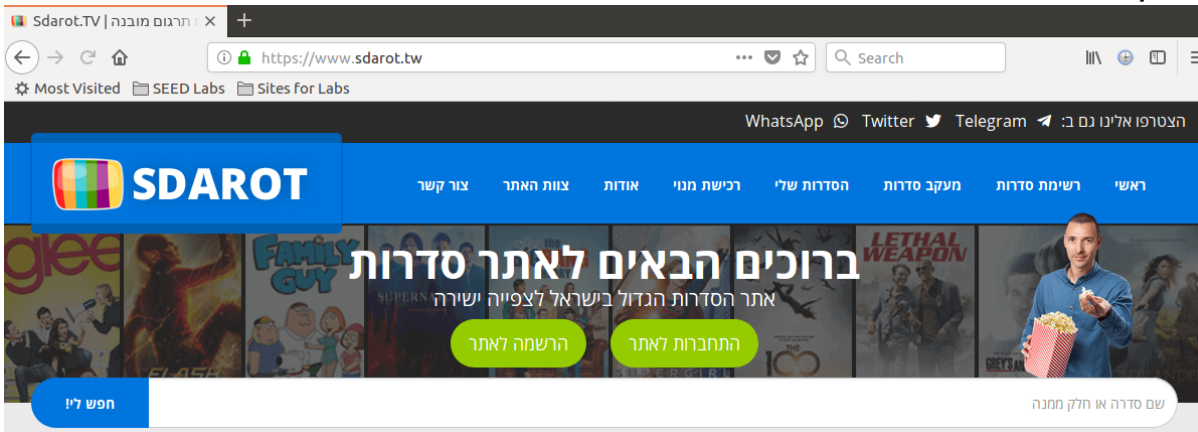
חיבור הTELNET הצליח

ביצוע חיבור TELNET ממחשב B למחשב A

```
[Fri May 05 17:02:28] Computer B:~$ telnet 10.0.2.4
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: Connection closed by foreign host.
```

חיבור הTELNET הצליח

ניסיון התחברות לאתר סדרות



ניסיון החביר הצליח

ניסיון יצירת חיבור SSH ממחשב A למחשב B

```
[Fri May 05 18:00:28] Computer A:~$ ssh 10.0.2.100
seed@10.0.2.100's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Fri May  5 17:57:32 2023 from 10.0.2.4
[Fri May 05 18:00:37] Computer B:~$
```

חיבור הSSH בין המחשבים הצליח

ניסיון שליחת פינג לאתר סדרות

```
[Fri May 05 17:24:04] Computer A:~$ ping www.sdarot.tw
PING www.sdarot.tw (37.221.65.66) 56(84) bytes of data.
64 bytes from localhost (37.221.65.66): icmp_seq=1 ttl=54 time=173 ms
64 bytes from localhost (37.221.65.66): icmp_seq=2 ttl=54 time=98.5 ms
64 bytes from localhost (37.221.65.66): icmp_seq=3 ttl=54 time=97.5 ms
^C
--- www.sdarot.tw ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3004ms
```

הפינג התקבל בהצלחה

כעת נרשום את הקוד הבא:

```
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/icmp.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/string.h>
#define MAX_RULE_NUM 5

static struct nf_hook_ops FilterHookRule[MAX_RULE_NUM];
static int regist_num = 0;

|
//-----
//      Function to compare between given dst ip and packet dst ip
//-----
int eq_daddr(const struct iphdr *iph, const char *ip_addr)
{
    char source[16];
    snprintf(source, 16, "%pI4", &iph->daddr);
    if (strcmp(source, ip_addr) == 0)
        return 1;
    return 0;
}

//-----
//      Function to compare between given src ip and packet src ip
//-----
int eq_saddr(const struct iphdr *iph, const char *ip_addr)
{
    char source[16];
    snprintf(source, 16, "%pI4", &iph->saddr);
    if (strcmp(source, ip_addr) == 0)
        return 1;
    return 0;
}
```

פונקציות אשר בודקות את האייפי ב-PACKET לבין IP שנשלח לפונקציה,
פונקציה אחת ל-SRC ואחת ל-DST.

פקודת ה-SNPRINTF לוקחת את האיפוי מההדר של ה-PACKET והופכת אותו מביטים ב-HEX לצורה שניתן לקרוא כגון 10.0.2.4

```
//-----  
//      RULE 1  
//-----  
unsigned int block_telnet_A_B(void *priv, struct sk_buff *skb,  
                             const struct nf_hook_state *state)  
// rule for Prevent A from doing telnet to B  
{  
    struct iphdr *iph;  
    struct tcphdr *tcph;  
  
    iph = ip_hdr(skb);  
    tcph = (void *)iph + iph->ihl * 4;  
  
    if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) && eq_saddr(iph, "10.0.2.4") && eq_daddr(iph, "10.0.2.100"))  
    {  
        printk(KERN_INFO "Dropping telnet from %pI4 packet to %pI4\n", &iph->saddr, &iph->daddr);  
        return NF_DROP;  
    }  
    else  
    {  
        return NF_ACCEPT;  
    }  
}
```

חוק 1 – מונע גישה בתקשורת TELNET ממחשב A למחשב B
Htons – ממיר את הפורט ל-HEX כך שנוכל להשוות עם הפורט שרשום בתוך ההדר של TCP
Tcph – לוקחים את המצביע להדר של IP שהוא ההתחלה של הבאפר של ה-PACKET ומוסיפים את אורך ההדר של IP כפול 4 ביטים ומגיעים לחלק שאחרי ההדר של IP והוא ההדר של TCP.

```
//-----  
//      RULE 2  
//-----  
unsigned int block_telnet_B_A(void *priv, struct sk_buff *skb,  
                             const struct nf_hook_state *state)  
// rule for Prevent B from doing telnet to A  
{  
    struct iphdr *iph;  
    struct tcphdr *tcph;  
  
    iph = ip_hdr(skb);  
    tcph = (void *)iph + iph->ihl * 4;  
  
    if (iph->protocol == IPPROTO_TCP && tcph->dest == htons(23) && eq_saddr(iph, "10.0.2.100") && eq_daddr(iph, "10.0.2.4"))  
    {  
        printk(KERN_INFO "Dropping telnet from %pI4 packet to %pI4\n", &iph->saddr, &iph->daddr);  
        return NF_DROP;  
    }  
    else  
    {  
        return NF_ACCEPT;  
    }  
}
```

חוק 2 – מונע גישה בתקשורת TELNET ממחשב B למחשב A

```
//-----
//      RULE 3
//-----
unsigned int block_sdarot(void *priv, struct sk_buff *skb,
                          const struct nf_hook_state *state)
// rule for Prevent A from connect to www.sdarot.tw
// the hosts obtained from the terminal using the command: sudo host www.sdarot.tw
{
    struct iphdr *iph;
    struct tcphdr *tcph;
    iph = ip_hdr(skb);
    tcph = (void *)iph + iph->ihl * 4;
    if ((tcph->dest == htons(80) || tcph->dest == htons(443))
        && (eq_daddr(iph, "37.221.65.66") || eq_daddr(iph, "185.224.81.69") || eq_daddr(iph, "79.133.51.206"))
        && eq_saddr(iph, "10.0.2.4"))
    {
        printk(KERN_INFO "Dropping http/https from %pI4 packet to %pI4\n", &iph->saddr, &iph->daddr);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}
```

חוק 3 – מונע ממחשב A לגשת לאתר www.sdarot.tw על ידי חסימת גישה בפורט 80 HTTP ובפורט 443 HTTPS לכל כתובות הHOST של האתר

```
//-----
//      RULE 4
//-----
unsigned int block_ssh(void *priv, struct sk_buff *skb,
                      const struct nf_hook_state *state)
// rule for Prevent A from doing ssh
{
    struct iphdr *iph;
    struct tcphdr *tcph;
    iph = ip_hdr(skb);
    tcph = (void *)iph + iph->ihl * 4;

    if (tcph->dest == htons(22) || tcph->source == htons(22))
    {
        printk(KERN_INFO "Dropping SSH packet from %pI4 packet to %pI4\n", &iph->saddr, &iph->daddr);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}
```

חוק 4 – מונע ממחשב A לקבל או ליצור חיבורי SSH מכל מחשב שהוא בפורט 22 שזה הפורט המיועד לSSH

```
//-----
//      RULE 5
//-----
unsigned int block_icmp_to_sdarot(void *priv, struct sk_buff *skb,
                                const struct nf_hook_state *state)
// rule for Prevent A from ping to www.sdarot.tw
{
    struct iphdr *iph;
    struct icmphdr *icmph;
    iph = ip_hdr(skb);
    icmph = (void *)iph + iph->ihl * 4;
    if (iph->protocol == IPPROTO_ICMP && icmph->type == ICMP_ECHO
        && (eq_daddr(iph, "37.221.65.66") || eq_daddr(iph, "185.224.81.69") || eq_daddr(iph, "79.133.51.206"))
        && eq_saddr(iph, "10.0.2.4"))
    {
        printk(KERN_INFO "Dropping ICMP Echo Request to www.sdarot.tw from %pI4\n", &iph->saddr);
        return NF_DROP;
    }
    else
    {
        return NF_ACCEPT;
    }
}
}
```

