LUCRAREA NR. 8
SECURITATEA ÎN REȚELE

1. Obiective

Obiectivele acestui laborator sînt: întelegerea scopului asigurării securității în rețele, familiarizarea cu vulnerabilitățile şi amenințările din rețele, învățarea soluțiilor de rezolvare a problemelor de securitate, familiarizarea cu iptables, configurarea filtrării pachetelor şi utilizarea iptables.

2. Considerații teoretice

2.1 Introducere

Securitatea în rețele de calculatoare începe să fie din ce în ce mai mult luată în serios în ultima perioadă. Motivul pentru această tendință este bineînțeles numărul mare de atacuri. În afară de faptul că numărul de atacuri a crescut îngrijorător de mult în ultima perioadă, informația digitală devine din ce în ce mai valoroasă pentru un atacator (numere de cărți de credit, informații confidențiale, tranzacții bancare) astfel încât acesta este dispus să investească mai mult timp şi bani pentru a căpăta acces la aceste informații. Soluția folosită până nu de mult, era ``baricadarea'' informațiilor confidențiale în fortărețe impenetrabile pentru oricine. Era comun pentru o companie mare să păstreze informațiile confidențiale pe un mainframe la care accesul se putea face doar din sala de terminale. Mai târziu, au apărut rețele de calculatoare locale, dar ele erau izolate de exterior, astfel încât securitatea în acele rețele de calculatoare se implementa printr-o politică extrem de severă de pedepsire a celor vinovați (concedierea sau intentarea unui proces celui vinovat erau măsurile cele mai des folosite). O dată cu apariția Internetului rețelele de calculatoare au încetat să mai fie forturi impenetrabile, trebuind să-şi deschidă porțile astfel încât compania să poată rămâne competitivă.

Securitatea rețelelor este procesul prin care informația digitală este protejată. Asigurarea securității în rețea vizează confidențialitatea – calitatea unei rețele de a asigura accesul la informație doar persoanelor autorizate, integritatea – calitatea unei rețele de a asigura că informația nu a fost

modificată de persoane neautorizate, disponibilitatea - timpul în care rețeaua de calculatoare şi resursele din cadrul ei sunt operaționale.

Vulnerabilitatea este o slăbiciune a sistemului care permite o acțiune neautorizată. Acestea sunt erori care apar în diferite faze ale dezvoltării, respectiv folosirii sistemelor. Acestea pot fi clasificate în vulnerabilități de proiectare – eroare care apare în prima fază a vieții unui produs, aceea de concepție, şi pe care chiar o implementare ulterioară perfectă nu o va înlătura, vulnerabilități de implementare şi vulnerabilități de configurare – apare ca urmare a erorilor făcute în configurarea sistemelor, cum ar fi folosirea codurilor de acces implicite sau a drepturilor de scriere a fişierelor cu parole.

2.2 Elementele de baza ale securității rețelelor

Trei concepte de bază ale securității, importante în ceea ce priveşte informațiile de pe Internet, sunt confidențialitatea, integritatea şi disponibilitatea. Conceptele legate de oamenii care utilizează aceste informații sînt autentificarea, autorizarea şi acceptarea.

O soluție de securitate conține cinci elemente de bază:

- identificarea, autorizare şi monitorizarea activității utilizatorilor: identificarea precisă şi pozitivă a utilizatorilor, hosturilor, aplicațiilor, serviciilor şi resurselor din rețea. Tehnicile standard pentru asigurarea identificării includ protocoale de autentificare RADIUS (Remote Access Dial-In User Service), TACACS+ (Terminal Access Controller Access Control System Plus), Kerberos, certificate digitale, smart card-uri, servicii de tip directory;
- securizarea perimetrului rețelei**:** controlul accesului utilizatorilor la aplicațiile critice, la datele şi serviciile din rețea. Acest tip de control se realizează pe echipamentele de rutare prin filtrarea pachetelor şi firewall sau prin firewall-uri dedicat.
- asigurarea confidențialității şi integrității datelor: prin tehnologii de tunelare, GRE (Generic Routing Encapsulation) sau L2TP (Layer 2 Tunneling Protocol), tehnologii de criptare şi protocoale de criptare.
- monitorizarea securității: pentru a asigura persistența securității este important ca rețeaua să fie testată şi monitorizată prin scanere de vulnerabilitate şi sisteme de detecție a intruziunilor;

- managementul echipamentelor şi infrastructurii de securitate.

