INTRODUCERE ÎN SECURITATE CIBERNETICĂ ȘI HACKING

CONCEPTE DE HACKING ȘI CYBERSECURITY

Ramon Nastase

Cuprins

Introducere

In primul rand vreau sa te felicit si sa iti multumesc pentru faptul ca ai luat **decizia de a investi in tine** si de a deveni mai bun. Vreau sa-ti spun acest ghid de va lua de la 0 (in domeniul Securitatii Cibernetice) si te va duce la un nivel de baza astfel incat sa fi constient de lucrurile care se intampla in jurul nostru, in Internet in fiecare zi.

Imi doresc ca aceasta carte sa te schimbe. Sa-ti schimbe mentalitatea, modul in care gandesti si sa-ti dea o perspectiva noua asupra lucrurilor. Conceptele explicate aici sunt atat teoretice cat si practice.

Hai sa-ti arat cateva din lucrurile pe care le vei invata aici:

- *Cum sa gandesti ca un **Hacker***
- *Care sunt pasii unui **Atac Cibernetic***
- *Cum sa Scanezi si sa vezi **Traficul** altora folosind **Kali** Linux*
- Concepte de Securitate & Hacking pentru retele
- Si multe altele :)

Aceasta carte este structurata in 5 capitole care cuprind diferite teme, apartinand conceptelor de baza din CyberSecurity. Un lucru pe care vreau sa-l stii este faptul ca daca acum incepi pentru prima oara in IT, aceasta carte nu este alegerea cea mai potrivita. De ce ? Pentru ca ai nevoie de cunostinte (cel putin medii) de Linux, Retelistica si (putina) programare pentru a putea intelege o parte din lucrurile pe care le explic eu aici.
De asta vreau sa-ti spun (inca de la inceput) faptul ca: inainte de a invata sa spargi si sa securizezi lucrurile este important sa intelegi cum functioneaza tehnologia. Avand acest lucru in minte, iti urez mult spor in ceea ce faci, trage cat mai tare pentru ca in final, vei vedea, cu munca si efor sustinut in mod constant vei ajunge sa realizezi ceea ce ti-ai propus.

Seteaza-ti tinte inalte,
Ramon (cel care te sustine in procesul tau de crestere)

PS: iar daca ai intrebari nu ezita sa ma contactezi pe **email**, **Facebook** sau **YouTube**.

I. Procesul de Hacking

In general cand vorbim de Hacking exista o structura foarte bine gandita in spate. Nu vrem sa gasim un server si sa “sarim” direct pe el pentru ca avem prea putine informatii despre acesta pe moment si ne expunem la riscul de a fi prinsi daca nu luam in calcul cei 5 pasi existenti in acest proces.

Cum se desfasoara procesul de Hacking ?

Sper ca observi ca am spus “*procesul de Hacking*”, proces care poate dura si cateva zile, saptamani, chiar luni (depinde de tinta si de riscul existent). Acest proces cum am spus si mai devreme este alcatuit din 5 pasi (figura 1.1):

1. **Reconnaissance -** Information Gathering
2. **Scanning**
3. **Gaining Access**
4. **Maintaining Access**
5. **Covering Tracks**

Figura 1.1

Iar acum haide sa luam pe rand si sa discutam despre fiecare in parte:

1) Reconnaissance - "*Information Gathering*"

Unul dintre cele mai importante lucruri pe care Hackerii il fac in momentul in care s-au decis ca vor sa atace un sistem (server, retea etc.) este sa acumuleze cat mai multe date despre el.

Gandeste-te in felul urmator: in momentul in care vrei sa pleci intr-o vacanta intr-un loc/tara in care nu ai mai fost, ce faci ? Cel mai probabil iti faci temele de casa. Adica te interesezi de acea locatie. Cauti pe Google diferite lucruri (ce poti face acolo, cum este vremea/mancarea, review-urile localurilor din zona etc.). Cu alte cuvinte **te informezi despre tinta ta**.

Exact prin acest proces trece si un Hacker in momentul in care decide sa atace un sistem. Exista diferite metode prin care poti afla mai multe despre un site/server, una dintre cele mai simple metode este sa cauti pe Google informatii despre acesta.

Printr-o comanda simpla precum **nslookup** (sau **dig**) poti afla cu adresa IP a unui site, iar prin comanda whois poti afla mult mai multe informatii despre acel domeniu.

>**nslookup** google.ro
>**whois** google.ro

Termenul de Reconnaissance (sau de Information Gathering) vine de la ideea de a cerceta, de a te informa despre un anumit subiect inainte de a trece la actiune. Mai pe scurt, practic inseamna **documentare** inainte de **actiune**.

Ca interval de timp acest proces este cel mai "costisitor". De ce ? Pentru ca un atacator trebuie sa fie foarte bine informat, trebuie sa cunoasca lucrurile in amanunt pentru ca altfel (asa cum am spus si la pasul #5) isi risca propria libertate.

2) Scanning - "*Scanarea sistemului*"

Urmatorul pas in "Procesul de Hacking" este **scanarea**. Odata ce un Hakcer are mai multe informatii despre tinta sa va incepe sa afle si mai multe informatii (de data aceasta tehnice).
Si cum va face asta ? Folosind diferite unelte (precum Nmap) cu care se pot scana retele, servere si care ii ofera informatii mult mai clare despre topologia retelei, despre echipamentele folosite, sistemul de operare etc.

De ce sunt acestea importante ? De ce este important sa stie un Hacker daca un anumit server web ruleaza pe Windows sau pe Linux ? Pentru ca odata ce are aceasta informatie poate sa mearga mai departe (la pasul 3), cu un mic research pe Google, sa descopere anumite vulnerabilitati existente si sa incerce sa profite de ele cu scopul de obtine acces in acel sistem (sau de a extrage anumite date).

Despre scanare si diferitele metode prin care putem face asta vom vorbi mai pe larg in capitolul 6. Cu ajutorul acestor date acumulate din urma scanarii, Hackerul va trece la pasul #3.

3) Gaining Access - "*Obtinerea Accesului*"

Avand temele facute (a facut research, a scanat retelele/serverele, a aflat informatii din diferite surse - Google, Facebook, Forumuri - despre tinta), Hackerul poate incepe atacul. Atacul trebuie gandit foarte bine pentru a fi in modul stealth (fara a declansa alarme si - daca se poate - fara a generea prea multe log-uri).

Exista foarte multe **tool-uri** (*Burp Suite, SQLmap, Metasploit* etc.) care pot fi folosite pentru a genera un atac cibernetic, totul depinde de tehnologie si obiectiv.

Obtinerea accesului se poate face in mai multe moduri si din mai multe puncte de vedere:

- Obtinere acces la nivel de **root** pe un server Linux
- Obtinere acces la **panoul de administrare** al unui site
- Obtinere acces pe un anumit **echipament** din **retea** (Router, Firewall, Switch etc.)
- Obtinere acces pe un **end-device** din retea (smartphone, tableta, laptop etc.)

Odata ce Hackerul are acces pe unul dintre elementele enumerate mai devreme, el este infiltrat in retea si astfel poate obtine foarte multe informatii despre organizatia in care se afla (digital).

Vom discuta in capitolul 5 mai multe despre cateva tipuri de atacurile cibernetice si cum le putem face. Totodata in **cursul de Securitate** iti arat pas cu pas cum poti face aceste atacuri cibernetice indiferent ca este vorba de retea, wireless, servere sau site-uri.

4) Maintaining Access

Odata intrat in retea, Hacker are optiunea de a-si mentine accesul. In foarte multe situatii cand au fost sparte diferite servere ale marilor companii (Yahoo, Google, Microsoft etc.), Hackerii si-au lasat mereu “portite deschise” pentru a intra inapoi in sistem.

Aceste portite se numesc “**backdoor**” si sunt lasate intentionat de catre Hackeri (sau chiar de catre dezvoltatorii de software ale unor aplicatii pe care tu si eu le folosim zi de zi) pentru a avea acces ulterior in sistem.

Astfel ei pot extrage in mod constant date, pot urmarii ce se intampla in organizatii, pot detine “controlul din spate”, urmand ca ulterior sa faca ceva cu aceste date (de obicei ele sunt vandute pe piata neagra din Deep Web).

Dupa acest proces, urmeaza pasul #5 care este extrem de important.

5) Covering Tracks - “Acoperirea Urmelor”

Acest proces este unul foarte important (cel de “Acoperire a urmelor lasate”). Un proces pe care foarte multi Hackeri (mai ales cei care sunt la inceput de drum) il omit. Pur si simplu nu sunt atenti (sau constienti) sa-si acopere urmele si ajung sa fie prinsi (in Romania de DIICOT, SRI sau STS) si pedepsiti in instanta pentru faptele facute.

Repet faptul ca **accesul neautorizat** intr-un sistem poate duce la consecinte grave din punct de vedere penal:

- confiscarea bunurilor informatice - laptop-uri, hard disk-uri externe etc.
- punerea sub supraveghere
- sau chiar arestul, acestea fiind doar cateva dintre consecintele

Pentru a nu lasa astfel de urme cu posibilitatea de a fi descoperiti, aici intervine un element cheie: *SA INTELEGI CUM FUNCTIONEAZA TEHNOLOGIA*.

La ce ma refer ? Ma refer la faptul ca este extrem de important sa intelegi cum functioneaza “acel server de baze de date, acel server de mail sau web” - atat din punctul de vedere al modului in care il configurezi, cat si din punctul de vedere al monitarizarii si jurnalizarea (log-uri) acestuia.

De asemenea este important sa stii cum sa functioneaza sistemul de operare **Windows** sau **Linux**. “Cum sunt creati userii ? Unde sunt stocate datele acestora ? Datele de logare ? Ce se intampla in momentul in care te loghezi pe un astfel de sistem ? Unde se scriu acele log-uri ?” etc.

Hacking (profesionist, etic si sigur) nu este pentru toate lumea si de aceea trebuie sa fi foarte bine pregatit pentru ca in anumite situatii libertatea ta poate fi pusa in joc.

Inca un lucru foarte important pe care vreau sa-l retii este faptul ca nimeni, **NIMENI**, nu face “Hacking” **de la el de acasa**. Este foarte important sa-ti ascunzi urmele pe cat de mult poti. Asta inseamna sa schimbi locatia in care te aflii, sa folosesti servicii VPN (despre care vom discuta in capitolul 10) si/sau Tor pentru criptarea si anonimizarea traficului.

Cum stergem urmele dintr-un sistem ?

Acum hai sa vedem cateva metode prin care iti poti acoperii urmele lasate odata ce ai intrat intr-un sistem (retea, server, laptop etc.)

a) **Stergerea Log-urilor din diferite aplicatii (web, mail etc.)**
b) **Stergerea Log-urilor userilor**
c) **Stergerea logurilor din diferite sisteme de monitorizare**

Fiecare sistem are diferite moduri de monitorizare a acestuia cu scopul de a face debuging sau troubleshooting in cazul aparitiei unei probleme

Pentru a face toate acestea nu este necesar, ca un Hacker, sa merga pas cu pas, din fisier in fisier sa caute si sa stearga ultimele jurnalizari (log-uri). Ci poate folosi diferite scripturi existente (in Internet) ale altor persoane cu care isi poate curata urmele.