חוק 5 – מונע ממחשב A לבצע PING לשרתים של אתר www.sdarot.tw
(מונע שליחת ICMP ECHO REQUEST)

```
int setUpFilter(void)
{
    int i;
    printk(KERN_INFO "Registering filters.\n");
    FilterHookRule[0] = (struct nf_hook_ops){.hook = block_telnet_A_B, .hooknum = NF_INET_LOCAL_OUT, .pf =
PF_INET, .priority = NF_IP_PRI_FIRST};
    FilterHookRule[1] = (struct nf_hook_ops){.hook = block_telnet_B_A, .hooknum = NF_INET_LOCAL_IN, .pf =
PF_INET, .priority = NF_IP_PRI_FIRST};
    FilterHookRule[2] = (struct nf_hook_ops){.hook = block_sdarot, .hooknum = NF_INET_LOCAL_OUT, .pf =
PF_INET, .priority = NF_IP_PRI_FIRST};
    FilterHookRule[3] = (struct nf_hook_ops){.hook = block_ssh, .hooknum = NF_INET_LOCAL_IN, .pf =
PF_INET, .priority = NF_IP_PRI_FIRST};
    FilterHookRule[4] = (struct nf_hook_ops){.hook = block_icmp_to_sdarot, .hooknum =
NF_INET_LOCAL_OUT, .pf = PF_INET, .priority = NF_IP_PRI_FIRST};

    // set the amount of filter rules
    regist_num = 5;

    for (i = 0; i < regist_num; i++)
        nf_register_hook(&FilterHookRule[i]);
    return 0;
}

void removeFilter(void)
{
    int i;
    printk(KERN_INFO "Filters are being removed.\n");
    //unregist hooks one by one
    for (i = 0; i < regist_num; i++)
        nf_unregister_hook(&FilterHookRule[i]);
    regist_num = 0;
}

module_init(setUpFilter);
module_exit(removeFilter);

MODULE_LICENSE("GPL");
```

פונקציות setUpFilter ו- removeFilter מתארות מה יתבצע כאשר נרצה
לטעון את המודול לקרנל ולהסירו.

נפתח קובץ חדש בשם Makefile אשר יכיל את הקוד הבא:

```
obj-m += ex2.o
all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

הקובץ הנ"ל אומר למערכת לבנות את המודל שכתבנו ולהוסיף אותו לתיקיית המודולים בקרנל.

נבצע הרצה לקוד על ידי כתיבת הפקודה make אשר מבצעת את הפקודות שרשמנו בקובץ Makefile

```
[Fri May 05 17:02:49] Computer A:~$ make
make -C /lib/modules/4.8.0-36-generic/build M=/home/seed/Codes/EX5/Rules modules
make[1]: Entering directory '/usr/src/linux-headers-4.8.0-36-generic'
CC [M] /home/seed/Codes/EX5/Rules/ex2.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/seed/Codes/EX5/Rules/ex2.mod.o
LD [M] /home/seed/Codes/EX5/Rules/ex2.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.8.0-36-generic'
```

הקבצים נבנו בהצלחה

כעת נטען את המודול לקרנל

```
[Fri May 05 17:04:37] Computer A:~$ sudo insmod ex2.ko
[Fri May 05 17:05:04] Computer A:~$ lsmod
Module              Size  Used by
ex2                  16384  0
```

ניתן לראות שהמודול נטען לקרנל בהצלחה

כעת ננסה לבצע את החיבורים בהתחלה שוב פעם ונרצה לראות שהם נכשלים

נבצע חיבור TELNET ממחשב A למחשב B

```
[Fri May 05 17:05:50] Computer A:~$ telnet 10.0.2.100
Trying 10.0.2.100...
```

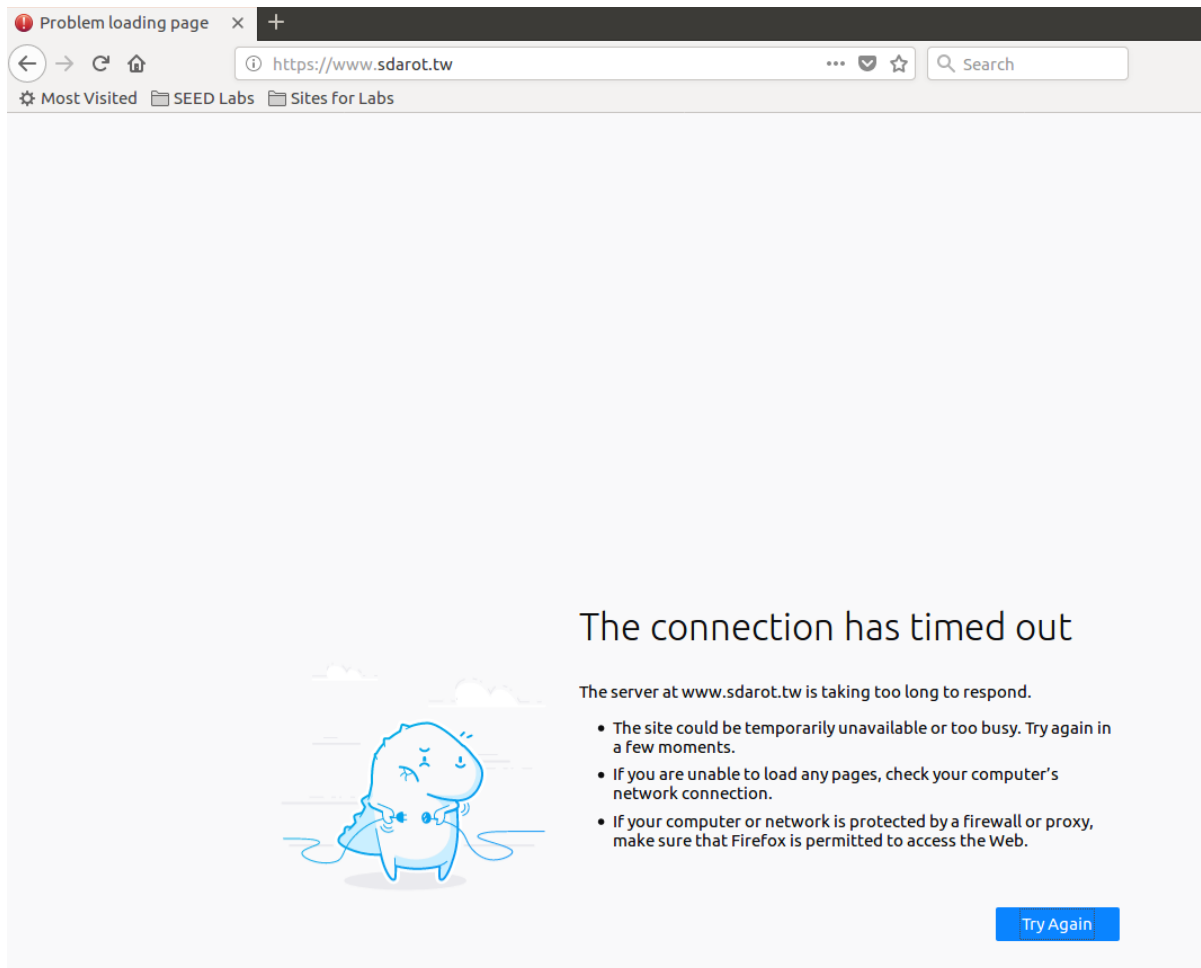
ניתן לראות שהחיבור נכשל בהצלחה מאחר ולא נוצר חיבור בין המחשבים

נבצע חיבור TELNET ממחשב B למחשב A

```
[Fri May 05 17:05:52] Computer B:~$ telnet 10.0.2.4  
Trying 10.0.2.4...
```

ניתן לראות שהחיבור נכשל בהצלחה מאחר ולא נוצר חיבור בין המחשבים

ננסה להתחבר ממחשב A לאתר www.sdarot.tw



ניתן לראות שהחיבור פג תוקף ולכן החיבור נכשל בהצלחה

ננסה ליצור חיבור SSH ממחשב A למחשב B

```
[Fri May 05 17:59:36] Computer A:~$ ssh 10.0.2.100
```

ניתן לראות שהחיבור נכשל בהצלחה מאחר ולא נוצר חיבור בין המחשבים

ננסה לשלוח ICMP ECHO REQUEST (PING) לאתר www.sdarot.tw

```
[Fri May 05 17:06:51] Computer A:~$ ping www.sdarot.tw
PING www.sdarot.tw (37.221.65.66) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- www.sdarot.tw ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3073ms
```

ניתן לראות שביצוע PING נכשל בהצלחה מאחר והבקשות הICMP אבדו ולא התקבלו אצל היעד.

ניתן לראות את הלוגים שהודפסו על ידי הקוד שלנו בקובץ הLOG של
הKERNEL בנתיב /var/log/syslog/

