**Protectie si siguranta in functionare**

Studenti: **Chirigiu Corin-Adrian**

**Otelea Stefan** Profesor coordonator: **Conf.Dr.Ing.Stefan Stancescu**

Grupa:**433A**

**Cuprins**

Capitolul I:Concepte Fundamentale de Protectie si Siguranta(**Otelea Stefan**)………………………………….….……………....…4

1.1.Notiuni generale de securitate……………………………………………………………………………………………………….……..…..…4

1.2.Protectie si siguranta in Internet………….…………………………………………………………………………………………………....…5

1.2.1.Securitatea terminalelor mobile……………………………………………………………………………………………………..…..5

1.2.2.Securitatea retelelor………………………………………………………………………………………………………………………..…..6

1.2.3.Securitatea WWW………………………………………………………………………………………………………………………………..6

1.3.Principalele amenintari………………………………………………………………………………………………………………………………...7

1.3.1.Vulnerabiliati……………………………………………………………………………………………………………………………………….7

1.3.2.Exploatarea(“EXPLOIT”)……………………………………………………………………………………………………………………….7

1.3.3.Calul Troian………………………………………………………………………………………………………………………………………….8

1.3.4.Virusi si “Viermi”………………………………………………………………………………………………………………………………….8

1.3.5.Malware……………………………………………………………………………………………………………………………………..……….9

1.3.6.Payloads…………………………………………………………………………………………………………………………………………..….9

1.4.Comparatie Linux/Windows…………………………………………………………………………………………………………………..…….10

1.4.1.Caracteristici de comparatie……………………………………………………………………………………………………………....10

1.4.2.Linux/Unix……………..…………………………………………………………………………………………………………………………..11

1.4.3.Windows ……………………………………………………………………………………………………………………………………………18

Capitolul II:Functiile de securitate ale Win32API(**Otelea Stefan**).....................................................……………………..…19

2.1.Notiuni generale de Win32API…………………………………………………………………………………………………………………….19

2.2.Protectie si siguranta in functionare……………………………….…………………………………………………………………………….20

2.2.1.Accesibilitate…………………………………………………………………………………………………………………………………………20

2.2.2.Autentificare………………………………………………………………………………………………………………………………………….21

2.2.3.Autorizatie…………………………………………………………………………………………………………………………………………….24

2.2.4.Criptografie…………………………………………………………………………………………………………………………………………..25

2.2.5.Managementul parolei prin MS-CHAP(Microsoft Challenge-Handshake Authentication Protocol)…….….26

2.3.Securitatea informatiei in Romania…………………………………………………………………………………………………………………27

Capitolul III:Apelurile de sistem legate de securitate la Linux(**Chirigiu Corin**)............................................................29

3.1.Anatomia unui apel……………………………………………………………………………………………………………………………………….29

3.2.Exemple de apeluri de sistem………………………………………………………………………………………………………………………..30

Capitolul IV:Implementarea securitatii la Windows/Linux(**Chirigiu Corin**)………………………..………………….…………………35

4.1.Caracteristici generale Windows/Linux…………………………………………………………………………………………………..……….35

4.2. Accesul bazat pe niveluri de privilegii Windows/Linux…………………………………………………………………………………….37

4.3.Concluzii…………………………………………………………………………………………………………………………………………………………..41

Capitolul V:Bibliografie…………………………………………………………………………………………………………………………………………..42

**Capitolul I:Concepte Fundamentale de Protectie si Siguranta(Otelea Stefan)**

* 1. **Notiuni generale de securitate**

Securitatea informatiei, de multe ori prescurtata InfoSec(Information Security) consta in gasirea unei solutiei de protectie a informatiei fata de accesul, folosirea, dezvaluirea, intreruperea, modificarea, inregistrarea sau distrugerea neautorizata a acesteia.Termenul de de natura generala si poate fi folosit in orice fel de stiinta sau domeniu indiferent de forma pe care o ia informatia.