2.3 Vulnerabilități şi amenințări în rețea

Vulnerabilitățile sunt probleme ale sistemelor de operare, protocoalelor TCP/IP, dispozitivelor de rețea prin care un atacator poate accesa rețeaua fără a respecta politica de securitate implementată.

2.3.1 Recunoaşterea

Recunoaştere se poate defini ca procesul prin care un atacator descoperă maparea sistemelor, a serviciilor şi vulnerabilităților în rețea. În această fază atacatorul strânge informații şi, de cele mai multe ori, această fază precede un atac efectiv de tip DoS. Într-o primă fază atacatorul foloseşte utilitare gen nslookup sau whois pentru a descoperi spațiul de adrese alocate organizației țintă. Apoi, prin ping sweep încercâ să determine care din adresele de IP sunt alocate şi care din sisteme sunt pornite. Se foloseşte apoi un port scanner pentru a determina ce servicii sunt active. După determinarea serviciilor şi sistemului de operare, atacatorul încearcă să obțină versiunea sistemului de operare şi versiunile serviciilor rulate. Acest lucru se poate face prin conectarea cu utilitare de gen telnet pe portul serviciului respectiv şi examinarea mesajele afişate. Pe baza acestor informații atacatorul poate determina ce vulnerabilități există şi ce sisteme poate ataca.

2.3.2 Inteceptarea rețelei (sniffing)

Interceptarea rețelei se face prin captarea pachetelor TCP sau a altor pachete şi decodificarea conținutului lor cu ajutorul unui analizor de protocoale.

2.3.3 Obținerea accesului

Obținerea accesului constă în abilitatea unui intrus neautorizat de a obține acces la echipamente pentru care nu deține cont de utilizator sau parolă. Obținerea accesului se face prin execuția unui script sau utilizarea unui tool care exploatează o vulnerabilitate cunoscută a sistemului sau a aplicației.

Una dintre cele mai sigure metode de obținere a accesului privilegiat este a sparge parola. Acest lucru presupune că atacatorul are deja acces pe o maşină din rețea şi doreşte acces privilegiat (root, Administrator). Deşi un atac ``brute force'' nu are cum să dea rezultate decât pe sistemele la care

parolele sunt limitate la 6-7 caractere, există atacuri care se folosesc de unele particularități ale parolelor.

Unul dintre cele mai dese tipuri de exploit-uri de la distanță este buffer overflow. Această tehnică se bazează pe bug-uri în aplicații. Astfel, există aplicații care acceptă şiruri de caractere de la utilizatori. Aceste şiruri sunt copiate apoi în memorie pentru a fi utilizate, de multe ori într-o variabilă locală alocată pe stivă. Dacă variabila este alocată cu o dimensiune fixă, şi datele acceptate de la utilizator depăşesc dimensiunea variabilei, acestea vor suprascrie în stivă adresa de retur din procedură cu alta, aleasă de atacator să indice undeva în şirul introdus. Astfel se poate executa cod arbitrar pe maşina atacată. Folosirea exploit-urilor de tip buffer overflow, combinată cu faptul că unele programe au bitul *set uid* setat poate crea exploit-uri locale eficiente. De aceea, la securizarea unei maşini trebuie avute în vedere aceste programe.

Altă metodă de exploit-uri de la distanță este deturnarea unei conexiuni TCP (TCP session hijack). Ea constă în aşteptarea ca un utilizator să se logheze pe sistem ce doreşte să fie atacat, şi apoi în trimiterea de pachete către portul pe care rulează serviciul, luând locul utilizatorului care s-a autentificat. Această metodă se foloseşte dacă se pot prezice numerele de secvență dintr-un pachet TCP. O variantă de deturnare de conexiuni TCP este man in the middle attack. În acest caz, atacatorul trebuie să aibă acces la o maşină pe care trece traficul dintre două entități A şi B. În acest caz, atacatorul interceptează cererea de conexiune de la A la B şi răspunde lui A, stabilind cu A o conexiune. Apoi stabileşte şi cu B o conexiune. Toate datele trimise de A vor fi apoi trimise lui B şi invers. Astfel atacatorul are acces la convorbirea dintre A şi B, chiar dacă traficul este criptat din punctul de vedere al lui A şi B.

Viruşii pot constitui de asemenea metode eficace de atac, atunci când poartă cu ei troieni. Troienii sunt programe simple, care deschid uşi de acces pe sistemele infectate de virus.