```
May 5 17:05:04 VM kernel: [19258.853535] Registering filters.
May 5 17:05:58 VM kernel: [19312.121967] Dropping telnet from 10.0.2.4 packet to 10.0.2.100
May 5 17:05:59 VM kernel: [19313.152874] Dropping telnet from 10.0.2.4 packet to 10.0.2.100
May 5 17:06:00 VM kernel: [19314.423585] Dropping telnet from 10.0.2.100 packet to 10.0.2.4
May 5 17:06:01 VM kernel: [19315.168679] Dropping telnet from 10.0.2.4 packet to 10.0.2.100
May 5 17:06:01 VM kernel: [19315.445469] Dropping telnet from 10.0.2.100 packet to 10.0.2.4
May 5 17:06:03 VM kernel: [19317.461273] Dropping telnet from 10.0.2.100 packet to 10.0.2.4
May 5 17:06:05 VM kernel: [19319.264231] Dropping telnet from 10.0.2.4 packet to 10.0.2.100
May 5 17:06:07 VM kernel: [19321.682848] Dropping telnet from 10.0.2.100 packet to 10.0.2.4
May 5 17:06:13 VM kernel: [19327.456609] Dropping telnet from 10.0.2.4 packet to 10.0.2.100
May 5 17:06:16 VM kernel: [19329.870100] Dropping telnet from 10.0.2.100 packet to 10.0.2.4
May 5 17:06:29 VM kernel: [19343.584721] Dropping telnet from 10.0.2.4 packet to 10.0.2.100
May 5 17:06:32 VM kernel: [19345.991480] Dropping telnet from 10.0.2.100 packet to 10.0.2.4
May 5 17:06:57 VM kernel: [19371.217026] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:06:57 VM kernel: [19371.217063] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:06:57 VM kernel: [19371.472152] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:06:58 VM kernel: [19372.224098] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:06:58 VM kernel: [19372.224101] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:06:58 VM kernel: [19372.480373] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:00 VM kernel: [19374.240254] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:00 VM kernel: [19374.240258] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:00 VM kernel: [19374.496409] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:04 VM kernel: [19378.400077] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:04 VM kernel: [19378.400080] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:04 VM kernel: [19378.660038] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:12 VM kernel: [19386.596083] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:12 VM kernel: [19386.596087] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:12 VM kernel: [19386.848176] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:18 VM kernel: [19392.252438] device enp0s3 entered promiscuous mode
May 5 17:07:22 VM kernel: [19396.510479] Dropping ICMP Echo Request to www.sdarot.tw from 10.0.2.4
May 5 17:07:23 VM kernel: [19397.536116] Dropping ICMP Echo Request to www.sdarot.tw from 10.0.2.4
May 5 17:07:24 VM kernel: [19398.560102] Dropping ICMP Echo Request to www.sdarot.tw from 10.0.2.4
May 5 17:07:25 VM kernel: [19399.584333] Dropping ICMP Echo Request to www.sdarot.tw from 10.0.2.4
May 5 17:07:28 VM kernel: [19402.720441] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:28 VM kernel: [19402.720444] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:07:29 VM kernel: [19402.976203] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:02 VM kernel: [19436.512163] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:02 VM kernel: [19436.512235] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:02 VM kernel: [19436.512240] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:27 VM kernel: [19461.493167] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:28 VM kernel: [19461.918951] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:28 VM kernel: [19462.169853] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:29 VM kernel: [19462.944455] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:29 VM kernel: [19463.200465] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:31 VM kernel: [19464.960230] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:31 VM kernel: [19465.217067] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:35 VM kernel: [19469.024760] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:35 VM kernel: [19469.280567] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:43 VM kernel: [19477.216814] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:43 VM kernel: [19477.476089] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:59 VM kernel: [19493.344352] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:08:59 VM kernel: [19493.600698] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:09:02 VM CRON[8490]: (root) CMD ( [ -x /usr/lib/php/sessionclean ] && /usr/lib/php/sessionclean)
May 5 17:09:32 VM kernel: [19526.624146] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
May 5 17:09:32 VM kernel: [19526.624245] Dropping http/https from 10.0.2.4 packet to 37.221.65.66
```

ניתן לראות שהודפסו הודעות על כל הPACKETS שנזרקו עקב החוקים שהגדרנו
בFIREWALL.

סיכום המשימה

הצלחנו לבצע את המשימה, ניתן לראות שלפני הגדרת החוקים ב-KERNEL יכולנו ליצור את כל החיבורים שהם:

TELNET בין A ל B

TELNET בין B ל A

חיבור לאתר www.sdarot.tw

יצירת חיבורי SSH ממחשב A למחשב B

שליחת PING לאתר www.sdarot.tw

ולאחר טעינת המודול שבנינו ל-KERNEL ניסיון נוסף ליצירת החיבורים האלו נכשל בהצלחה.

גילינו כיצד להגדיר חוקים לחסימת גישות בעזרת BACKEND ב-FIREWALL וכיצד לטעון ולהסיר מודולים מה-KERNEL.

התוצאות התאימו למצופה מאחר ולפני כתיבת החוקים הייתה לנו אפשרות לגשת לחיבורים שהוזכרו מעלה, ולאחר טעינת החוקים ב-KERNEL כל החיבורים שהוזכרו נחסמו לאלתר.

כדי להצליח במשימה נעזרנו רבות בדוקומנטציה של NETFILTER בקישור הבא: <https://netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO-4.html>

Task 3: Evading Egress Filtering

מבוא:

תיאור

במשימה זו נרצה לעקוף את החוקים המוגדרים בחומת האש בעזרת
SSH TUNNEL

מטרה

נגדיר חוקים בחומת האש אשר מונעים חיבור בTELNET מהמחשב
החוצה, ומונעים חיבור לאתר ספציפי מהמחשב, ונרצה לעקוף את
החסימה על ידי יצירת SSH TUNNEL והעברת התעבורה דרכה.

תוצאה מצופה

נצליח לבצע חיבור TELNET מהמחשב החסום ונצליח להתחבר לאתר
החסום.

ביצוע המשימה

Block all the outgoing traffic to external telnet servers

נרצה לבצע חסימה לכל התעבורה שיוצאת ממחשב A בתקשורת TELNET

לכן נכתוב את הפקודה הבאה בטרמינל:

```
[Sun May 07 18:54:37] Computer A:~$ sudo iptables -A OUTPUT -p
tcp --dport 23 -j DROP
[Sun May 07 18:54:40] Computer A:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
DROP      tcp  --  anywhere              anywhere
cp dpt:telnet
```

ניתן לראות שהחוק נוסף בהצלחה אל טבלת החוקים של OUTPUT

ננסה לבצע חיבור TELNET למחשב חיצוני

```
[Sun May 07 18:54:42] Computer A:~$ telnet 10.0.2.100
Trying 10.0.2.100...
```

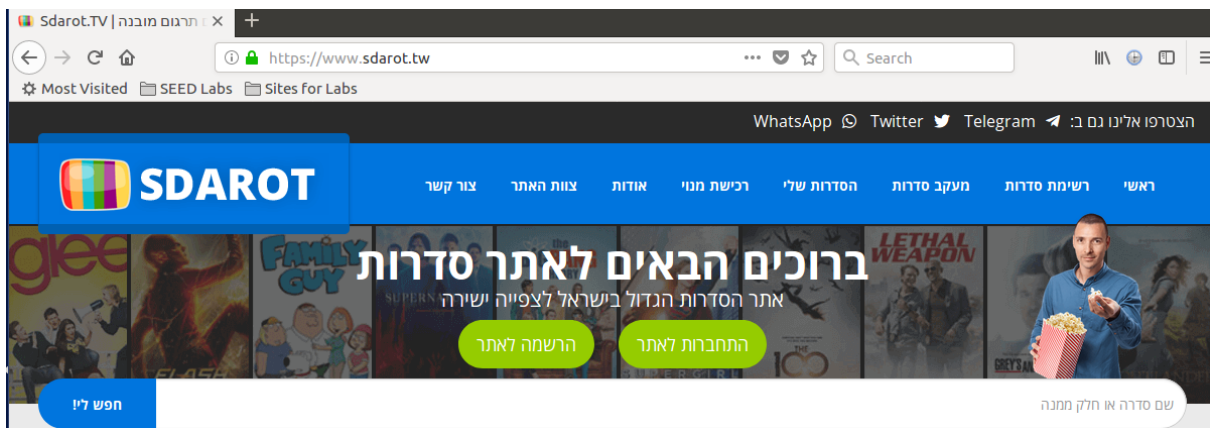
ניתן לראות שהחיבור נכשל בהצלחה מאחר ולא נוצר חיבור בין שני המחשבים

Block all the outgoing traffic to www.sdarot.tw

נרצה להגדיר חוק חדש אשר יחסום גישה לאתר www.sdarot.tw כדי לראות מה הכתובות של האתר