Cateva dintre aspectele generale ale securitatii informatiei sunt:

->**Securitatea Tehnologiei Informatiei**.De multe ori cunoscuta si ca Securitatea Calculatoarelor, acest concept consta in siguranta informatiilor aplicata in tehnologie(de cele mai multe ori pe diferite tipuri de calculatoare).Este important de stiut faptul ca prin termenul de calculator(computer) nu se face referire neaparat la un calculator personal(Desktop), un computer fiind de fapt orice sistem de calcul format din unul sau mai multe procesoare si un spatiu de memorie de diferite dimensiuni sau formate. Aceste dispzitive pot include de la simple sisteme de calcul la dispozitive mobile conectate constant la o retea de comunicatii cum ar fi smartphoneurile sau tabletele.Specialistii in domeniu sunt intalniti in majoritatea industriilor si mediilor de afaceri datorita cresterii nivelului de tehnologizare la nivel mondial si a vulnerabilitatii datelor care sunt din ce in ce mai valoroase.Acestia au responsabilitatea de a pastra in interiorul companiei pentru care lucreaza tehnologia utilizata si de a apara datele utilizate de atacuri cibernetice care adesea au scopul de a gasi brese in sistemele de securitate si de a patrunde neautorizat preluand controlul sistemelor interne.

->**Asigurarea Informatiei.**Procesul de asigurare a informatiilor nu este afectat de aparitia unor evenimente de tipul dezastrelor natural, intreruperea functionarii calculatoarelor sau a serverelor, furtul fizic sau oricare alte evenimente nedorite care ar conduce la pierderea datelor. In zilele noastre, una din cele mai comode si moderne metode de a asigura integritatea datelor este crearea unor copii de rezerva(backups) care sa foloseasca in cazul intampinarii uneia din problemele de mai sus.

Guvernele, serviciile secrete, corporatiile, institutiile financiare, spitalele si intreprinderile private acumuleaza o mare cantitate de informatii confidentiale despre angajatii lor, clientii, produsele, cercetareasi situatia financiara . Cele mai multe dintre aceste informatii sunt in prezent colectate, procesate si stocate pe calculatoare electronice si transmise prin retele la alte calculatoare .  
In cazul in care informatii confidentiale despre un client sau noua linie de produse se ajung in proprietatea unui concurent sau aunui hacker, afacerea si clientii sai ar putea suferi pierderi financiare de amploare, ireparabile, mai mult, se poate pune si problema reputatiei companiei. Protejarea informatiilor confidentiale este o cerinta obligatorie in toate mediile de afaceri si, in multe cazuri, o cerinta etica si legala.  
Pentru un individ, securitatea informatiilor are un efect semnificativ asupra vietii private ,aspect care difera foarte mult de la o cultura la alta.  
Domeniul de securitate a informatiilor a crescut si a evoluat in mod semnificativ in ultimii ani. Exista multe modalitati de a obtine o cariera in acest domeniu. Acesta ofera multe domenii de specializare, incluzand securizarea retelelor si a infrastructurii conexe, securizarea aplicatiilor si a bazelor de date, teste de securitate, sisteme de informatii de audit, planificarea continuitatii afacerii dar si criminalistica digitala.

**Integritatea informatiei.**In securitatea informatiilor, integritatea datelor inseamnamentinerea si asigurarea acurateteisi consistentei datelor pe intregul sau ciclu de viata.Acest lucru inseamna ca datele nu pot fi modificate intr-un mod neautorizat sau denedetectat. Integritatea este incalcat atunci cand un mesaj este modificat In mod activ si rau intentionatin tranzitul informatiei de la un sistem la altul. Sistemele de securitate genereaza de obicei mesaje referitoare la inegritatea informatiei transmise intrun sistem de telecomunicatii pe langa mesajele referitoare la confidentialitatea informatiei.