Falsificarea IP (IP spoofing) este o metodă de atac, dar poate fi folosită şi pentru a ascunde identitatea atacatorului sau pentru a lansa atacuri. Prin acest atac, pachetele TCP/IP sunt manipulate, falsificând adresa sursă. În acest mod atacatorul poate căpăta acces atribuindu-şi o identitate (adresa de IP) care are autorizare să acceseze resursa atacată. Datorită falsificării adresei sursă a pachetului IP, atacatorul nu poate stabili decât o comunicație unidirecțională (presupunând că nu este prezent în rețeaua locală a maşinii

atacate). Acest lucru face protocolul TCP nesusceptibil pentru asemenea atacuri. Există însă numeroase servicii UDP care pot fi exploatate cu acest tip de atac.

2.3.4 Denial of Service

Atacurile de tipul denial of service (DoS) opresc sau încetinesc foarte mult funcționarea unor rețelele, sisteme sau servicii. Ele sunt cauzate de un atacator care doreşte să împiedice accesul utilizatorilor la resursele atacate. Atacatorul nu are nevoie să fi căpătat înainte acces pe calculatorul pe care doreşte să efectueze atacul. Există multe posibilități prin care un atac DoS se poate manifesta. Efectul însă este acelaşi: se împiedică accesul persoanelor autorizate de a folosi serviciile de pe sistem prin utilizarea la maxim a resurselor sistemului de către atacator.

Tipuri de amenințări de tip DoS:

- ping of death**:** foloseşte pachete IP modificate care indică faptul că pachetul are mai multe date decât are de fapt. Acest atac determină blocarea sau resetarea maşinii pe sistemele care nu verifică acest lucru;
- atac de tip SYN flood**:** deschide foarte multe conexiuni pe o maşină. În cazul în care sistemul de operare nu limitează numărul de conexiuni deschise, acest proces duce la încetinirea procesării traficului şi la consumarea inutilă a memoriei pe maşina atacată. Dacă se deschide un număr suficient de mare de conexiuni, până la urmă maşina va cedacereri multiple de realizare de conexiuni;
- fragmentarea pachetelor şi reasamblare;
- bombe e-mail (e-mail bombs): trimiterea repetată a unui mesaj (de dimensiuni mari) spre o adresă de e-mail a unui utilizator;
- CPU hogging**:** programe de genul cai troieni sau viruşi care afectează procesorul, memoria sau alte resurse;
- applet-uri malițioase: programe Java, Java Script sau ActiveX care acționează ca şi caii troieni sau viruşii;
- atacuri out-of-band (de exmplu, WinNuke) – transmitere de date pe portul 139 pe sistemele Windows 95 şi Windows NT. Acest atac va determina sistemul de operare să se blocheze sau să se reseteze. Datele out of band sunt date care nu fac parte din fluxul normal de date schimbat între doi socketi. Acest tip de date sunt

trimise doar în cazuri speciale şi au prioritate față de datele normale.;

- Land.c: program ce generează pachete TCP SYN care specifică adresa hostului ca adresă sursa şi adresă destinație. Deasemnea, programul utilizează acelaşi port al hostului (113 sau 139) ca port sursă şi port destinație cauzînd oprirea sistemului;
- Teardrop.c: implementează fragmentarea IP astfel încît reasamblarea la destinație să necesite foarte multe resurse şi să ducă la oprirea sistemului.

2.3.5 Atacuri distribuite (DDoS - Distributed Denial of Service)

Atacurile DoS distribuite sunt astfel concepute încât să satureze lărgimea de bandă pe legătura ce conectează rețeaua la Internet cu pachete de date trimise de atacator, astfel încât pachetele legitime nu mai pot fi trimise. Pentru a realiza acest lucru un atacator se foloseşte, direct sau indirect, de mai multe sisteme Atacul funcționează în două etape: atacatorii infiltrează viruşi sau viermi în zeci, sute sau chiar mii de hosturi, hosturile infectate executînd apoi un program de atac care aşteaptă comenzi de la distanță şi în momentul primirii unei astfel de comenzi pornesc un atac de tip concertat asupra victimei.