```
[Thu May 04 16:30:42] Computer A:~$ sudo host www.sdarot.tw
www.sdarot.tw has address 37.221.65.66
www.sdarot.tw has address 79.133.51.206
www.sdarot.tw has address 185.224.81.69
www.sdarot.tw has IPv6 address 2001:678:6d4:6010::6d4
www.sdarot.tw has IPv6 address 2a01:7e0:0:151:1fff:ffff:ffff:405a
```

קיבלנו 3 כתובות IPv4 שאותם נרצה לחסום כדי למנוע כניסה לאתר
כעת ננסה להיכנס לקישור ונראה שהאתר תקין



ניתן לראות שהקישור מפנה לאתר בהצלחה

נוסיף את החוקים הבאים לטבלת הOUTPUT לצורך חסימת גישה לאתר

```
[Sun May 07 19:06:39] Computer A:~$ sudo iptables -A OUTPUT -p
tcp --dport 443 -d 79.133.51.206 -j DROP
[Sun May 07 19:06:46] Computer A:~$ sudo iptables -A OUTPUT -p
tcp --dport 443 -d 185.224.81.69 -j DROP
[Sun May 07 19:06:50] Computer A:~$ sudo iptables -A OUTPUT -p
tcp --dport 443 -d 37.221.65.66 -j DROP
```

כתבנו חוק לכל כתובת IP

נרצה לוודא שהחוקים אכן נוספו לטבלת הOUTPUT

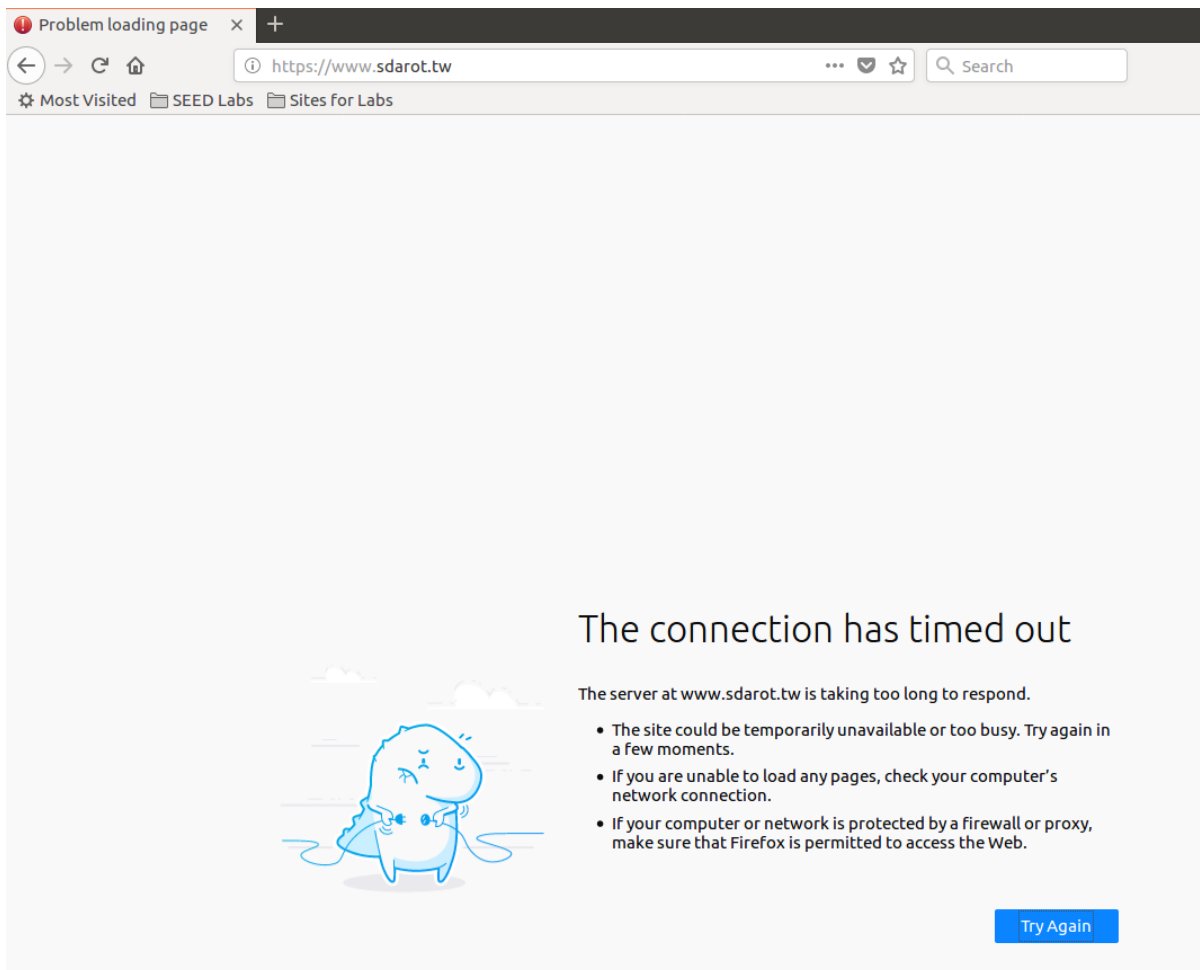
```
[Sun May 07 19:07:03] Computer A:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DROP      tcp  --  anywhere             anywhere             t
cp dpt:telnet
DROP      tcp  --  anywhere             79.133.51.206        t
cp dpt:https
DROP      tcp  --  anywhere             abelohost-69.81.224.18
5.dedicated-ip.abelons.com tcp dpt:https
DROP      tcp  --  anywhere             localhost            t
cp dpt:https
```

ניתן לראות שהחוקים נוספו בהצלחה לטבלה

ננסה להתחבר ממחשב A לאתר www.sdarot.tw



ניתן לראות שהחיבור פג תוקף ולכן החיבור נכשל בהצלחה

Task 3.a: Telnet to Machine B through the firewall

תחילה נבצע חיבור SSH בין מחשב A למחשב B

```
[Fri May 12 12:33:30] Computer A:~$ ssh 10.0.2.100
seed@10.0.2.100's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-gen
eric i686)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.
```

ניתן לראות שהחיבור נוצר בהצלחה

כעת לאחר יצירת הTUNNEL בין מחשב A למחשב B בעזרת SSH נרצה ליצור חיבור TELNET בין מחשב B לעצמו ובכך באופן עקיף יצרנו חיבור TELNET ממחשב A למחשב B ועקפנו את החוקים של חומת האש.

ביצענו חיבור TELNET למחשב B כאשר אנחנו כבר בתוך מחשב B

```
Last login: Fri May 12 12:33:22 2023 from 10.0.2.100
[Fri May 12 12:33:46] Computer B:~$ telnet 10.0.2.100
Trying 10.0.2.100...
Connected to 10.0.2.100.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Fri May 12 12:33:45 IDT 2023 from 10.0.2.
4 on pts/1
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-gen
eric i686)
```

ניתן לראות שהחיבור הצליח ונבדוק שאכן הIP שמתקבל במכונה הוא הIP של מחשב B

```
[Fri May 12 12:33:54] Computer B:~$ ifconfig  
enp0s3      Link encap:Ethernet  HWaddr 08:00:27:73:fc:  
14  
              inet addr:10.0.2.100  Bcast:10.0.2.255  Mas  
k:255.255.255.0
```

קיבלנו את הIP של מחשב B 10.0.2.100

כעת נרצה לנתק את החיבורים על ידי EXIT

```
[Fri May 12 12:33:58] Computer B:~$ exit  
logout  
Connection closed by foreign host.  
[Fri May 12 12:34:36] Computer B:~$ exit  
logout  
Connection to 10.0.2.100 closed.
```

ניתן לראות שהיינו צריכים לבצע פעמיים EXIT כדי לחזור למחשב A, פעם ראשונה כדי לנתק את חיבור הTELNET ופעם שניה כדי לנתק את חיבור הSSH

כאשר בדקנו בWIRESHARK לא ראינו PACKETS שעוברים בTELNET מאחר וביצענו בין המחשב לעצמו

לכן ניסינו לבצע חיבור בין מחשב A לב B בSSH ולאחר מכן חיבור בין מחשב B למחשב A בTELNET ובכך יצרנו חיבור בין מחשב A לעצמו בTELNET שדבר זה לא אמור להיות אפשרי עקב החוקים שהגדרנו בחומת האש, וניתן לראות שהצלחנו והWIRESHARK הסניף את הPACKETS המראות חיבור בין המחשבים.

10.0.2.4	10.0.2.100	SSHv2
10.0.2.100	10.0.2.4	TELNET

חיבור בין מחשב A לב B בSSH

חיבור בין מחשב B לא A בTELNET

Task 3.b: Connect to Sdarot using SSH Tunnel

נרצה לעקוף את חוקי חומת האש ולנסות לגשת לאתר www.sdarot.tw דרך TUNNEL שניצור בחיבור SSH בין מחשב A למחשב B

ניצור חיבור SSH בין מחשב A למחשב B על ידי פורט 9000