Iata **aici** un exemplu de program pentru **Windows** (de asemenea se mai poate folosi si EventViewer).

Iar pentru **Linux** se pot da urmatoarele comenzi:

#**rm ./bash_history** - pentru stergerea comenzilor date de catre utilizatorul curent

#**vim /var/log/messages** - loc in care se pot sterge logurile

Sau in orice alt fisier din **/var/log**, depinde cu ce aplicatie s-a incercat exploatarea. Mai exista un alt mod prin care putem sterge logurile folosind Meterpreter (o aplicatie destinata PenTesterilor, despre care vorbesc si iti arat cum sa faci in **cursul de CyberSecuritate**).

6) (Pentru cei etici) Raportarea

Un alt pas foarte important, mai ales in procesul de Ethical Hacking este #6, **Reporting (Rapotarea)**, pasul in care Hackerul genereaza un raport asupra vulnerabilitatilor gasite (si exploatate), modurile prin care acestea pot fi remediate si alte informatii care sa duca la solutionarea si securizarea sistemului.

Acestia au fost cei 5 pasi (6 pentru Ethical Hackers) care constituie **procesul de Hacking.** In urmatorul capitol vom incepe discutia despre cele 3 elementele fundamentale care stau la baza securitatii cibernetice.

II. Instalarea si folosirea OS-ului Kali Linux

Daca esti curios sa afli cum se fac atacurile cibernetice, atunci ai ajuns la capitolul potrivit pentru ca acum iti voi arata un tutorial de instalare a Kali Linux (distributia de Linux folosita de Hackeri).

Ce este Kali Linux ?

Kali Linux este **distributia de Linux** (cea mai) folosita de catre **Hackeri** si **Pentesteri** profesionisti datorita numarului de programe, preinstalate, existente pe aceasta. In Kali Linux poti gasi extrem de multe programe axate pe partea de securitate si pe partea de testare a vulnerabilitatii sistemului. Indiferent ca vorbim de scanari, atacuri DoS, atacuri Web sau orice alt tip de atac, Kali este alegerea perfecta pentru orice doreste sa invete securitate. In Figura 2.1 de mai jos poti sa vezi logo-ul oficial al acestei distributii de Linux. Denumirea de Kali vine de la zeul razboiului din mitologia hindusa.

Figura 2.1

Desi, la inceput, poata parea putin greu de utilizat asta nu trebuie sa te descurajeze din a persevera si din a invata constant lucruri noi. De ce spun ca e greu de utilizat ? Pai in primul rand este vorba de Linux, iar daca nu ai mai interactionat cu Linux pana acum (din Terminal) s-ar putea sa ti se para destul de dificil, la inceput.

In al 2-lea rand este vorba de numarul mare de programe de PentTesting existente pe Kali. Acestea sunt dificil de folosit (mai ales la inceput), daca nu stii care este scopul lor (practic ce face tehnologia din spatele acelui tool) si daca nu stii sintaxa acestuia (dar aceasta se poate invata - la fel ca si celelalte).

Pasii de instalare Kali Linux in Masina Virtuala

Cand vine vorba de instalarea oricari distributii de Linux (deci si Kali) avem 2 optiuni:

- **Instalare Dual-Boot**
  - Linux, respectiv Windows se afla instalat pe partitii diferite
  - Cele 2 OS-uri ruleaza pe rand
  - Necesita reboot-ul laptopului/desktop-ului pentru a alege OS-ul dorit
- **Instalare in Masina Virtuala**
  - Linuxul vine instalat intr-o aplicatie (Virtual Box) si poate fi folosit in acelasi timp cu Windows
  - Nu necesita reboot, iar cele 2 OS-uri pot fi utilizate simultan
  - Consuma mai multe resurse (CPU & RAM) pentru ca acestea trebuie alocate catre 2 OS-uri in acelasi timp

Personal prefer a 2-a metoda pentru ca este mult mai simpla si rapida. In plus iti spun din proprie experienta faptul ca daca folosesti prima varianta foarte des vei omite sa intrii in Linux si vei spune “Lasa, alta data. Acum nu am chef sa dau restart”. Dar cu varianta a 2-a nu prea ai scuze :D

Pentru a instala Kali Linux, avem nevoie sa trecem prin cativa pasi. In primul rand avem nevoie de programul **VirtualBox** (sau un alt program de virtualizare - ex: VMware Workstation) si de imaginea **OS-ului Kali Linux** asa cum poti vedea in Figura 2.2.

Iti recomand sa selectezi **versiunea pe 64 de biti**, iar downloadarea sa o faci folosind Torrent pentru ca va fi mult mai rapida.

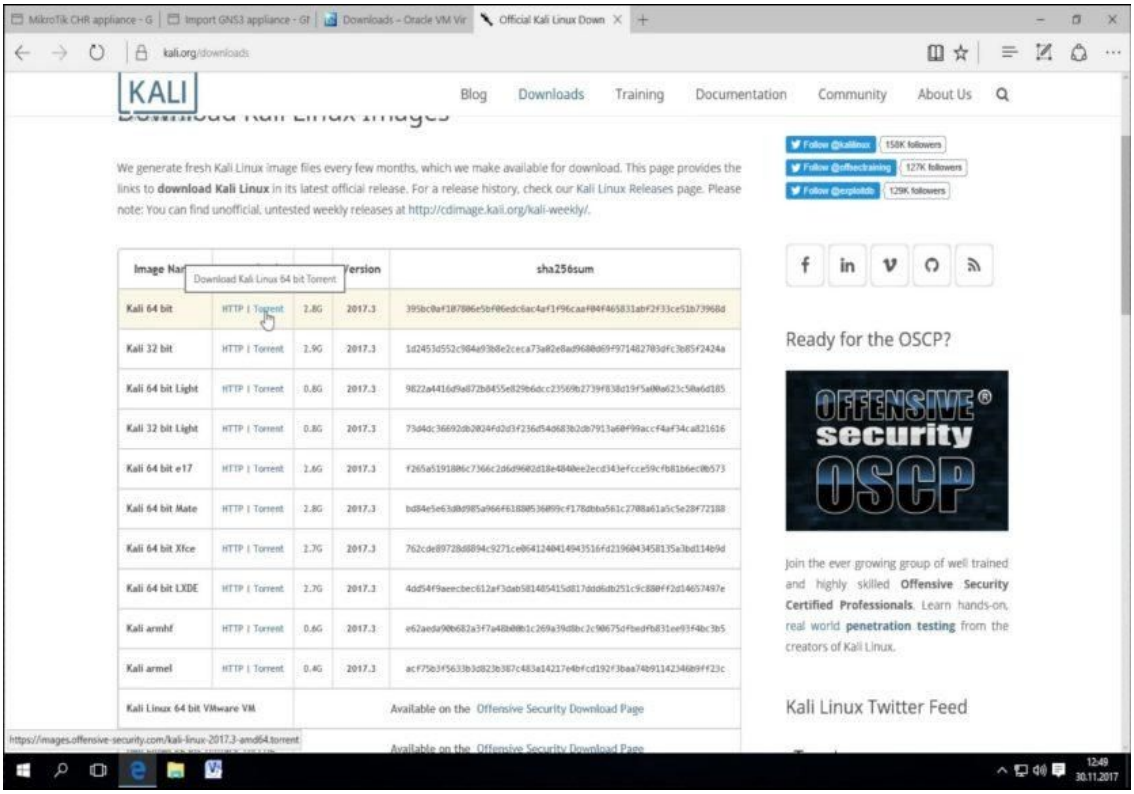

Figura 2.2

Urmatorul pas, dupa descarcarea imaginii OS-ului Kali Linux si al programului de virtualizare **Virtualbox** revine **procesului de instalare:**

1. Crearea unei masini virtuale - vezi **AICI** cum poti face asta
2. Inceperea procesului de instalare - dupa cum poti vedea in figurile 3.3 si 3.4.

Procesul este unul simplu, iar cu ajutorul **acestui tutorial** sunt convins ca vei putea sa duci la capatat toata instalarea si sa incepi sa te joci cu Kali ;)

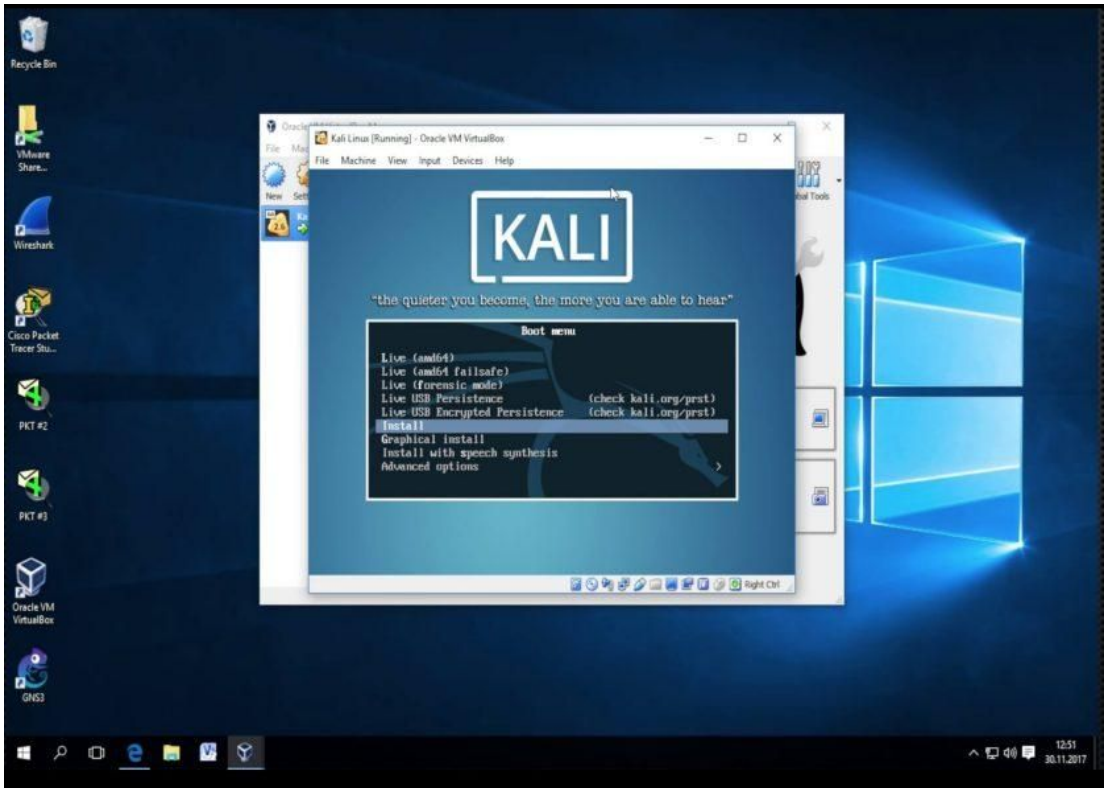

Figura 2.3

Daca vrei sa testezi Kali fara sa-l instalezi atunci poti opta pentru optiunea **Live**. Singura problema este ca de fiecare data cand vei porni masina virtuala ti se vor sterge setarile/munca pe care ai depus-o pana in acel moment. Daca scrii un script si esti in modul Live, acesta la reboot va fi sters, nu va fi salvat !