Exemple de atacuri DDoS sînt: Smurf, Tribe Flood Network, Stacheldraht.
Smurf începe prin a trimite un număr mare de mesaje ICMP de tip echo-request (sau ping) către adrese de broadcast, sperând că aceste pachete vor fi trimise unui întreg segment de rețea. De asemenea aceste pachete sunt falsificate pentru a avea ca adresă sursă adresa sistemului țintă. Dacă pachetul trece de dispozitivul de rutare, el va fi recepționat de către toate stațiile de pe un segment de rețea. Acestea vor răspunde cu un pachet de tip echo-reply către adresa falsă din pachet. Astfel, stațiile vor genera trafic către adresa specificată de atacator. Acest tip de atac poate fi uşor contracarat dacă pe ruter se dezactivează rutarea pachetelor de broadcast direcționat.

Tribe Flood Network şi Tribe Flood Network 2000 au capacitatea de a genera pachete de IP cu adresa sursă falsificată. În prealabil însă, sistemele de pe care se face atacul trebuie sa fi fost instalate cu aceste utilitare. Acest lucru se face în primul pas, când se atacă stațiile (denumite drone-uri). Un

TFN master poate apoi comanda drone-urile cauzând atacuri DoS distribuite.

2.4 Soluții de securizare a rețelelor

2.4.1 Firewall-uri

Cea mai utilizată soluție de securizare a unei rețele este firewall-ul, un sistem sau un grup de sisteme care implementează politica de acces între două sau mai multe rețele. Firewall-urile sînt de trei categorii:

- firewall-uri dedicate: maşini ce rulează un sistem de operare special conceput pentru filtrarea de pachete şi translatarea de adrese;
- firewall-uri de servere: rulează ca un software suplimentar pe sistemele de operare de rețea (Linux, NT, Win2K, Novell). Exemple: Microsoft ISA Server, Netfilter, Novell BorderManager si Check Point Firewall-1;
- firewall-uri personale: sunt instalate pe calculatoarele personale. Ele sunt concepute pentru a preveni atacuri doar asupra calculatorului pe care rulează.

2.4.1.1 Filtrarea pachetelor

Filtrarea de pachete este procesul prin care firewall-ul lasă să treacă în rețeaua locală doar anumite pachete, pe baza unor reguli. Filtrarea de pachete este folosită pentru a proteja o rețea de atacuri din exterior (Internet).

Regulile de filtrare sunt formate dintr-o parte care identifică pachetul şi o parte care specifică cum să se trateze pachetul. În partea de identificare se poate specifica adresa sursă, adresa destinație, adresa de rețea sursă, adresa de rețea destinație, protocolul (TCP, UDP, ICMP), portul sursă sau destinație (doar pentru TCP sau UDP), tipul mesajului (pentru ICMP), interfața de intrare sau ieşire, adresele MAC. Deciziile care pot fi luate vizează: acceptarea pachetului, ignorarea pachetului (eliminarea pachetului şi nenotificarea sursei), elimiarea pachetului (rejectare însoțită de notificarea sursei).

Filtrele de pachete pot fi statice sau dinamice. Filtrele statice determină acceptarea sau blocarea unui pachet pe baza informației din header. Aceste filtre sînt implementate în sistemele de operare şi rutere şi utilizează un set de reguli pentru determinarea sorții unui pachet. Administratorul este cel care crează regulile sub forma unei liste ordonate, fiecare pachet care soseşte fiind comparat cu fiecare regula pîna se realizează un matching. Dacă nu se realizeaza nici un matching se aplică regula implicită – deny all. Filtrele dinamice operează într-o manieră similară dar pastrează şi informații legate de sesiunea de comunicare între două hosturi pentru a controla fluxul dintre cele două prin deschiderea/închiderea dinamică a porturilor necesare pentru comunicare. Filtrele dinamice sînt implementate în produsele de tip firewall pentru a controla traficul înspre/dinspre o rețea. Filtrarea dinamica se aplică doar în cazul pachetelor TCP, nu e aplicabilă pentru pachetele ICMP şi UDP datorită neorientării pe conexiune a acestor protocoale.

2.4.1.2 Translatarea de adrese

Translatarea de adrese sau NAT este procesul prin care un ruter modifică adresele sursă sau destinație din anumite pachete care trec prin ruter pe baza unor reguli.

Avantajul folosirii translatării de adrese dinamice constă în faptul că se poate folosi o partajare a adreselor rutabile disponibile organizației. Astfel, calculatoarelor din rețeaua locală li se alocă adrese private, iar ruterul va face o translatare de adrese dinamice din mulțimea de adrese private în mulțimea de adrese publice alocate organizației. Se observă însă că această abordare permite ca doar calculatoare din rețeaua locală să aibă conversații TCP sau UDP cu Internetul. Alt avantaj al folosirii translatării de adrese este acela că se ascunde astfel exteriorului maparea reală de adrese. Translatarea de adrese statică se foloseşte atunci când în rețeaua locală avem un server pe care dorim să îl accesăm din exterior. În acest caz se face o mapare unu la unu între adresa din interior şi cea din exterior.