```
[Fri May 12 13:05:47] Computer A:~$ ssh -D 9000 -C 10.0.2.100
seed@10.0.2.100's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Fri May 12 13:05:29 2023 from 10.0.2.4
```

-D 9000 – יוצר פורט האזנה על מחשב A לצורך העברת הנתונים דרך פורט זה

-C – מכווץ את נתוני התעבורה שעוברים בפורט כדי להקטין את השימוש ברוחב פס

ניתן לראות שיצרנו חיבור SSH עם socks proxy בפורט 9000 וכעת נרצה להגדיר שאפליקציית firefox תעביר את תעבורת הרשת שלה דרך ה-SOCKET שפתחתנו וכך תבצע העברה למחשב B דרך ה-ssh tunnel שפתחתנו.

SOCKS PROXY יוצר פורט האזנה לאפליקציות על המחשב, דרך הפורט הזה נוכל ליצור חיבור למחשב אחר ובכך להעביר את תעבורת הרשת שנשלחת לפורט הנ"ל.

נכנס בFIREFOX ללשונית

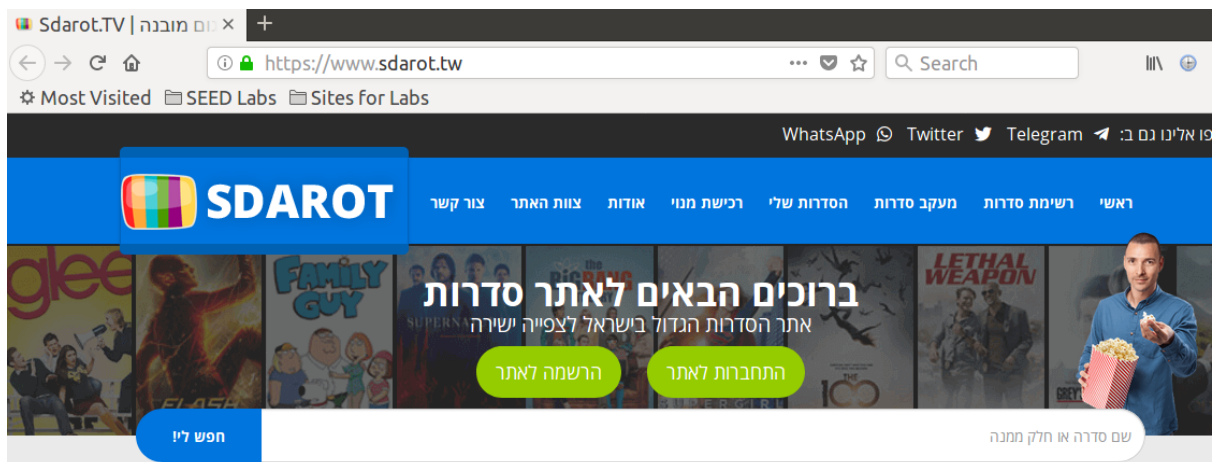
EDIT->PREFERENCES->NETWORK PROXY->SETTING

ונשנה את ההגדרות לפי התמונה מטה

The screenshot shows the 'Connection Settings' dialog box in Firefox. The title bar says 'Connection Settings' with a close button. The main heading is 'Configure Proxy Access to the Internet'. There are four radio buttons: 'No proxy', 'Auto-detect proxy settings for this network', 'Use system proxy settings', and 'Manual proxy configuration'. The 'Manual proxy configuration' option is selected. Below it, there are fields for 'HTTP Proxy' and 'Port' (set to 0). A checkbox 'Use this proxy server for all protocols' is unchecked. Below that are fields for 'SSL Proxy' and 'Port' (set to 0), and 'FTP Proxy' and 'Port' (set to 0). The 'SOCKS Host' is set to '127.0.0.1' and the 'Port' is set to '9000'. There are two radio buttons for 'SOCKS v4' and 'SOCKS v5', with 'SOCKS v5' selected. Below these is a section 'No Proxy for' with a text box containing 'localhost, 127.0.0.1'. An example is given: 'Example: .mozilla.org, .net.nz, 192.168.1.0/24'. There is a radio button for 'Automatic proxy configuration URL' which is unchecked, followed by a text box and a 'Reload' button. At the bottom, there are two checkboxes: 'Do not prompt for authentication if password is saved' (unchecked) and 'Proxy DNS when using SOCKS v5' (unchecked). At the very bottom are buttons for 'Help', 'Cancel', and 'OK'.

כעת הגדרנו שכל תעבורת האפליקציה של FIREFOX תעבור דרך הSOCKET שפתחתנו בפורט 9000 על המחשב הלוקאלי מחשב A.

כעת נתחבר ממחשב A לאתר www.sdarot.tw



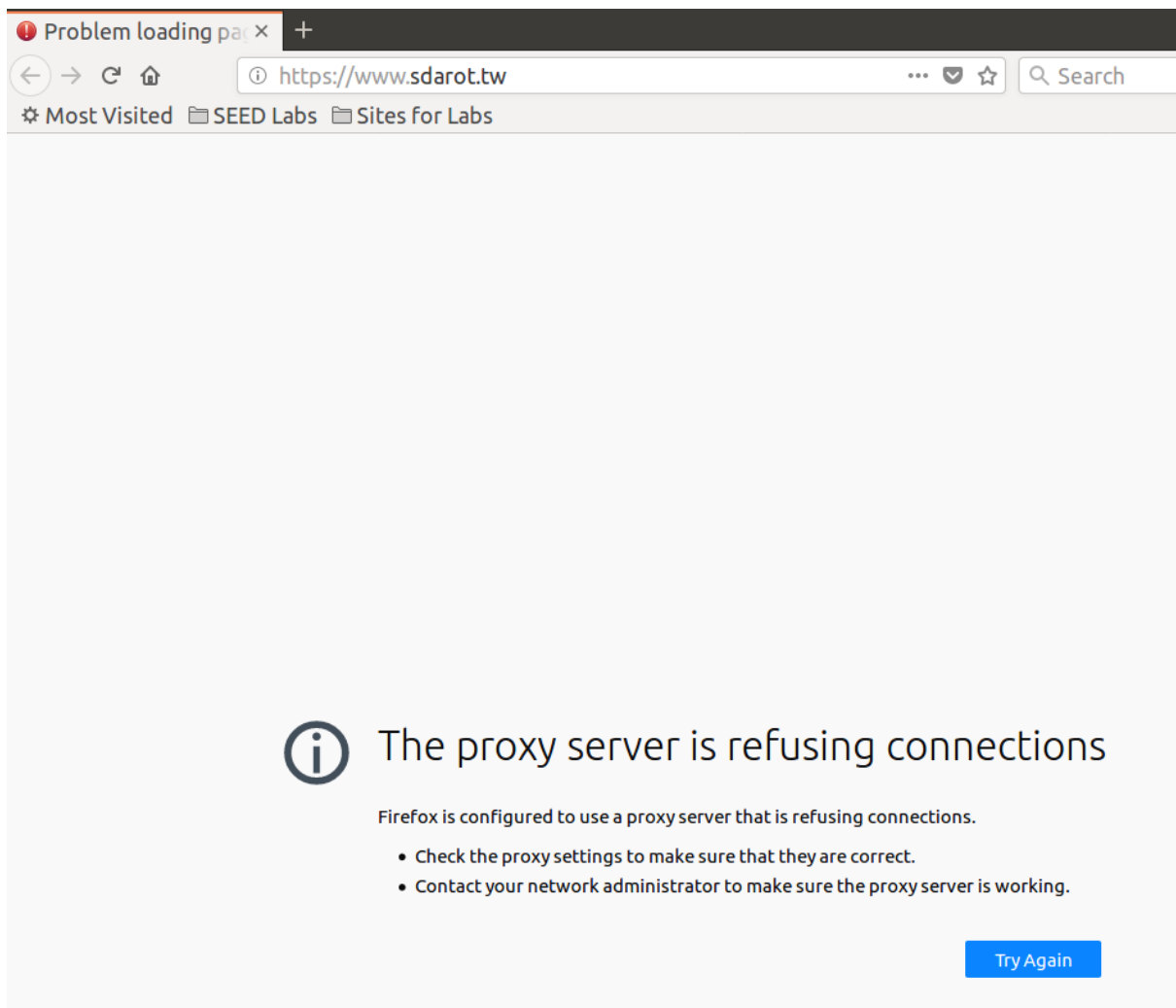
ניתן לראות שהחיבור לאתר הצליח לאחר העברת התעבורה דרך פורט 9000
המחובר בSSH למחשב B

כעת ננתק את חיבור הSSH בין מחשב A למחשב B