Deci si aici iti recomand sa mergi pe instalarea clasica pentru ca toate datele tale sa fie salvate pe disk.

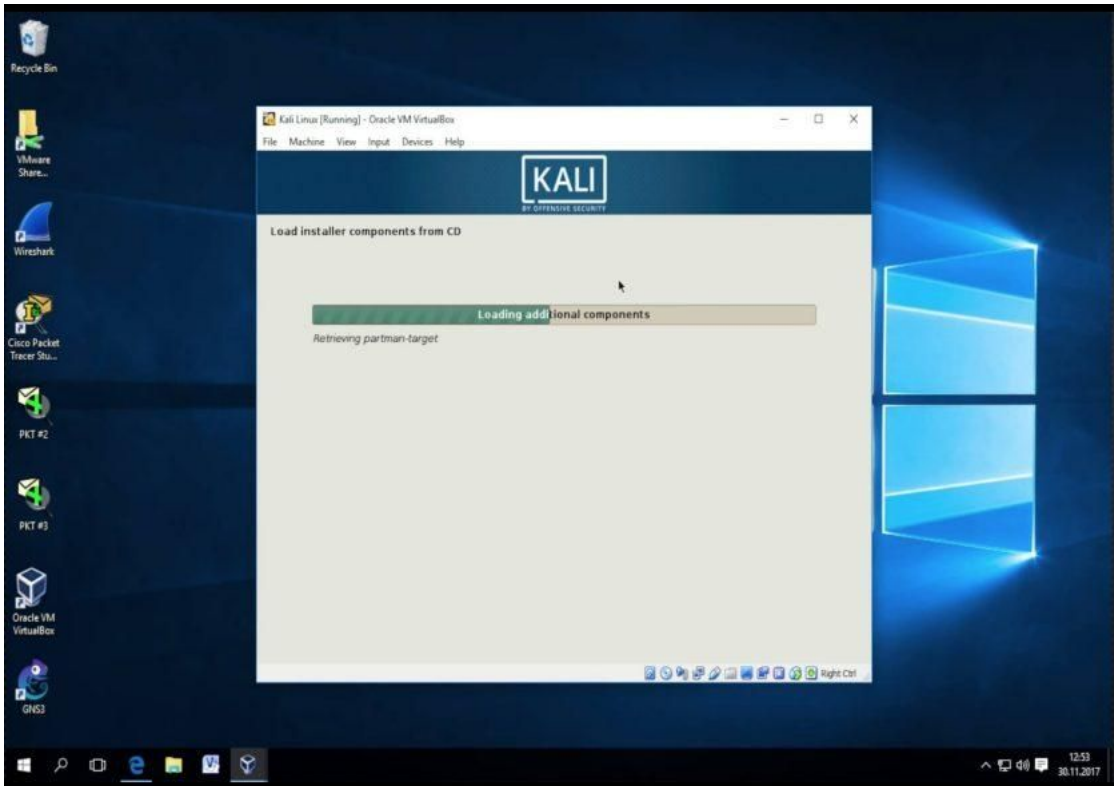

Figura 2.4

In continuarea procesului de instalare (Figura 2.4) in mare parte trebuie sa mergi **next -> next -> finish**, iar apoi sa astepti putin pana cand totul este gata. Daca nu te descurci la un moment dat (intervin anumite setari pe care nu le intelegi sau iti apare o eroare) te incurajez MAXIM sa faci **research pe Google**.

In ultimii ani am constat ca un skill, o abilitate, tot mai necesara in ziua de astazi este cea de **a cauta pe Google**. Probabil ca te amuza ceea ce spun eu aici, dar vreau sa stii ca vorbesc cat se poate de serios. Iti spun din proprie experienta ca acesta abilitate m-a scos de foarte multe ori din incurcatura, indiferent de situatia in care m-am aflat (construirea site-ului, terminarea proiectelor din timpul facultatii, documentarea si nu in ultimul rand gasirea forumurilor cu subiecte de interes pentru mine).

Deci daca iti apare o eroare la instalare sau in orice alta situatie. *Don't panic. Think for yourself. And search on Google :)*

Aaa.... si apropo, **userul default** pentru Kali Linux e **root** cu parola **toor**. Acum te-am scapat eu de o cautare ;)

Prezentare Distributie Kali Linux

Acum, dupa ce ai terminat cu instalarea si ai reusit sa pornesti si sa intri in Desktop, propun sa mergem mai departe si sa-ti prezint pe scurt Kali-ul astfel incat sa intelegi si sa identifci o parte din uneltele pe care le ai la dispozitie (in functie de obiectivul tau). Dupa cum poti sa vezi in Figura 2.5, ne aflam in starea default a Kali-ului, mai exact pe Desktop. In partea stanga ai o bara cu o parte din unelte, dar sus de tot (pe pozitia a 2-a) poti sa vezi terminalul (cel mai probabil cea mai importanta componenta pe care iti recomand sa o stapanesti cat mai bine ;).

Figura 2.5

Mergand in stanga sus, avem un meniu fooooarte interesant :D Meniul cu aplicatiile de PenTesting pe care le putem folosi (cu unele chiar ai experimentat din capitolele anterioare). Dupa cum poti sa vezi in Figura 2.6, avem de unde alege (ba chiar mai mult,

ele sunt puse in diferite categorii, iar aici intervenim noi - sa alegem cele mai eficiente programe pe interesul nostru).
Aceste aplicatii sunt defapt programe de “Hacking” care pot fi folosite atat cu intentii bune cat si mai putin bune. Totul depinde de tine acum sa le folosesti in scopuri cat mai bune (psss…. Ethical Hacking).

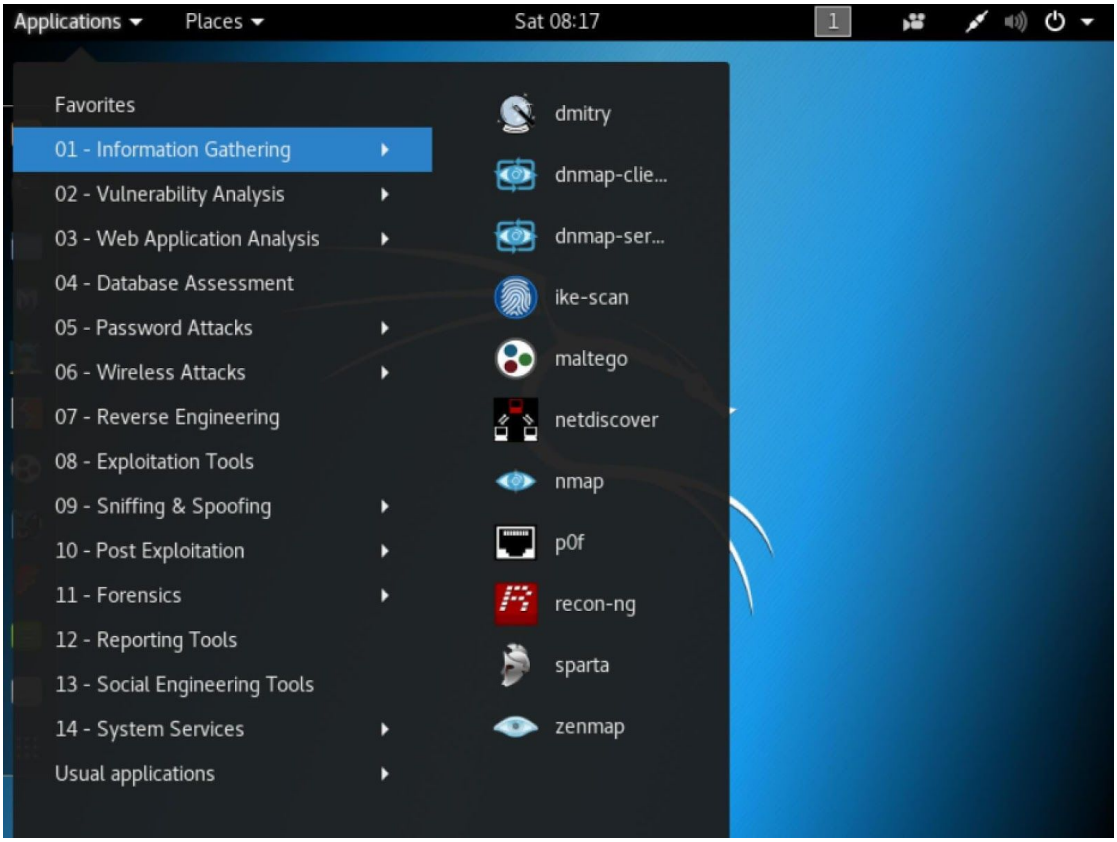

Figura 2.6

Si dupa cum poti sa vezi, chiar prima categorie se numeste “*Information Gathering*”, **exact** ca **primul pas** din *Procesul de Hacking* (despre care am vorbit mult mai in detaliu in capitolul 2). Aceste tool-uri pe care le vezi in Figura 2.6 ne ajuta sa obtinem mai multe informatii despre tinta noastre. Pe unele dintre ele chiar le-am folosit sau le-am mentionat (nmap, zenmap).
Un lucru pe care vreau sa-l retii este faptul ca, in momentul in care apesi pe unul dintre aceste programe (oricare ar fi ele) se pot intampla aceste 2 lucruri:

1. Se deschide programul cu interfata GUI
2. Se deschide un terminal care ruleaza programul si iti afiseaza informatii de tip “help” ale acestuia

In primul caz poate fi destul de intuitiv ce poti face cu el, iti vei da seama pe parcursul folosire (exemple de **programe GUI in Kali**: *Yersinia, Maltego, Burp suite, Wireshark* etc.). In al 2-lea caz s-ar putea sa nu fie atat de evident inca de la prima utilizare pentru ca, dupa cum spuneam si mai devreme, ti se va deschida un terminal cu un fel de meniu help/descriere a acelui tool. In ambele cazuri (mai ales in cazul 2) este important sa inveti acel program. Sa intelegi ce face ele cu adevar si “cu ce sa mananca”.
In Figura 2.7 de mai jos poti sa vezi la ce ma refer mai exact:

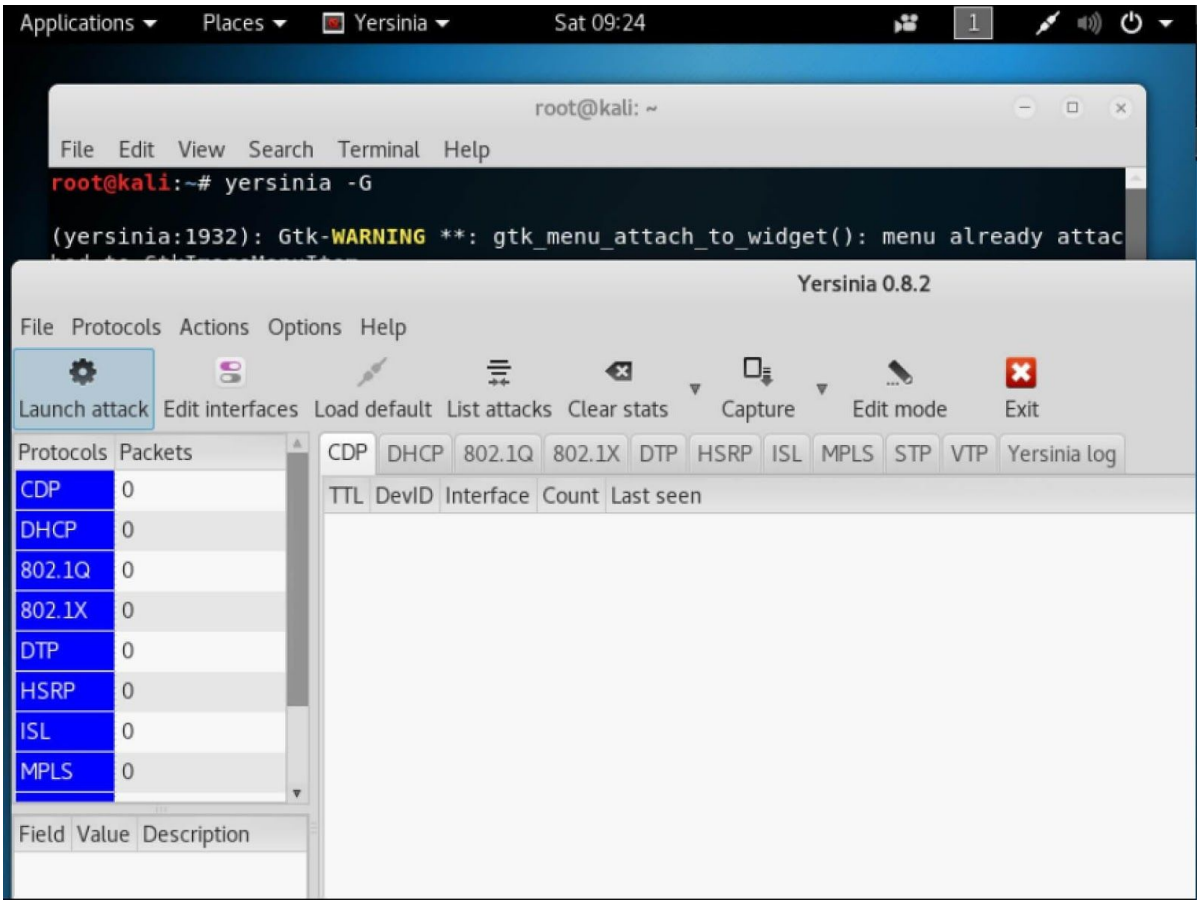

Figura 2.7

Yersinia este un tool grafic cu ajutorul caruia putem face foarte usor atacuri de tip MITM (mai ales daca in retea exista echipamente Cisco - Routere, Switch-uri nesecurizate). Yersinia are si o varianta in terminal care este mult mai puternica si customizabila. Pentru a porni versiunea GUI a Yersinia trebuie sa dam urmatoarea comanda:

#yersinia -G

De aici, te las pe tine sa experimentezi cu acest program :D Tot ce pot sa-ti spun este ca in partea stanga, vor aparea numarul de pachete capturate de tipul respectiv (statistica care iti da si un indiciu clar pe ce tip de atac sa te focusezi).
Mergand mai departe, in Figura 2.8 de mai jos poti sa vezi o alta categorie care contine diferite tool-uri (unele din le-am folosit - *ettercap, Wireshark, macchanger*) care au scopul de a asculta, respectiv capta traficul intr-un atac cibernetic de tip MITM (Man-In-The Middle):

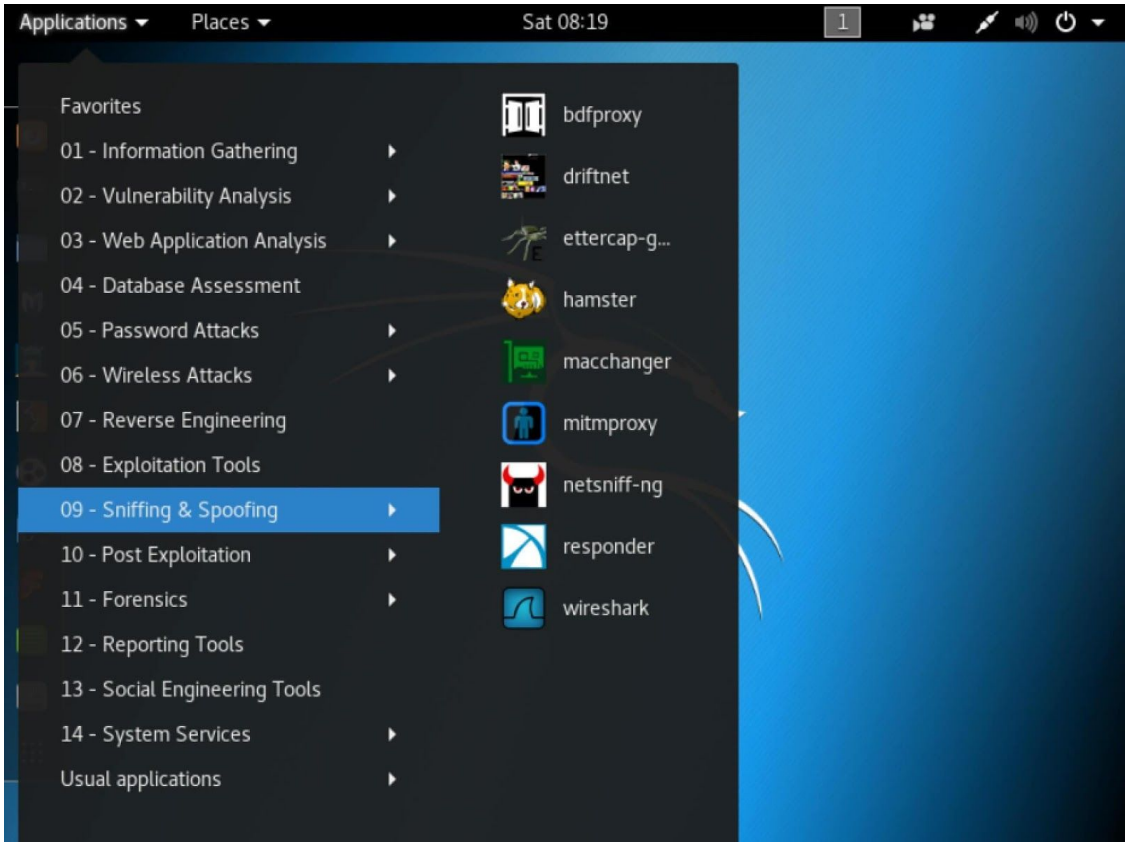

Figura 2.8

Acesta a fost doar o scurta introducere pe partea de Kali Linux. Daca vrei sa inveti mult mai multe elemente practice de PenTesting si CyberSecurity atunci te invit sa citesti cartea de ***Introducere in Securitate Cibernetica si Hacking.***

III. Tipuri de MALWARE si Atacuri Cibernetice

In acest capitol incepem discutia despre **tipurile de malware**-uri, iar mai tarziu vom discuta despre **Atacuri Cibernetice**. Pentru inceput vom discuta despre *Virusi, Trojeni, Viermi, Ransomware* si alte tipuri de programe care au fost concepute cu rea intentie. Dar inainte de toate acestea sa raspundem la urmatoarea intrebare:

1) Ce este un Malware ?

Un **malware** (aka. **malicious software**) este un *program conceput cu intentii rele* (aka. software malitios care vrea sa ne fure, distruga sau corupa datele stocate pe dispozitivele noastre).

Figura 3.1

Foarte multa lume foloseste **termenul generic de virus**, care nu este neaparat corect pentru ca pot exista mai multe *tipuri de programe periculoase.*

Iata mai jos (doar) o partea dintre acestea:
1) *Virusi*
2) *Troieni*
3) *Viermi*
4) *Ransomware*
5) *Spyware*
6) *Adware*
7) si multe, multe altele (Rootkit, time bombs, backdoor, etc.)

Iata in imaginea de mai jos extrasa de pe Wikipedia, proportia (in 2011) a malware-ului din Internet. De atunci si pana acum multe lucruri sau schimbat, dar este interesant ca avem o astfel de ierarhie cu cele mai intalnite tipuri de malware-uri.

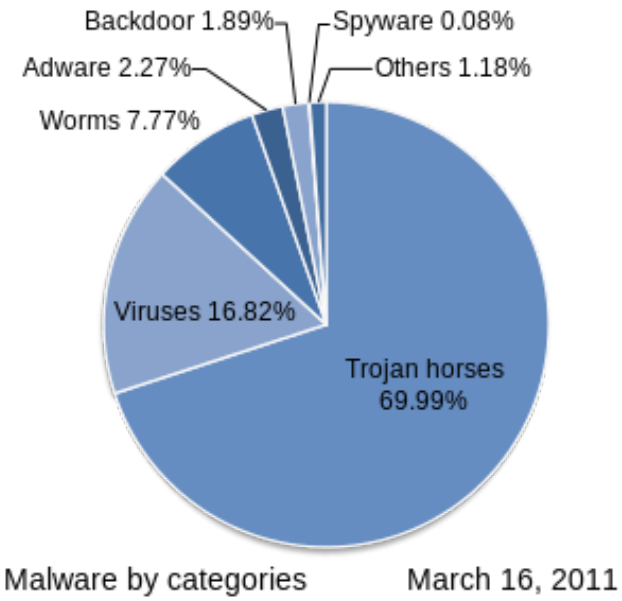

Figura 3.2

Si acum sa luam cateva dintre aceste malware-uri si sa discutam mai in detaliu despre ele:

1) Virusi

Un virus este un program cu care totii suntem obisnuiti. Fie ca am avut calculatorul infectat cu un virus sau ca am auzit/vazut la altcineva, stim ca acesti virus pot fi cu adevarat periculosi pentru noi (si mai ales pentru datele noastre stocate pe calculator - elementul cel mai important pentru noi).
Programatorii de virusi profita de vulnerabilitatile existente pe diferite sisteme de operare (in mod special Windows) si scriu software care sa profite de acestea (si de utilizatorii acestor dispozitive).

2) Trojeni

Un troian este un tip de program conceput sa para in folosul celui care il foloseste, dar in spate exista un cod malitios care are cu totul alte intentii. Aceste tipuri de programe se intalnesc cel mai des in Internet (dupa cum ai putut vedea si in imaginea de mai sus) si sunt folosite pentru ca sunt usor de mascat in fata unui utilizator neexperimentat. Astfel in momentul (primei) rularii programului, troianul este instalat si se va ascunde, facandu-si treaba "in liniste". Termenul de troian vine de la povestea calului troian din mitologia greaca, expusa in filmul Troia.

3) Viermi

Un vierme (**worm**) este o *forma de malware* care odata ce infecteaza un dispozitiv (PC, laptop, server etc.) va face tot posibilul sa se extinda si sa infecteze altele din retea. Astfel un worm reuseste sa **incetineasca** device-urile conectate la retea (prin consumul de resurse CPU si RAM) si chiar si reteaua pentru ca, calculatoarele infectate vor genera un consum anormal de trafic.