2.4.2 Sisteme de detectie a intruziunilor

Sistemele de detecție a intruziunilor au abilitatea de a detecta atacurile împotriva unei rețele. Aceste sisteme identifică, opresc şi semnalează atacurile asupra resurselor rețelei. Există două tipuri de sisteme de detecție a intruziunilor: pentru stații si pentru rețele.

Un HIDS (Host based IDS) sau un sistem de detecție a intruziunilor pentru stații înregistrează atât operațiile efectuate, cât şi utilizarea resurselor sistemului. Un avantaj al HIDS este faptul că el poate preveni atacuri necunoscute. De exemplu un HIDS poate monitoriza accesul la fişiere şi reacționa când un atacator încearcă să şteargă fişiere critice. Chiar dacă tipul atacului este nou şi nu poate fi recunoscut de un NIDS (Network based IDS) un HIDS poate sesiza atacul.

Cea mai simplă formă de HIDS este pornirea proceselor de logare pe sistem. O astfel de metodă se spune că este pasivă. Dezavantajul acestei metode este faptul că necesită multe ore de muncă din partea administratorului pentru a analiza logurile. Sistemele HIDS curente folosesc agenți software care sunt instalați pe fiecare maşină şi care monitorizează activ sistemul (pot reacționa dacă detectează atacuri). Atunci când HIDS-ul este configurat să răspundă activ, el va opri serviciile de rețea pentru a preveni eventuale pagube şi a putea analiza exact atacul. Un exemplu de sistem de detecție a intruziunilor este Linux Intrusion Detection System (LIDS).

NIDS-urile sunt dispozitive de inspectare a traficului şi acționează prin colectarea de date de la senzori amplasați în rețea. NIDS-urile captează si analizează traficul ce traversează rețeaua. Avantajul folosirii unui NIDS este faptul că nu trebuie aduse modificări pe stații. În schimb, datorită faptului că la baza NIDS-urilor stă detecția bazată pe semnături ale atacurilor, el nu poate opri sau detecta atacuri noi. Un exemplu de sistem de detecție a intruziunilor în rețea este SNORT.

2.5 Iptables

Iptables este utilitarul cu ajutorul căruia se pot configura politica şi regulile de filtrare de pachete sau translatare de adrese pentru Linux 2.4. El face parte din Netfilter, ce implementează în Linux filtrarea de pachete şi NAT.

2.5.1 Tabele, lanturi, actiuni

În iptables o regulă are două părți: o parte care identifică pachetele (partea de match) şi una care specifică cum trebuie tratate pachetele respective (partea țintă). Procesarea regulilor se face secvențial începând cu prima regulă. Dacă regula curentă face match se execută acțiunea asociată țintei. Dacă nu, se trece la următoarea regulă. Dacă s-au epuizat toate regulile dintr-un lanț definit de utilizator sau dacă ținta este RETURN**,** se continuă

analizarea regulilor din lanțul precedent. Dacă s-au epuizat toate regulile dintr-un lanț predefinit, se execută acțiunea asociată politicii implicite a lanțului.

Există trei tabele independente: filter - tabelul implicit, conține lanțurile predefinite INPUT (pentru pachetele care sosesc la host), FORWARD (pentru pachetele care sînt rutate prin host), OUTPUT (pentru pachetele generate de host); nat - tabelul consultat cînd apare un pachet de creare a unei conexiuni noi, conține lanturile predefinite: PREROUTING OUTPUT, POSTROUTING; mangle - tabel utilizat pentru modificări specializate, conține lanțurile predefinite: PREROUTING, OUTPUT.

Tipuri de ținte: ACCEPT (aceptarea pachetului), DROP (eliminarea pachetului), REJECT (eliminarea pachetului cu notificarea sursei printr-un mesaj de eroare ICMP 'port unreachable'), QUEUE (pachetele sint puse in buffere.

2.5.2 Utilizarea Iptables

În cadrul iptables se pot creea/şterge/modifica lanțuri, afişa regulile dintr-un lanț, schimba politicile dintr-un lanț, şterge regulile dintr-un lanț.