**Disponibilitatea informatiei.**Pentru ca orice sistem de informatii sa-si serveasca scopul sau, informatiile trebuie sa fie disponibile atunci cand este nevoie. Acest lucru inseamnaca sistemele de calcul utilizate pentru a stoca si procesa informatiile, controalele de securitate folosite pentru a proteja informatiile si canalele de comunicare folosite pentru a accesa informatiile trebuie sa functioneze corect. Sistemele cu o inalta disponibilitate au ca principal scop sa ramanadisponibile in orice moment, prevenind intreruperile de servicii din cauza penelor de curent, erorilor de hardware sau din cauza actualizarilor de sistem.Pentru a asigura disponibilitatea este necesara prevenirea de atacuri de blocare a serviciilor cum ar fi supraaglomerarea cu mesaje a sistemului tinta, ceea ce il forteaza sa se inchida pentru a asigura un alt atribut esential si anume integritatea informatiei.

**Autenticitatea.**In domeniul tehnologiei informatiei cum ar fi afacerile desfasurate pe o platforma online, este necesar sa se asigure ca datele, tranzactiile, comunicatiile sau documentele (electronice sau fizice) sunt autentice. De asemenea, este important pentru autenticitate faptul caambele parti implicate intrun schimb de date electronicesa fie cu adevarat cine pretind a fi. Unele sisteme de securitate a informatiilor incorporeaza caracteristici de autentificare cum ar fi "semnaturi digitale", care dau dovadaca datele sunt autenticesi a fost trimis de cineva care poseda cheia corecta de semnare.

* 1. **Protectie si siguranta in Internet**

1.2.1.Securitatea terminalelor mobile

**„Mobile Security”** sau securitatea telefon mobil a devenit tot mai importanta in domeniul computerelor mobile(smartphone-urilor). Este de interes special deoarece informatiile personale si legate de afaceri sunt acum stocate pe smartphone-uri .  
Tot mai multi utilizatori si companii folosesc **smartphone-uri** ca instrumente de comunicare, dar, de asemenea, ca un mijloc de planificare si organizare a muncii lor dar si a vietii private.In cadrul companiilor, aceste tehnologii cauzeaza schimbari profunde in organizarea sistemelor de informatii si, prin urmare, au devenit sursa unor noi riscuri.Intr-adevar, smartphone-urile aduna si compileaza o cantitate tot mai mare de informatii sensibile la care accesul trebuie sa fie controlat pentru a proteja viata privata a utilizatorului si proprietatea intelectuala a companiei. Conform ABI Research piata Mobile Security Services va totaliza aproximativ 1.88 miliarde dolari la sfarsitul anului 2013,acesta prefigurandu-se a fi doar startul a ceva cu totul indispensabil in viitor.  
Toate smartphone-urile, la fel ca si calculatoarele, sunt preferate a fi tinte ale atacurilor . Aceste atacuri exploateza punctele slabe ale smartphone-urilor care pot proveni de la mijloace de comunicare cum ar fi serviciul de **mesaje scurte ( SMS ), serviciul de mesaje multimedia ( MMS ), retelele Wi-Fi, Bluetooth si GSM** , standardul global de facto pentru comunicatii mobile. Exista, de asemenea, atacuri care exploateaza vulnerabilitatile de software atat din browser-ul web cat si din sistemul de operare. In cele din urma, exista forme de software rau intentionate care se bazeaza pe cunoasterea slaba a utilizatorilor neavizati .  
Diferite contra -masuri de securitate sunt dezvoltate si aplicate pe smartphone-uri, de la securitate in diferite straturi ale software-ului pentru diseminarea de informatii catre utilizatorii finali.Exista bune practici care trebuie respectate la toate nivelurile, de la proiectare pana la dezvoltarea prin intermediul sistemelor de operare, straturilor de software si aplicatiilor descarcate.