4) Ransomware

Un tip de malware tot mai popular in ultima perioada este ransomware-ul, a carui **scop este sa cripteze** hard disk-ul (sau SSD-ul) victime si sa ceara o **rascumparare** in bani acesteia pentru cheia de decriptare.

5) Adware

Sunt programe care odata instalate pe un dispozitiv (sau in browser) va incepe sa afiseze reclame (enervante).

6) Spyware

Sunt programe concepute cu scopul de a extrage anumite date de la utilizatori. Acestea nu sunt gandite sa ingreuneze (prin consumul de resurse) sau sa afecteze in vreun fel victima, ci pur si simplu sa extraga date si sa le trimita catre "serverele mama" (cele care au initiat "spionajul").

In primul rand trebuie sa fii constient de existenta unor astfel de programe dupa care trebuie sa iei masuri de protectie/prevenire impotriva lor.
In aceasta situatie programele de tip **anti-virus** sunt foarte bine venite pentru ca ele contin baze de date foarte mari (numite **semnaturi**) care verifica fiecare program/fisier aflat pe sistemul tau de operare (fie el Windows, Linux sau Mac).

Acum poate stii si tu ca pe Windows exista cel mai mare numar de programe malware (virusi, trojeni, ransomware etc). De ce ? Pentru ca Windows este cel mai utilizat sistem de operare la nivel mondial, iar Hackerii au ce sa “fure”. De aceea focusul principal al atacatorilor si al companiilor care se ocupa cu Securitatea Cibernetica este pe Windows.

Mac-ul si Linux-ul nu sunt nici ele ferite de malware, doar ca numarul lor nu este atat de mare. Acestea au fost concepute si cu un grad de securitate mai mare in minte si opereaza complet diferit fata de Windows.

2) Exemple de Atacuri Cibernetice

In acesta sectiune a capitolului 5, vom vorbi despre **Atacurile Cibernetice** si vom vedea cateva exemple de **Atacuri Cibernetice** (si *metode de Hacking*) din Internet. Aceste metode de hacking sunt foarte des intalnite, iar fiecare dintre ele serveste un scop anume.

Ce este un Atac Cibernetic ?

Un **atac cibernetic** este un mijloc prin care o persoana (cu rele intentii) **profita de vulnerabilitatile** existente pe un anumit sistem (server, calculator, echipament de retea, aplicatie etc.). Iata in lista de mai jos cateva atacuri foarte des intalnite in Internet:

1) **MITM** - **M**an **i**n **t**he **M**iddle
2) **DoS** - **D**enial **o**f **S**ervice
3) **DDoS** - **D**istributed **D**enial **o**f **S**ervice, link atacuri: http://www.digitalattackmap.com/
4) **SQLi** - SQL injection
5) **XSS** - Cross-Site Scripting

IV. Scanarea Retelei si a Serverelor

Sunt sigur ca ai auzit de multe ori de conceptul de scanare a unei retele (sau a unui server). Well, in acest capitol vom discuta mai in detaliu despre tot acest concept si despre cum poti **scana retele si server** din doar cateva comenzi din (Kali) Linux sau Windows.

Ce inseamna sa "Scanez o Retea de Calculatoare" ?

Inainte sa trecem la a vedea cum pot sa scanez o retea, vreau sa-ti explic ce inseamna aceasta scanare. La ce ma refer pana la urma cand spun am scanat reteaua X.Y.Z.A ? Ma refer la faptul ca am folosit un anumit program (in acest caz **Nmap**) cu ajutorul caruia am aflat care sunt dispozitivele conectate la retea in momentul de fata.

Nu doar ca am aflat care sunt acest dispozitive (afland adresa lor **IP** si **MAC**-ul) ci putem afla si alte informatii precum:

- Tipul dispozitivului
- Sistemul de operare si versiunea folosita
- Porturile deschise
- Aplicatile care ruleaza pe acele porturi
- etc.

Odata avand aceste informatii ne putem folosi de ele pentru a intelege mai bine cum este structurata reteaua, pentru a scana si ulterior testa reteaua si serverele de vulnerabilitati (intr-un mod etic), pentru a ne asigura ca toate dispozitivele sunt up & running. Acestea , bineinteles, sunt doar cateva motive pentru care scanarea retelei si a device-urilor din ea are sens.

Cum Scanez o Retea ?

Scanarea retelei (si a componentelor ei) se face foarte usor, folosind Nmap. Nmap vine de la Network Mapper si ne ajuta sa "mapam" reteaua intr-un output (din terminal) destul de usor de inteles.

Programul pe care il folosesc in exemplul de mai jos (pentru scanarea retelei) se numeste **Nmap**. Nmap (pe Windows, programul cu interfata grafica se numeste **Zenmap**) este un tool gratuit extrem de folosit de hackeri si de ethical hackeri.
Cu ajutorul lui putem descoperi dispozitivele conectate la retea, porturile deschise de pe acestea si chiar si Sistemul lor de Operare.

Iata cateva exemple de comenzi pe care le putem da cu Nmap pentru a atinge diferite scopuri:

1) Scanare la nivel de retea cu Nmap

`# nmap -sP 192.168.1.0/24` -- scanare pe baza de ICMP (ping), afisand numarul de dispozitive din retea (aka **PING SCAN**)

`# nmap -sS 192.168.1.0/24` -- scanarea intregii retele cu scopul de a gasi porturile deschise (TCP, folosind SYN) de pe fiecare dispozitiv (aka. **PORT SCAN**)

In Figura 4.1 de mai jos poti vedea in detaliu informatile despre o parte din echipamentele conectate la retea (mai exact Routerul cu IP-ul 192.168.1.1 si un alt dispozitiv cu IP-ul 192.168.1.3). Pe langa faptul ca a descoperit ca aceste dispozitive sunt conectate la retea, Nmap a mai descoperit si porturile deschise pe aceste echipamente.

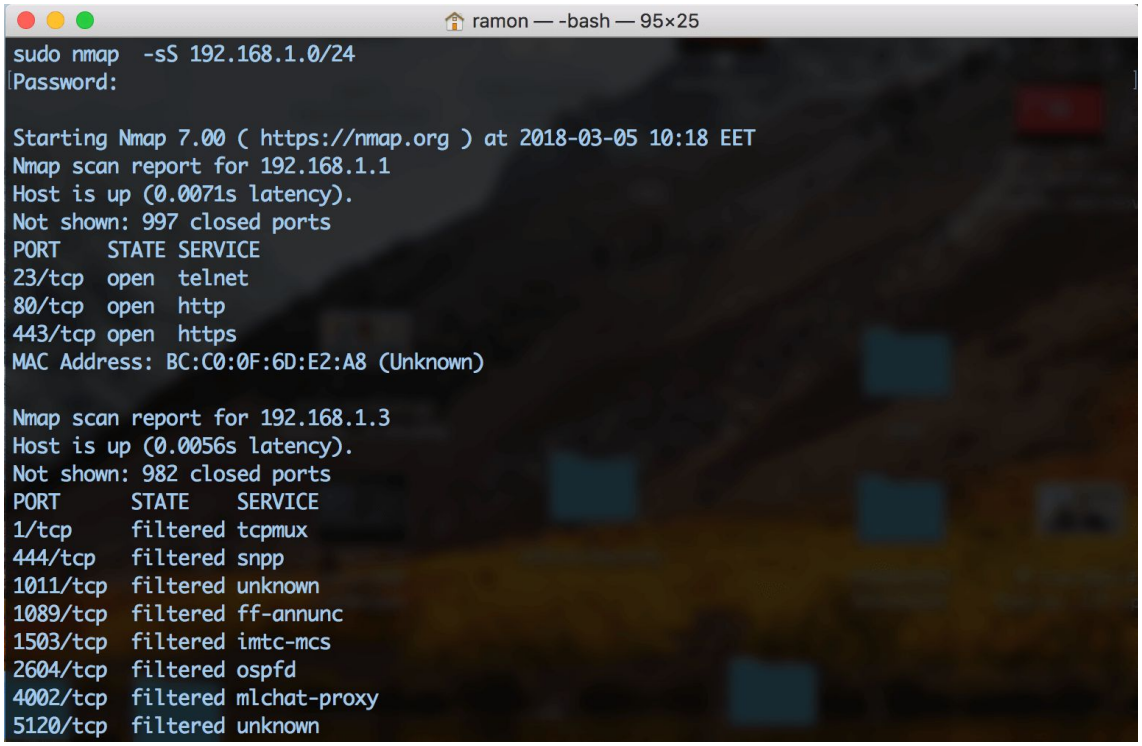

Figura 4.1

Dupa cum poti sa vezi Routerul (192.168.1.1) are deschise 3 servicii destul de importante care pot fi vulnerabile si exploatate. Spre exemplu, portul 23 reprezinta Telnet ceea ce inseamna ca ne putem conecta de la distanta la acesta putand chiar sa avem acces la CLI (linia de comanda).

De asemenea, portul 80, respectiv 443, care identifica traficul Web sunt deschise (deci ne putem conecta prin browser si putem incerca sa obtinem acces pe acest Router). Acum sper ca ai inteles rolul si puterea scanarii ;)

2) Scanare folosind Nmap la nivel de dispozitiv (server, laptop, telefon etc.)

 # nmap -A *192.168.1.1* -- scaneaza un singur device pentru *obtinerea serviciilor* (porturilor) si a sistemului de operare (aka. **OS SCAN**)

 # nmap -sT 192.168.1.254 -- scaneaza folosind pachete TCP

 # nmap -sU 192.168.1.1 -- scaneaza folosind pachete UDP

In Figura 4.2 poti vedea rezultatul primei comenzi la care am mai adaugat si **-sS** pentru scanarea TCP a porturilor:

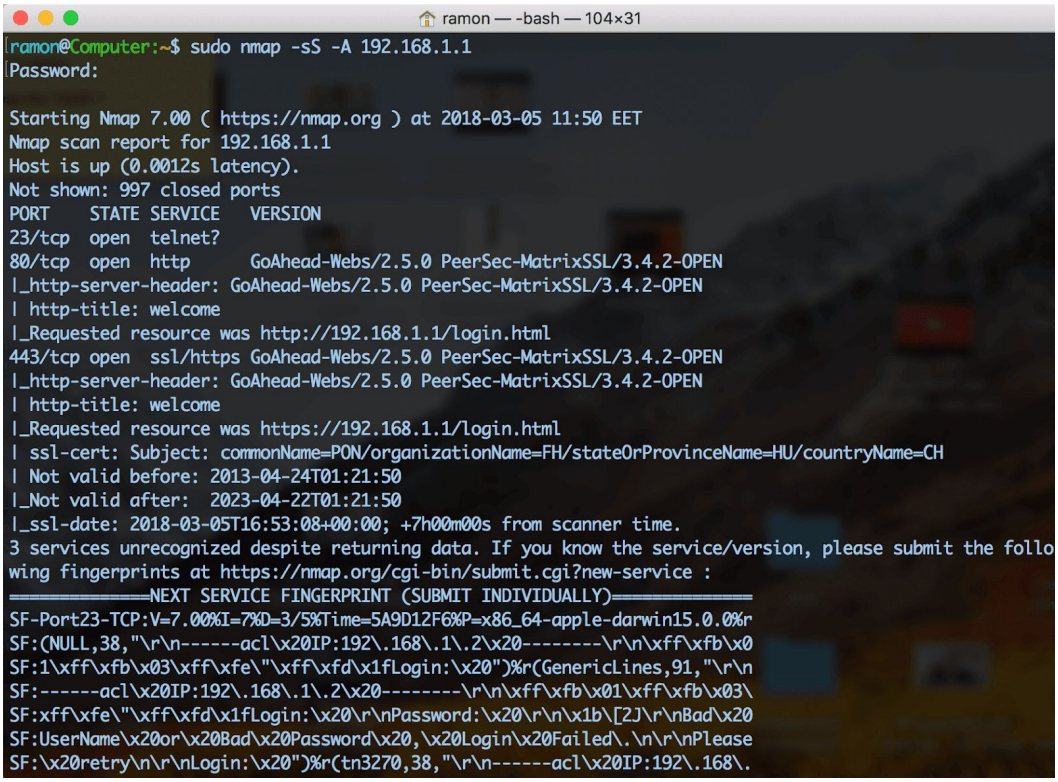

Figura 4.2

Daca vrei sa obtii (mai multe) informatii in timp real legate de scanare, atunci iti recomand sa adaugi **-v** la orice tip de comanda Nmap doresti. De asemenea, mai poti apasa si pe “Space” pentru a obtine date legate de progresul scanarii (vei vedea X% finalizat si timpul estimat).

Alte exemple de scanare folosind Nmap:

 # nmap -F 192.168.1.0/24 -- scaneaza fiecare device din retea pentru top 100 cele mai folosite porturi

 # nmap -sS -p 80,443,23,22,25 192.168.1.1 -- scaneaza device-ul pentru porturile date cu argumentul -p folosind pachete TCP SYN

 # nmap -F -oN *rezultat_scanare.txt* 192.168.1.0/24 -- scaneaza reteaua rapid si stocheaza rezultatul in fisierul rezultat_scanare.txt (foarte util in cazul unui script - Python)

3) Hping3

Un alt tool foarte util pe care il mai putem folosi pe langa Nmap este **hping3**. **Hping3** este un **generator de trafic**, similar cu ping, dar cu mult mai multe functionalitati. Este capabil sa trimita (pe langa trafic ICMP - ping) pachete de tipul TCP, UDP sau RAW-IP customizate (cu orice specificatie ii dam noi legat de aceste protocoale).

Spre exemplu putem trimite un packet TCP ACK sau FIN pentru a vedea cum reactioneaza serverul sau firewall-ul pe care dorim sa-l testam. Mai jos gasesti o lista cu lucrurile pe care le poti face cu acest tool.

Practic, hping3 ne ajuta sa facem face urmatoarele lucruri:

- Firewall testing
- Advanced port scanning
- Network testing, using different protocols, TOS, fragmentation
- Manual path MTU discovery
- Advanced traceroute, under all the supported protocols
- Remote OS fingerprinting
- Remote uptime guessing
- TCP/IP stacks auditing

Iata cateva **exemple de folosire a hping3**:

 # hping3 -h -- pentru a afla mai multe despre argumentele disponibile

 # hping3 -1 *IP_VICTIMA* -- se trimite un ping (ICMP) normal (Figura 3.3)

 # hping3 --traceroute -V -1 *IP_VICTIMA* -- se trimite un singur pachet de tip traceroute pentru a vedea pe unde merge acesta

 # hping3 -V -S -p 80 *IP_VICTIMA* -- se trimit pachete de tip TCP SYN pe portul 80 pentru a vedea daca raspunde aplicatia

In Figura 4.3 de mai jos poti am facut cateva teste si poti vedea o parte exemplele enuntate mai sus:

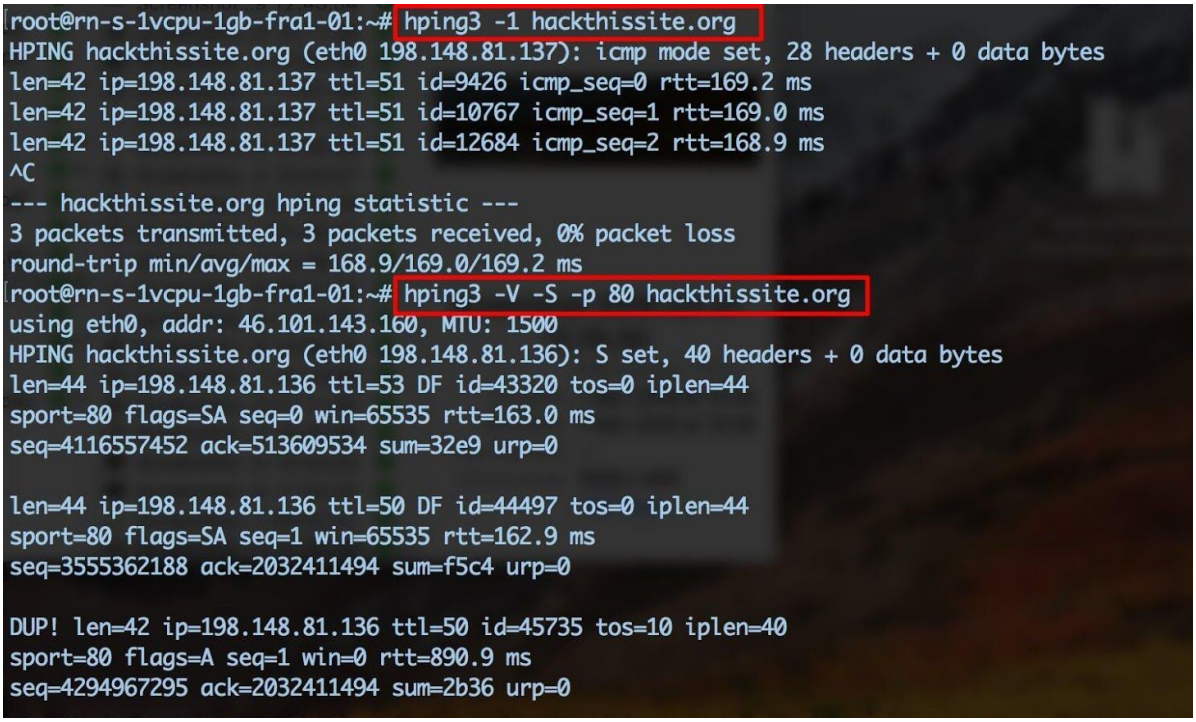

Figura 4.3

 # hping3 -c 1 -V -p 80 -s 5050 -A *IP_VICTIMA*

-- acest tip de scanare trimite un singur pachet TCP de tip ACK (-A) si ne ajuta sa ne dam seama daca un device este up in retea in momentul in care nu raspunde la ping (acesta fiind blocat de catre un firewall).

Argumentele comenzii (Figura 4.4) reprezinta:

- **-c 1**: se trimite un singur pachet
- **-V**: verbose (afiseaza in timp real rezultatul scanarii)
- **-p 80**: portul destinatie este 80 (HTTP)
- **-s 5050**: portul sursa este 5050 (poate fi orice altceva)
- **-A**: se trimit pachete de tip TCP ACK

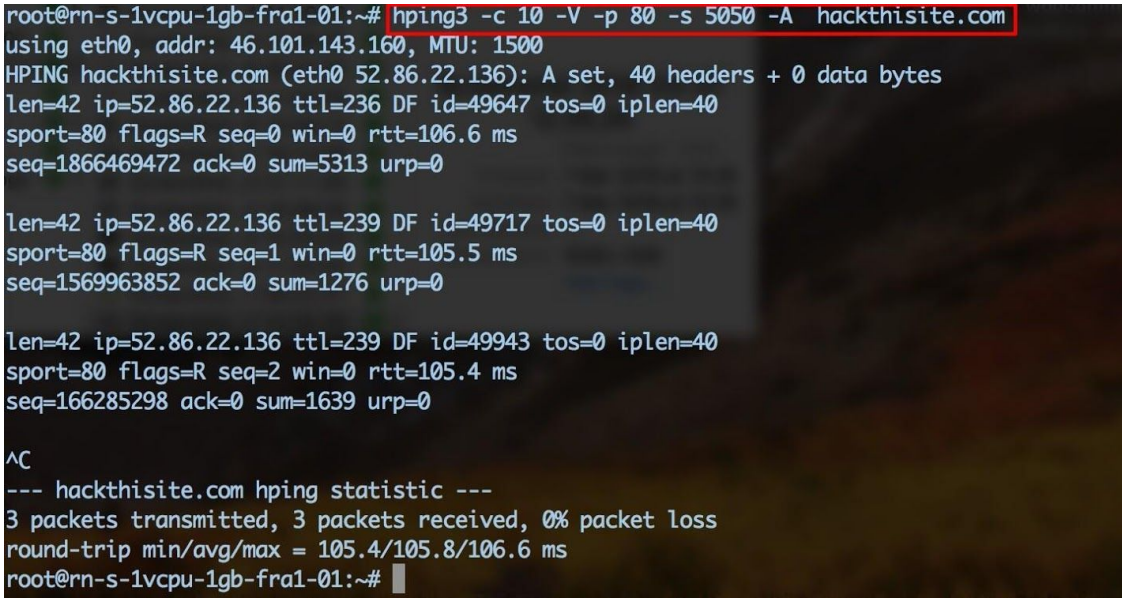

Figura 4.4

Daca dorim sa **ne acoperim urmele** (sa fim **anonimi**, sa nu se stie de unde provine scanarea/atacul) putem adauga argumentul **--rand-source:**

 # hping3 -c 1 -V -p 80 -s 5050 -A --rand-source *IP_VICTIMA*

Hping3 iti ofera posibilitatea sa fii foarte specific cu ceea ce faci. Spre exemplu daca folosim scanari de tipul TCP la nivel de server atunci ne putem folosi de mesajele TCP-ului (precum SYN, ACK, FIN, RST, URG etc.).
Revin cu ce am spus si la inceputul acestei carti: ESTE FOARTE IMPORTANT SA INTELEGI CUM FUNCTIONEAZA TEHNOLOGIA. In acest caz ma refer in mod special la modelul OSI, la protocolul TCP, la porturi etc.

In **capitolul 7** vom discuta despre **Firewall**, cum functioneaza acestea si vei vedea ca acest tip de scanare se potriveste in cazul lor pentru ca firewall-urile by default blocheaza tot traficul din Internet.

Dar cu **pachete trimise customizat** (un TCP ACK neasteptat spre exemplu) putem sa surprindem firewall-ul sau aplicatia (ex: server web) si sa ne dea un raspuns cu ajutorul caruia putem afla mai multe informatii despre el (un server pe Windows raspunde intr-un mod diferit fata de unul pe Linux etc. - exemplu de remote OS fingerprinting).