```
[Fri May 12 13:08:49] Computer B:~$ exit  
logout  
Connection to 10.0.2.100 closed.
```

ניתן לראות שהחיבור נותק בהצלחה

ננקה את זיכרון הCACHE של הFIREFOX ונסה להתחבר לאתר סדרות שוב פעם

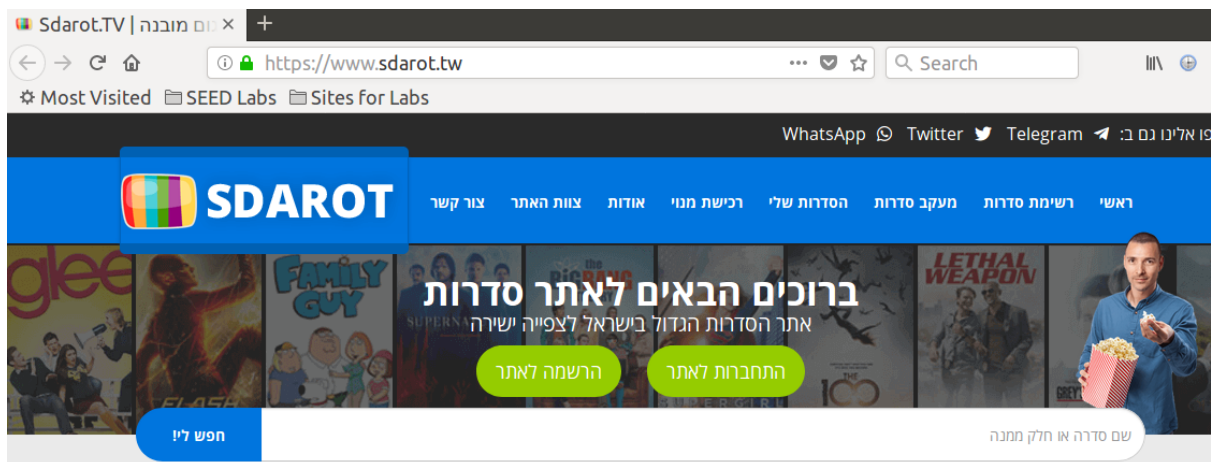


ניתן לראות שהחיבור כעת נכשל מאחר ואין חיבור לשרת הPROXY שהגדרנו
בהגדרות הFIREFOX

כעת ניצור את החיבור שוב פעם

```
[Fri May 12 13:09:56] Computer A:~$ ssh -D 9000 -C 10  
.0.2.100  
seed@10.0.2.100's password:  
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-gen  
eric i686)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
1 package can be updated.  
0 updates are security updates.  
  
Last login: Fri May 12 13:07:45 2023 from 10.0.2.4
```

ניתן לראות שהחיבור הצליח
נתחבר לאתר סדרות שוב פעם



ניתן לראות שכעת ניתן לצפות באתר סדרות והצלחנו לבצע מעקף לחוק בחומת האש
שהגדרנו שמונע התחברות לאתר סדרות

הקלטנו את הPACKETS שנשלחו בעת ביצוע החיבור לאתר סדרות בעזרת
הWIRESHARK

426	2023-05-12	13:07...	10.0.2.4	10.0.2.100	SSHv2
427	2023-05-12	13:07...	10.0.2.100	10.0.2.4	TCP
428	2023-05-12	13:07...	10.0.2.100	34.107.221....	HTTP
429	2023-05-12	13:07...	34.107.221....	10.0.2.100	HTTP
430	2023-05-12	13:07...	10.0.2.100	34.107.221....	TCP
431	2023-05-12	13:07...	10.0.2.100	10.0.2.4	SSHv2

בתמונה מעלה רואים שנוצר חיבור SSH בין מחשב A 10.0.2.4 למחשב B
10.0.2.100

לאחר מכן תעבורת הבקשה לאתר סדרות הועברה ממחשב B לשרתי האתר וחזרה
כתשובה בסוף למחשב A כדי שהוא יוכל להתחבר לאתר.

סיכום המשימה

תחילה הגדרנו חוקים במחשב A אשר חוסמים יציאת PACKETS ליצירת חיבור TELNET עם מחשב חיצוני וחוסמים PACKETS לאתר ספציפי

www.sdarot.tw

לאחר מכן יצרנו SSH TUNNEL למחשב B ודרך החיבור הזה הצלחנו במשימה ובצענו מעקף לחסימות שהוגדרו בחומת האש והצלחנו לבצע חיבור TELNET ממחשב A למחשב אחר והצלחנו לבצע חיבור לאתר סדרות דרך מחשב A.

גילינו כיצד ניתן לבצע מעקף לחוקים שהוגדרו בחומת האש, כיצד ניתן לפתוח SSH TUNNEL ואיך היא עוזרת לבצע את המעקף.

גילינו כיצד ניתן להגדיר שתעבורה של אפליקציה תעבור דרך ה SSH TUNNEL שיצרנו בעזרת SOCKS PROXY.

התוצאות התאימו למצופה מאחר והצלחנו להתחבר בTELNET ממחשב A למחשב אחר, וגם הצלחנו להתחבר ממחשב A לאתר סדרות למרות החוקים המוגדרים בחומת האש אשר אוסרים על ביצוע חיבורים אלו.

לא נתקלנו בבעיות במהלך ביצוע המשימה.

Task 4: Evading Ingress Filtering

מבוא:

תיאור

במשימה זו נרצה לעקוף את החוקים המוגדרים בחומת האש בעזרת REVERSE SSH TUNNEL ולגשת ממחשב מרוחק לאתר פנימי שרץ על מחשב אחר.

מטרה

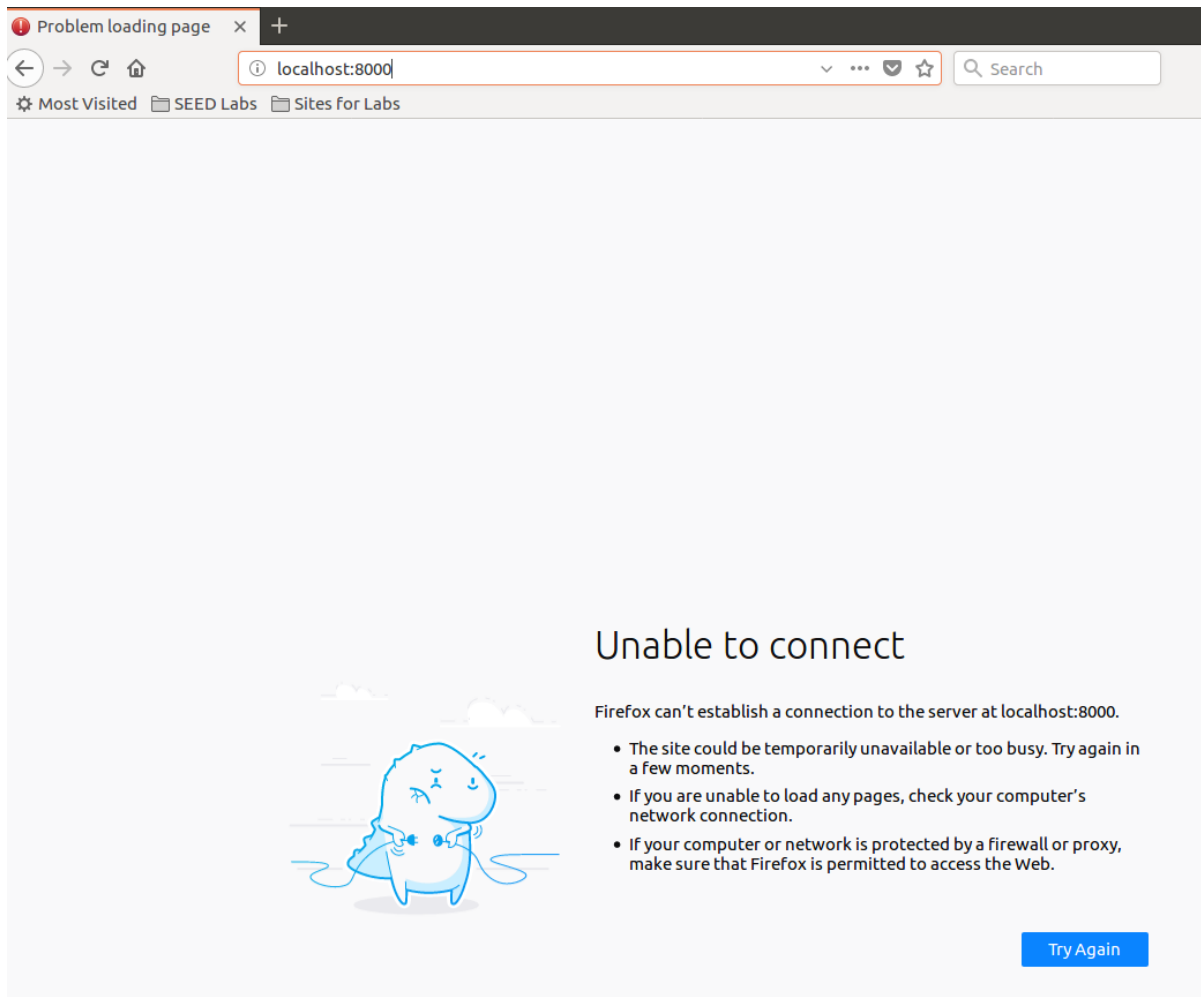
נגדיר REVERSE SSH TUNNEL ממחשב המריץ אתר פנימי שאין למחשבים חיצונים גישה אליו, ובעזרת הTUNNEL שהגדרנו למחשב מרוחק נוכל לגשת מהמחשב המרוחק לאתר הפנימי.

תוצאה מצופה

נצליח לגשת מהמחשב המרוחק לאתר הפנימי אשר רץ על מחשב אחר ממנו יצרנו את הreverse ssh tunnel.

ביצוע המשימה

תחילה ננסה להיכנס דרך מחשב B לאתר localhost:8000



ניתן לראות שלא רץ שום שרת אינטרנטי פנימי על הפורט 8000 ולכן קיבלנו שהחיבור נכשל

כעת נוסיף חוקים לחומת האש אשר חוסמים מחשבים חיצוניים לשליחת יצירות חיבור SSH למחשב A וחוסמים ממחשב B לשלוח בקשות בפורט 80 למחשב A