În cadrul unui lanț există mai multe posibilități de manipulare a regulior: adăugarea unei noi reguli, inserarea unei noi reguli pe o anumit poziție în lanț, înlocuirea unei reguli dintr-o anumită poziție, ştergerea unei reguli din lanț.

Cele mai utilizate comenzi sînt adaugare (-A) şi stergere (-D), celelalte comenzi (-I pentru inserare şi -R pentru înlocuire) sînt extensii ale celor două.

Comenzi pentru manipularea regulilor:

// adaugă o noua regulă în tabela *table,* lanțul *chain* unde *packet* reprezintă o serie de opțiuni ce identifică o clasă de pachete, iar *target* reprezintă ținta

```
ipchains -t table -A chain packet -j target
```

// şterge regula cu numărul *rule_no* (numerotarea începe de la 1) din tabela *table*

```
ipchains -t table -D chain rule_no
```

// înlocuiește regula cu numărul *rule_no* cu una nouă definită prin *packet* şi *target* din tabela *table*
```
ipchains -t table -R chain rule_no packet -j target
```

// inserează o nouă regulă în tabela *table* pe poziția *rule_no* definită prin *packet* şi *target*
```
ipchains -t table -I chain rule_no packet -j target
```

Comenzi pentru manipularea lanțurilor:

//crearea unui lanț nou în *table*
```
ipchains -t table -N nume_lanț
```

//ştergerea unui lanț din *table*
```
ipchains -t table -X nume_lanț
```

//definirea politicii unui lanț
```
ipchains -t table -P target
```

Un pachet poate fi identificat prin adresa sursă, adresa destinație, tipul pachetului, port (TCP, UDP), tipul mesajului (ICMP), interfața de intrare/ieşire sau prin caracteristica de fragmentat/nefragmentat. Opțiunile pentru identificarea unui pachet sînt:

//utilizarea unei negații
```
! argument
```

//identificarea protocolului: icmp, tcp, udp
```
-p [!] protocol
```

//adresa sursă
```
-s [!] address[/mask] [!] [icmp_type_no | [port][:port]]
```

//adresa destianație
```
-d [!] address[/mask] [!] [icmp_type_no | port][:port]]
```

//tipul mesajului icmp: *destination-unreachable, port-unreachable, echo-request, echo-reply*; pentru a afla toate tipurile de mesaje executați comanda *iptables -p icmp -h*
```
-icmp-type icmp_name
```

//port destinație şi port sursă
```
-destination-port port
-source-port port
```

//interfața de intrare/ieşire (ppp0, eth0, eth1 etc)
```
-i [!] interface_name[+]
-o [!] interface_name[+]
```

//acest pachet e fragmentat
```
 [!] -f
```

//adresa MAC a pachetului
```
-mac-source [!] mac-address
```

2.5.3 Exemplu

O regulă specifica un set de condiții pe care un pachet trebuie să le indeplinească şi o țintă. Să presupunem că vrem să eliminăm toate pachetele ICMP sosite de la adresa IP 127.0.0.1. În acest caz condițiile sînt ca protocolul să fie ICMP şi adresa sursă să fie 127.0.0.1. Ținta este DROP.

```
# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.2 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms

# iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP

# ping -c 1 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss
```

În exemplul de mai sus, se poate observa că primul `ping` are loc cu success. (opțiunea -c 1 determină trimiterea unui singur pachet).
Adăugam (-A), o regulă în lanțul INPUT care specifică ca pachetele sosite de la adresa sursă 127.0.0.1 (-s 127.0.0.1), pachete ICMP (-p icmp) să fie elimitate (-j DROP).

O regulă se poate şterge prin două metode. În exemplul de mai sus, ştim că regula definită de noi este singura regula din lanț deci putem şterge regula cu numarul 1 din lanțul INPUT.

```
# iptables -D INPUT 1
```

A doua metodă este utilizarea comenzii prin care s-a adăugat regula, (-A), în care înlocuim (-A) cu (-D). Această metodă este utilă în cazul în care avem un lanț complex (cu multe reguli):

```
# iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP
```

Sintaxa de ştergere trebuie să aiba aceleaşi opțiuni ca sintaxa de adăugare. Dacă exista mai multe reguli identice în lanț, prima regulă găsită este ştearsă.

3. Desfăşurarea lucrării

3.1 Studiați comenzile utilizate pentru Iptables.
3.2 Rulați exemplul de la 2.5.3.
3.3 Scrieți un set de reguli de filtrare pentru ca utilizatorii să aibă acces doar la trafic de tip ssh, web şi ftp.

Notițe