In continuare iti sugerez sa te joci cu aceste programe Nmap si hping3 sa faci putin mai mult research astfel incat sa le intelegi utilitatea si modul lor de functionare.

Cu **hping3** mai putem face si atacuri de tip **DoS** (despre care am vorbit in capitolul 5), in care facem flood catre un anumit dispozitiv:

```
# hping3 -V -c 2000000 -d 100 -S -w 64 -p 443 -s 591 --flood --rand-source IP_VICTIMA
```

Iar argumentele acestei comenzi reprezinta:

- --flood: trimite pachetele cat mai repede posibil
- --rand-source: genereaza adrese IP sursa diferite pentru fiecare pachet
- -V: cat mai explicit (ofera informatii in timp real)
- -c --count: numarul total de pachete
- -d --data: marimea pachetelor
- -S --syn: pachete TCP de tip SYN
- -w --win: window size (default 64)
- -p --destport: portul destinatie
- -s --baseport: portul sursa (by default este aleatoriu)

Acestea au fost doar cateva exemple. Acum tu poti incepe sa te joci cu acest tool (iti recomand sa folosesti Kali Linux si sa incepi cu reteaua ta locala). In loc de IP_VICTIMA poti da IP-ul Routerului tau sau a unui telefon/laptop/server din reteaua ta.

PS: in cazul in care folosesti Linux (si nu Kali Linux) poti instala Nmap sau hping3 folosind comanda:

```
# sudo apt-get install nmap hping3
```

De ce vrem sa Scanam Reteaua ?

Pentru ca mai departe putem folosi aceste informatii pentru a ne decide focusul, cand vine vorba de penetration testing. Pe ce ne focusam ? Care este tinta cea mai vulnerabila din retea si ce aplicatii exista pe ea de care sa putem profita ? Pentru un Hacker (fie el etic sau nu) raspunsul la aceste intrebari este extrem de important.

De ce ? Pentru ca, daca nu este foarte bine informat despre retea si despre componentele existente in ea, Hacker isi va pierde timpul cu anumite parti care nu pot fi exploatate si va risca sa fie detectat.

Acest proces de scanare face parte din ciclul de Penetration Testing care este compus din 5 etape (figura 4.5) si despre care vorbesc mult mai pe larg in cartea de ***Introducere in Securitate Cibernetica si Hacking.***

Figura 4.5

V. Firewall

Incepem un nou capitol interesant in care vorbim despre Firewall-uri si cum functioneaza acestea. Un lucru vreau sa-ti spun: atat timp cat vei lucra cu orice inseamna servere, retele, site-uri web (mai ales din punct de vedere CyberSecurity), te rog sa fii sigur ca **te vei intalni cu Firewall-uri.**

Ce este un Firewall ?

La fel ca si un Router sau un Switch, un **firewall** este un echipament de retea care are scopul de a securiza (proteja) reteaua de potentialii atacatori (hackeri) din Internet.

Prin securizarea retelei, ma refer in mod special la **filtrarea pachetelor** (adresa IP sursa/destinatie, porturi, filtrare URL etc.) cu scopul de a nu permite accesul unei persoane neautorizate in retea.

By default, **firewall-ul blocheaza tot traficul** din extern in intern si ramane la latitudinea administratorului sa configureze politicile de acces in retea, necesare companiei. *Cum face asta ?* Vei afla in cele ce urmeaza ;)

Un lucru pe care vrea sa-l retii: **NU exista securitate perfecta**, iar un Firewall nu este suficient pentru a securiza reteaua, ci el este o componenta importanta care ajuta la securizarea accesului extern (adica din Internet in LAN).

Pe langa acest element de retea mai sunt multe alte componente care trebuiesc luate in calcul cand vine vorba de securitate. Dupa cum poti sa vezi in figura 8.1, firewall-ul conecteaza toate echipamentele la Internet (deci practic ia locul Routerului in aceasta situatie). Scopul lui, in acest caz, este sa le protejeze de potentialele atacuri din Internet, care la un moment dat pot incerca sa exploateze anumite vulnerabilitati pe sisteme. Firewall-ul functioneaza la **nivelul 4** (by default), dar poate fi configurat sa functioneze si la **nivelul 7** (din modelul OSI).

Ce inseamna asta ? Inseamna ca se poate “uita adanc” in pachet. Poate chiar sa **vada** ce URL incerci sa accesezi si sa ti-l blocheze, poate chiar sa scaneze programul pe care incerci sa-l descarci si sa-si dea seama daca este virusat sau nu.

Asta se aplica in cazul firewall-urilor perfomante si scumpe, adesea numite **UTM** (Unified Threat Management).

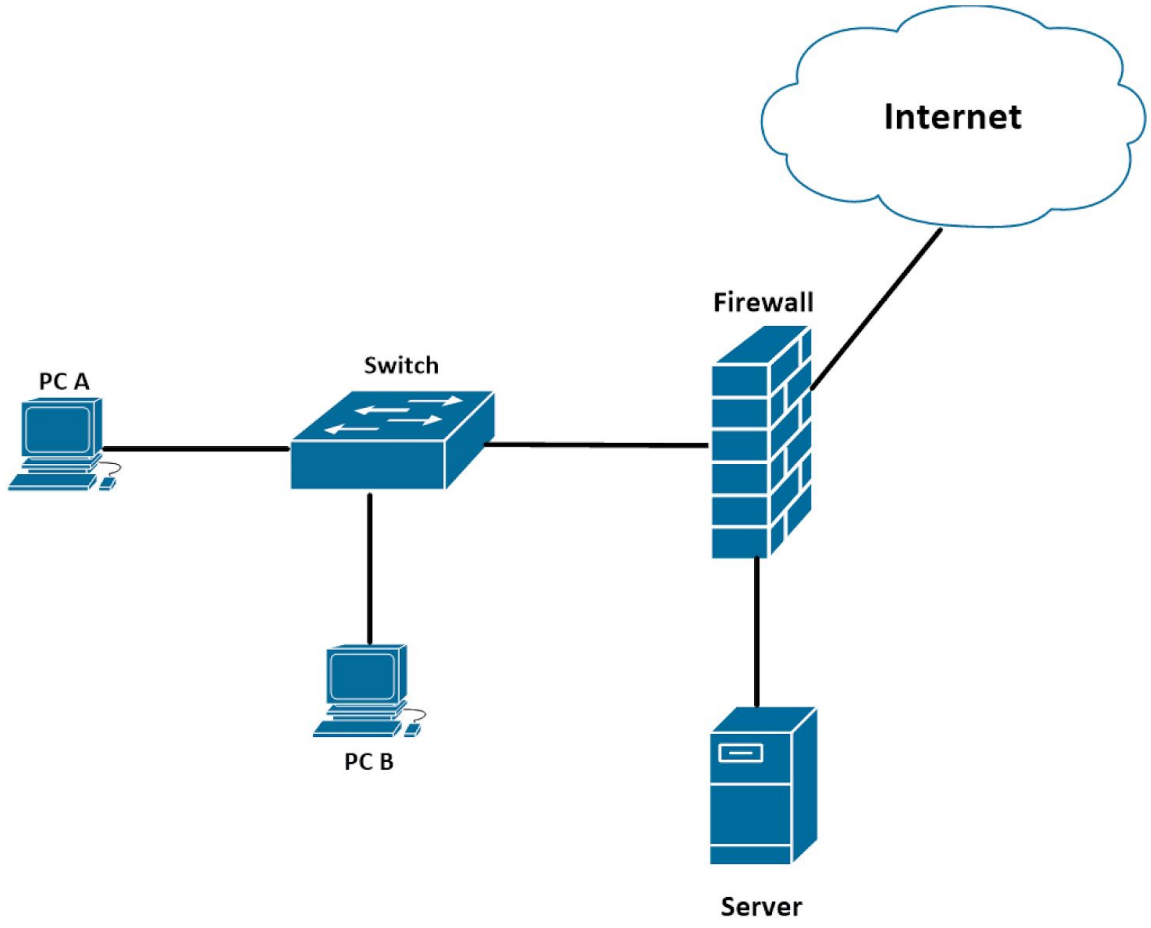

Figura 5.1

Cum functioneaza zonele de Securitate ale unui Firewall ?

Firewall-ul functioneaza pe baza de zone de securitate. Practic fiecare interfata a unui firewall reprezinta o zona de securitate. Noi putem creea mai multe astfel de zone si sa includem una sau mai multe interfate in ele. Aceste zone de securitate, de obicei, se impart in 3 categorii:

- **INSIDE** - zona care cuprinde reteaua interna (LAN)
- **OUTSIDE** - zona care cuprinde reteaua externa organizatiei (de obicei Internetul)
- **DMZ** - zona speciala care contine servere

Fiecarei zone in parte, firewall-ul ii va aloca un **nivel de securitate** (definit printr-un numar intre 0 - 100) care influenteaza comportamentul acestuia.

Spre exemplu: daca avem o zona de **INSIDE** cu un nivel de securitate de **100** si o zona de **OUTSIDE** cu un nivel de securitate de **0**, atunci **ORICE** trafic din INSIDE va fi **lasat** (de catre firewall) **sa treaca** in OUTSIDE (aka. Internet), dar traficul care vine din *OUTSIDE in INSIDE* va fi **oprit,** astfel fiind nevoie de reguli speciale (aka. invatare dinamica - statefull - a traficului) care trebuiesc a fi configurate pe firewall.

DMZ-ul (**D**emilitarized **Z**one) reprezinta o zona speciala in care sunt plasate echipamentele care necesita acces din Internet. In cea mai mare parte a timpului in aceasta categorie se incadreaza serverele (Web, Fisiere, VPN etc.). Acestea sunt resursele publice ale companiei la care, teoretic oricine are acces.

Aceste servere le izolam intr-o zona speciala (DMZ) astfel incat daca acestea ajung sa fie corupte, restul retelei sa nu sufere (Hackerul sa nu poata patrunde in LAN - reteaua interna companiei).

DMZ-ul are un **nivel de securitate mai mic** (30 dupa cum poti sa vezi si in figura 5.2) fata de cel al LAN-ului, dar mai mare fata de cel al Internetului. Astfel oricine din DMZ poate accesa Internetul si invers datorita unor regului din firewall despre vom vorbi imediat. In figura 5.2 poti sa vezi cum arata aceste zone de securitate.