```
[Fri May 12 14:39:26] Computer A:~$ sudo iptables -A INPUT -p tcp --dport 22 -j DROP
[Fri May 12 14:39:39] Computer A:~$ sudo iptables -A INPUT -p tcp --dport 80 -d 10.0.2.100 -j DROP
[Fri May 12 14:39:47] Computer A:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                                   destination                                tcp dpt:ssh
DROP       tcp  --  anywhere                                anywhere                                tcp dpt:ssh
DROP       tcp  --  anywhere                                10.0.2.100                             tcp dpt:http

Chain FORWARD (policy ACCEPT)
target     prot opt source                                   destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                                   destination
```

ניתן לראות שהחוקים נוספו בהצלחה לטבלת החוקים INPUT

ננסה ליצור חיבור SSH ממחשב B למחשב A

```
[Fri May 12 14:40:24] Computer B:~$ ssh 10.0.2.4
^C
```

ניתן לראות שיצירת החיבור נכשלה

כעת נגדיר REVERSE SSH TUNNEL במחשב A למחשב B

```
[Fri May 12 14:39:58] Computer A:~$ ssh -R 8000:localhost:80 seed@10.0.2.100
seed@10.0.2.100's password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

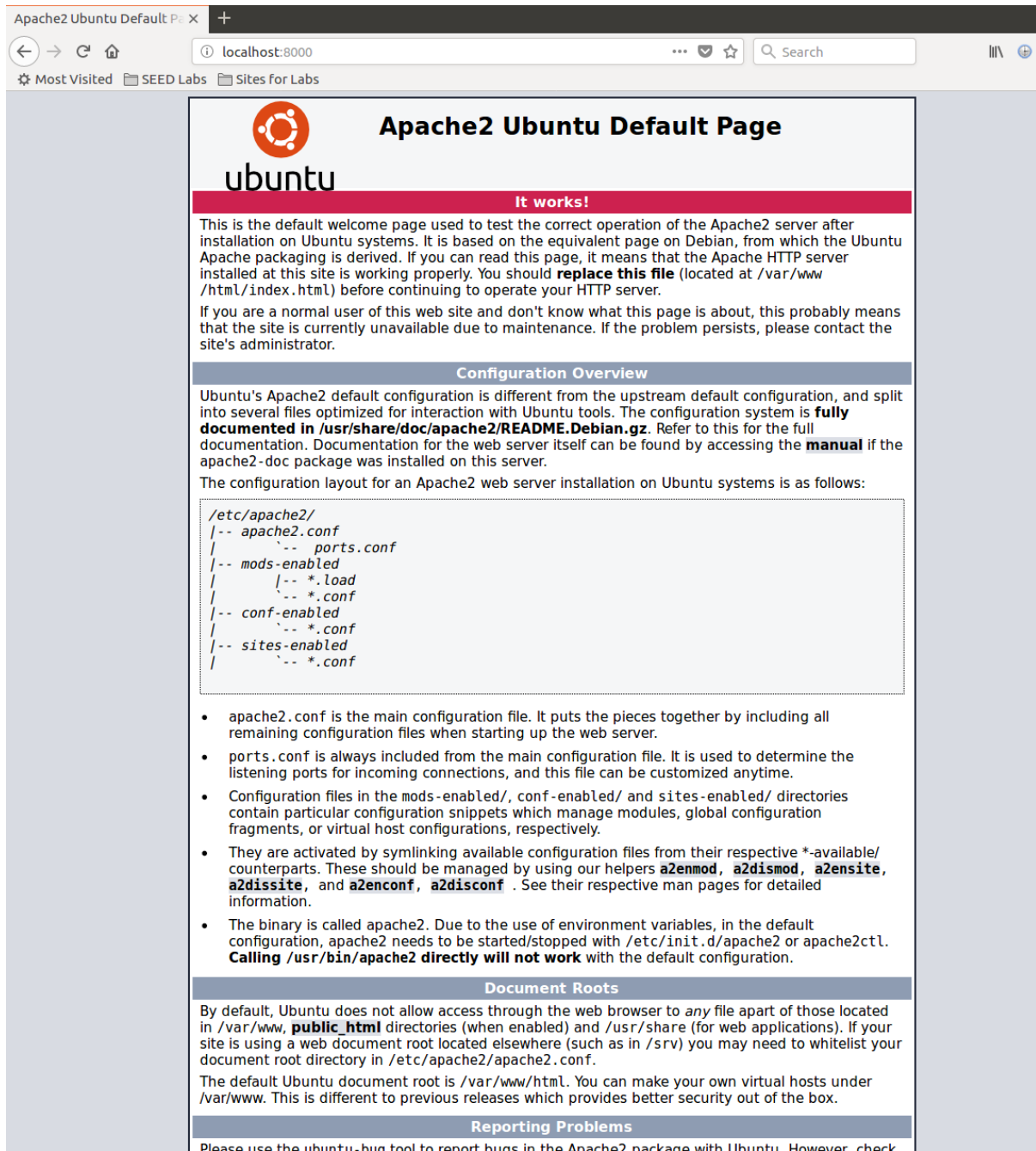
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

Last login: Fri May 12 14:38:08 2023 from 10.0.2.4
```

הreverse ssh tunnel מאפשר למחשב B לשלוח PACKETS למחשב A בפורט 8000 בחיבור SSH שנוצר ממחשב A, ומחשב A יעביר (port forwarding) את PACKETS האלו לפורט 80 וכך למחשב B תהיה גישה לאתר פנימי אשר רץ על מחשב A למרות החסימות שהגדרנו בחומת האש.

כעת ננסה שוב להתחבר ממחשב B לאתר localhost:8000



The screenshot shows a web browser window with the address bar set to `localhost:8000`. The page title is "Apache2 Ubuntu Default Page". The main content area features the Ubuntu logo and the text "It works!". Below this, there is a paragraph explaining that this is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It mentions that the configuration system is fully documented in `/usr/share/doc/apache2/README.Debian.gz`. A section titled "Configuration Overview" follows, detailing the configuration layout for an Apache2 web server installation on Ubuntu systems. It lists the files in `/etc/apache2/` and provides a list of configuration files and their purposes. The "Document Roots" section explains that by default, Ubuntu does not allow access through the web browser to any file apart of those located in `/var/www/public_html` directories (when enabled) and `/usr/share` (for web applications). The "Reporting Problems" section at the bottom suggests using the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu.

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www/public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check

ניתן לראות שהחיבור לאתר הפנימי של מחשב A הצליח מאחר והבקשה ששלחנו לפורט 8000 נשלחה דרך הreverse ssh tunnel שיצרנו למחשב A בפורט 80 וכך נכנסנו לאתר הלוקאלי של מחשב A.

סיכום המשימה

תחילה הגדרנו חוקים במחשב A אשר חוסמים כניסת PACKETS ליצירת חיבור SSH ממחשבים חיצוניים, וחוסמים PACKETS ממחשב B לפורט 80 של מחשב A.

יצרנו REVERSE SSH TUNNEL ממחשב A אשר ממיר את כל הPACKETS שנשלחים ממחשב B בפורט 8000 למחשב A לפורט 80 ומחזיר נתונים משרתי הWEB.

הצלחנו לבצע את המשימה, מאחר ובעת ניסיון התחברות לאתר הפנימי של מחשב A ממחשב B לפני יצירת הTUNNEL לא הצלחנו לגשת לאתר ולאחר הגדרת הTUNNEL הצלחנו לגשת לאתר הפנימי של A ממחשב B.

גילינו כיצד ניתן לתת גישה ממחשב פנימי למחשב מרוחק לשרתים פנימיים הרצים עליו גם אם קיימת חסימה בחומת האש ליצירת חיבור מהמחשב המרוחק למחשב הפנימי ובכך לאפשר עבודה מרוחק.

התוצאות התאימו למצופה מאחר והצלחנו להתחבר לאתר פנימי שרץ על מחשב A ממחשב מרוחק B אשר לא יכל ליצור חיבור SSH למחשב A.

לא נתקלנו בבעיות במהלך ביצוע המשימה.

סיכום כללי למעבדה

תחילה הגדרנו חוקים בחומת האש לחסימות שונות כגון: תקשורת TELNET נכנסת ויוצאת, גישה לאתר ספציפי www.sdarot.tw וכו' בעזרת כתיבת פקודות בטרמינל אשר מבוצעות על חומת האש.

והראנו שלפני הגדרת החוקים האלו החיבורים הצליחו ולאחר הגדרת החוקים הגישות נחסמו ולא יכולנו ליצור את החיבורים האלו.

גילינו כיצד להגדיר חוקים לחסימת גישות עבור תקשורת ספציפית ועבור אתר ספציפי.

לאחר מכן, ביצענו הגדרת חוקים לחומת האש על ידי הכנסת מודולים ל-KERNEL ישירות וטעינתם ללא צורך בביצוע ריסטארט או בנייה מחדש ל-KERNEL.

הגדרנו חמישה חוקים שונים והם:

חסימת TELNET בין A ל B

חסימת TELNET בין B ל A

חסימת חיבור לאתר www.sdarot.tw

חסימת יצירת חיבורי SSH ממחשב A למחשב B

חסימת שליחת PING לאתר www.sdarot.tw

לפני טעינת המודול שבנינו ל-KERNEL ניסיון יצירת החיבורים האלו צלח, ולאחר מכן ניסיון ליצירת החיבורים האלו נכשל בהצלחה.

גילינו כיצד להגדיר חוקים לחסימת גישות בעזרת BACKEND ב-FIREWALL וכיצד לטעון ולהסיר מודולים מה-KERNEL.

כדי להצליח במשימה נעזרנו רבות בדוקומנטציה של NETFILTER בקישור הבא: <https://netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO-4.html>

רצינו לראות כיצד ניתן לעקוף חוקים שהוגדרו בחומת האש למשל איך לעקוף חסימת ייצור תקשורת TELNET בין מחשבים.

תחילה הגדרנו חוקים במחשב A אשר חוסמים יציאת PACKETS ליצירת חיבור TELNET עם מחשב חיצוני וחוסמים PACKETS לאתר ספציפי www.sdarot.tw

לאחר מכן יצרנו SSH TUNNEL למחשב B ודרך החיבור הזה הצלחנו במשימה ובצענו מעקף לחסימות שהוגדרו בחומת האש והצלחנו לבצע חיבור

TELNET ממחשב A למחשב אחר והצלחנו לבצע חיבור לאתר סדרות דרך מחשב A.

גילינו כיצד ניתן לבצע מעקף לחוקים שהוגדרו בחומת האש, כיצד ניתן לפתוח SSH TUNNEL ואיך היא עוזרת לבצע את המעקף.

גילינו כיצד ניתן להגדיר שתעבורה של אפליקציה תעבור דרך ה SSH TUNNEL שיצרנו בעזרת SOCKS PROXY.

לבסוף רצינו לראות כיצד ניתן לתת אופציה להתחבר לאתר פנימי במחשב מרוחק על ידי יצירת חיבור מהמחשב הפנימי למחשב המרוחק ונעזרנו ביצירת REVERSE SSH TUNNEL אשר המירה PACKETS מהמחשב המרוחק בפורט ספציפי לפורט 80 במחשב הפנימי.

תחילה הגדרנו חוקים במחשב A אשר חוסמים כניסת PACKETS ליצירת חיבור SSH ממחשבים חיצוניים, וחוסמים PACKETS ממחשב B לפורט 80 של מחשב A.

יצרנו REVERSE SSH TUNNEL ממחשב A אשר ממיר את כל הPACKETS שנשלחים ממחשב B בפורט 8000 למחשב A לפורט 80 ומחזיר נתונים משרתי הWEB.

בעת ניסיון התחברות לאתר הפנימי של מחשב A ממחשב B לפני יצירת הTUNNEL לא הצלחנו לגשת לאתר ולאחר הגדרת הTUNNEL הצלחנו לגשת לאתר הפנימי של A ממחשב B.

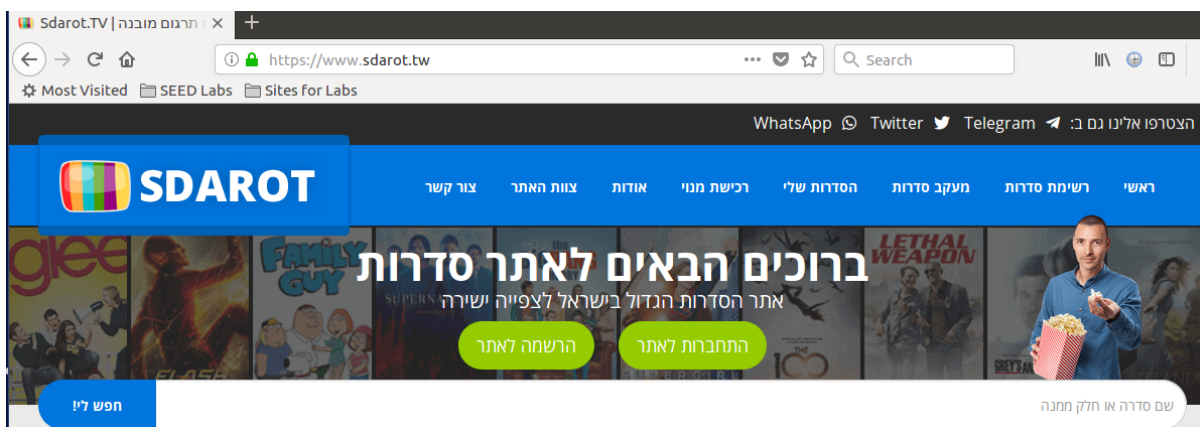
גילינו כיצד ניתן לתת גישה ממחשב פנימי למחשב מרוחק לשרתים פנימיים הרצים עליו גם אם קיימת חסימה בחומת האש ליצירת חיבור מהמחשב המרוחק למחשב הפנימי ובכך לאפשר עבודה מרוחק.

משהו חדשני:

חיפשנו דרך נוספת שבעזרתה יהיה ניתן לעקוף את החוקים אותם אנו מגדירים בחומת האש, ומצאנו שניתן לעקוף את החוקים בעזרת דפדפן TOR אשר מצפין את ה-DATA של ה-PACKET ועל ידי כך מקשה על ה-FIREWALL לזהות את הנתונים של ה-PACKET אותם הוא בודק לצורך השוואה מול החוקים ומצליח לעקוף את החוקים שאמורים לחסום את שליחתה.

איך הוכחנו זאת?

נכנסו לאתר סדרות



ראינו שניתן לגשת לאתר

לאחר מכן, הגדרנו חוקים לחסימת גישה לאתר

```
[Thu May 04 16:36:50] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -s 10.0.2.4 -d 79.133.51.206 -j DROP
[Thu May 04 16:39:09] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -s 10.0.2.4 -d 185.224.81.69 -j DROP
[Thu May 04 16:39:26] Computer A:~$ sudo iptables -A OUTPUT -p tcp --dport 443 -s 10.0.2.4 -d 37.221.65.66 -j DROP
```

נכנסו לדפדפן TOR

TOR יצר חיבור מאובטח עם מחשב IP 144.76.201.253

10.0.2.100	144.76.201.253	TCP	74 39638 → 4080	[SYN] Seq=
144.76.201.253	10.0.2.100	TCP	60 4080 → 39638	[SYN, ACK]
10.0.2.100	144.76.201.253	TCP	54 39638 → 4080	[ACK] Seq=

ניתן לראות שהחיבור נוצר בהצלחה בין מחשב 10.0.2.10 ל-144.76.201.253

חקרנו וגילינו ש-TOR יצר חיבורים נוספים לאורך הדרך מ-IP שונים, TOR מספק הצפנה למידע המועבר וחשאיות כך שלא יהיה ניתן לעקוב אחר השולח. במקרה שלנו TOR הצפין את הבקשה לגלוש לאתר www.sdarot.tv וניגש אל האתר באמצעות IP אחר ובעקבות זאת ה-FIREWALL לא הצליח לחסום את בקשת המחשב לגשת לאתר www.sdarot.tv כפי שהוגדר לו בחוקי

טבלת ה-OUTPUT כי ה-DATA היה מוצפן וה-FIREWALL לא יכל לבצע השוואה עם החוקים.

ניתן לראות שדרך הדפדפן השמאלי של FIREFOX לא היה ניתן לגשת לאתר סדרות אך דרך הדפדפן הימני של TOR הצלחנו לגשת לאתר סדרות בהצלחה.