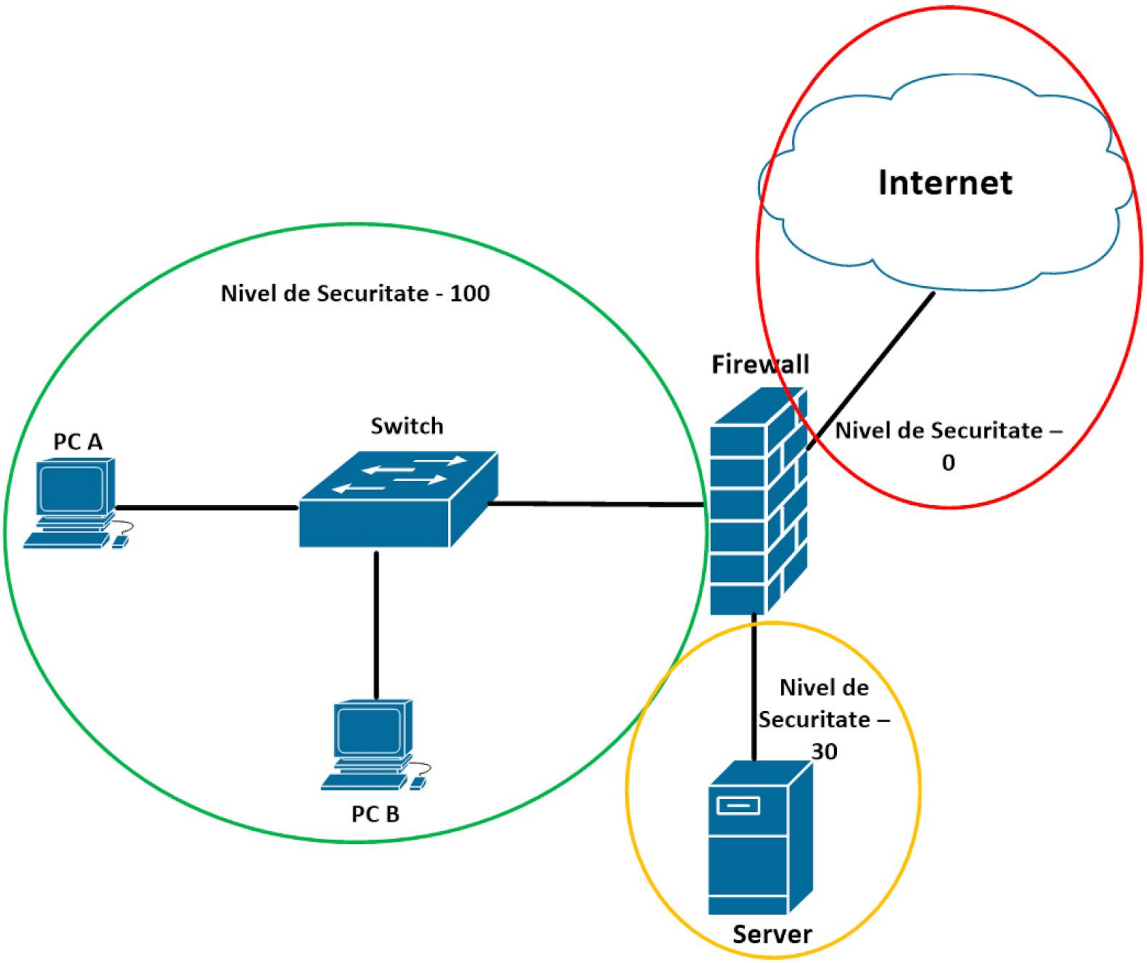

Figura 5.2

Poate la un moment dat te-ai intrebat: "*oare ce face firewall-ul asta 'in spate'* ? Adica cum functioneaza el defapt ?". Iar eu iti voi raspunde in cele ce urmeaza in sectiunea despre ACL-ul (practic acesta e "ingredientul" secret care ne protejeaza noua retelele... si DA, chiar si Routerul tau Wireless are un mini-firewall incorporat care foloseste ACL-uri :D)

ACL (Access Control List)

ACL-ul este componenta care sta la baza modului se functionare a unui firewall. Este componenta esentiala cand vine vorba de **protejarea** si **filtrarea traficului** (de obicei cel din Internet)

Un **ACL (Access Control List)** reprezinta un set de reguli cu scopul de a bloca sau permite accesul dintr-o retea la o anumita resursa. Aceste reguli sunt setate pe Routere sau pe Firewall-uri.

“ACL-urile stau la **baza *conceptului de securitate*** (limitare a accesului) intr-o sau dintr-o retea (ex: Din Internet in reteaua Interna - LAN sau invers)."

Gandeste-te la acest concept, ca la un Bodyguard care sta la intrarea unui club in care se organizeaza o petrecere privata. Acesta va avea o lista cu toti invitatii la acea petrecere.

Pe masura ce oamenii incearca sa intre in locatie, bodyguard-ul ii va verifica pe fiecare in parte; se va uita pe lista (**ACL**) si va decide pentru fiecare persoana daca are voie in club sau nu. Practic daca te afli pe lista vei fi lasat sa intri (permit) la petrecere, iar daca nu apari pe lista, nu vei avea acces (**deny**) inauntru.

Pentru inceput trebuie sa ne gandim ce tip de trafic (aka. reguli) vrem sa permitem in reteaua noastra, urmand ca apoi sa includem aceste reguli in ACL. Dupa cum vom vedea mai jos, aceste **reguli pot varia**: de la *permiterea unei retele intregi* sa acceseze o alta retea, la *permiterea sau respingerea accesului a unui singur PC* la un server pe un anumit port (ex: SSH - 22, Web - 80).

Dupa ce o astfel de lista de acces este creata si sunt adaugate reguli de ***permit*** *sau* ***deny,*** va fi pusa in functie (cum ? setand-o pe o interfata a Firewall-ului sau a Routerului pe o anumita directie - **IN** sau **OUT**). **IN**(side) reprezinta traficul care **intra** in Firewall, iar **OUT**(side) reprezinta traficul care **iese** din Firewall.

In mod normal, exista 2 tipuri principale de ACL-uri care pot fi setate (atat pe Firewall-uri cat si pe Routere sau servere Linux):

- **ACL Standard**
- **ACL Extended**

1) ACL Standard

Scopul ACL-urilor de tip Standard este sa faca filtrarea traficului dupa ***IP-ul sursa*** **!** Cel mai usor mod de a intelege este printr-un exemplu (precum cel din figura 5.3 si apropo, poti inlocui in mintea ta aceste Routere cu Firewall-uri, nu conteaza, ideea este sa intelegi conceptul):

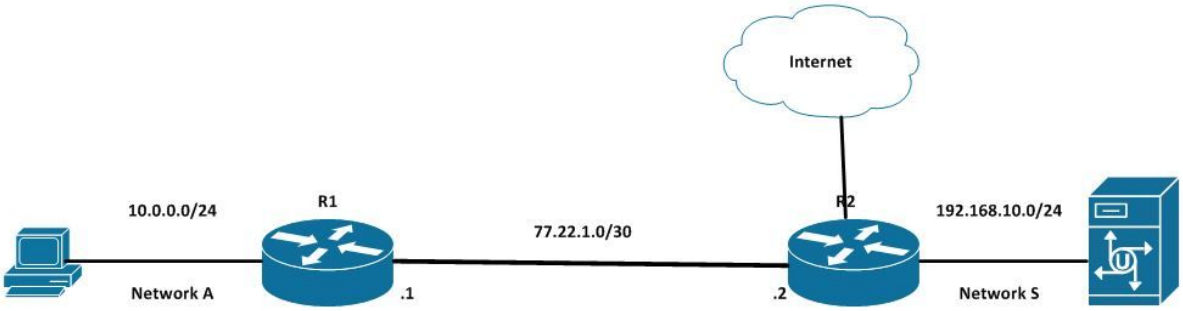

Figura 5.3

Sa spunem ca (din motive de securitate) PC-ului din reteaua A, cu IP-ul 10.0.0.8, nu ii vom da voie sa acceseze server-ul din reteaua S. Astfel tot ce trebuie sa facem este sa cream o lista de acces in care sa specificam acest lucru. Regulile acestei liste vor arata astfel:

#**deny** 10.0.0.8
#**permit any**

Aceasta regula va fi setata pe R2, pe **interfata cea mai apropiata** de server (in cazul acesta, cea direct conectata la server) in **directia OUT**. Am adaugat cea de a 2-a linie (permit any) deoarece, by default, la finalul fiecarui ACL apare o regula "implicita de deny" (#deny any). Noi dorim sa oprim traficul de la PC la server si sa **permitem in rest orice alt tip de trafic**.

ATENTIE: nu trebuie sa cunosti (sau sa retii) sintaxa pe care ti-o voi prezenta in acesta sectiune a cartii. Comenzile provin din linia de comanda pentru echipamentele Cisco, dar vreau tot ce vreau de la tine este **sa INTELEGI CONCEPTUL** si modul de functionare al acestor ACL-uri. *De ce ?* Pentru ca te vei intalni cu ele, foarte des. Fie ca le vei configura, fie ca doresti sa treci de ele este foarte important **sa stii cum functioneaza**.

2) ACL Extended

Scopul ACL-urilor de tip Extended este sa faca filtrarea traficului dupa:

- ***IP Sursa***
- ***IP Destinatie***
- ***Port Sursa***
- ***Port Destinatie***
- ***Protocol (IP, TCP, UDP etc.)***

Astfel, acest tip de liste ne ofera o flexibilitate mult mai mare cand vine vorba de control. Putem controla orice flux de trafic indiferent de sursa, destinatia si aplicatia folosita.

Scenariu practic

Pe scurt, practic ACL-urile au nevoie de urmatoarele informatii pentru a le putea pune in practica:

1. Lista impreuna cu regulile de **permit/deny** in functie de nevoi
2. Interfata pe care dorim sa aplicam aceste reguli
3. Directia traficului (IN/OUT) de pe interfata

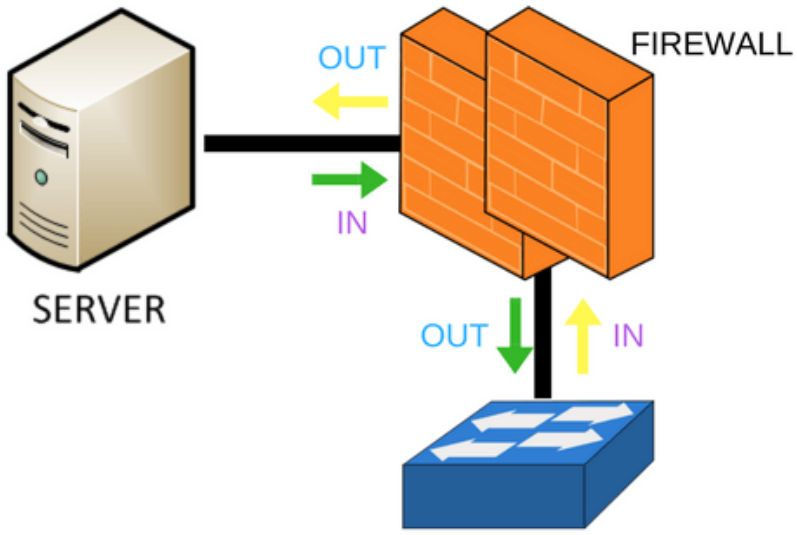

Figura 5.4

Dupa cum poti sa vezi si in figura 5.4, traficul trimis din reteaua Switch-ului va **INTRA (IN)** pe interfata Firewall-ului si va **IESII (OUT)** pe interfata conectata la Server. Exact opusul se intampla atunci cand server-ul raspunde celui care l-a contactat.

Daca ti-a placut ce ai invatat pana acum si vrei sa afli si mai multe, atunci iti recomand sa studiezi mult mai pe larg in cartea de ***Hacking cu Kali Linux - Invata elementele de baza despre Hacking si CyberSecurity.***