1.2.2.Securitatea retelelor

**Securitatea retelelor** este formata din dispozitiile si politicile adoptate de catre un administrator de retea pentru a preveni si a monitoriza accesul neautorizat, utilizarea incorecta, modificarea sau practicile prin care o retea sau resursele ei devin inoperabile.Securitatea retelelor presupune autorizarea de acces la date intr-o retea, care este controlata de administratorul de retea. Utilizatorii isi aleg sau le sunt atribuite un **ID si o parola** sau alte informatii de autentificare care le permite accesul la informatii si programe potrivite autoritatii lor. Securitatea retelelor se refera la o varietate de retele de calculatoare, atat publice, cat si private, care sunt utilizate in locuri de munca uzuale cum ar fi efectuarea de tranzactii si comunicarea intre companii, agentii guvernamentale si persoane fizice. Retelele pot fi private, cum ar fi intr- o companie, iar altele pot fi deschise accesului public. Securitatea retelelor este implicata in organizatii, intreprinderi, si alte tipuri de institutii.Astfel se explica titlul paragrafului: se asigura reteaua, precum si protejarea si supravegherea operatiunilor in curs de efectuare. Cel mai comun si simplu mod de a proteja o resursa de retea este de a ii atribui un nume unic si o parola corespunzatoare

1.2.3.Securitatea WWW

**World Wide Web** (abreviat ca WWW sau W3,cunoscut sub numele de web)este un sistem de documente hipertext interconectate si accesate prin intermediul internetului. Cu un browser web, se pot vizualiza pagini Web care pot contine text, imagini, clipuri video si alte date multimedia si naviga intre ele, prin intermediul hyperlink-urilor.

Web-ul a devenit cea mai noua nisa preferata de infractori de raspandire a malware-ului. **Criminalitatile cibernetice** efectuate de pe Web pot include furtul de identitate, frauda, spionaj si culegerea de informatii.Vulnerabilitatile online sunt acum mai numeroase decat problemele de securitate traditionale de calculator,si , masurata de Google, aproximativ unul din zece pagini web poate contine cod malitios.

De fiecare data cand o pagina web este ceruta de la un server web, serverul poate identifica, si, de obicei, se conecteaza adresa IP de la care cererea a sosit. In egala masura , cu exceptia anumitor cazuri, cele mai multe browsere web inregistreaza paginile web care au fost solicitate si vazute intro sectiune de istoric, unde este retinuta si cea mai mare parte a continutului de pe plan local.Cu exceptia cazului in care este folosita criptarea HTTPS , cererile de web si raspunsurile sunt transmise in text simplu pe internet si pot fi vizualizate, inregistrate si stocate de sistemele intermediare .  
Atunci cand o pagina web cere si informatiile de identificare personala, cum ar fi numele real, adresa, adresa de e –mail, etc, o conexiune poate fi facuta intre traficul web si acea persoana . In cazul in care site-ul foloseste cookie-uri HTTP, numele de utilizator si parola de autentificare, sau alte tehnici de urmarire, atunci acesta va fi capabil sa faca referire la alte vizite pe pagina respectiva, conform informatiilor de identificare furnizate. In acest fel este posibil ca o organizatie bazata pe web sa construiasca un profil al persoanelor care folosesc site-ul sau site-urile sale. Aceasta poate construi o baza de date confidentiala pentru o persoana, care include informatii cu privire la activitatile lor de petrecere a timpului liber, interesele lor comerciale, profesia lor, precum si alte aspecte legate de profilul lor demografic. Aceste profiluri sunt, evident, de interes potential pentru companiile de marketing, agentiile de publicitate si altele. In functie de termenii si conditiile site-ului si legile locale, informatiile de acest gen pot fi vandute, sau folosite de alte organizatii fara ca utilizatorul sa fie informat. Pentru multi oameni obisnuiti, aceasta inseamna mai mult decatniste e-mail-uri neasteptate in Inbox, sau niste anunturi de publicitate,utilizarea abuziva a acestor informatii devenind de multe ori probe in procese de santaj economic sau diferite tipuri de abuzuri. Deasemenea agentiile de combatere a terorismului si spionaj pot identifica anumite persoane tinta, bazate pe informatiile ce par a fi interesele si inclinatiile lor personale.

**1.3.Principalele amenintari**

1.3.1.Vulnerabilitatea

In securitatea calculatoarelor, o **vulnerabilitate** este o slabiciune care permite unui atacator sa reduca asigurarea de informatii dintr-un sistem. Vulnerabilitatea este la intersectia a trei elemente : un sistem susceptibil sau defect, accesul atacatorului la defect, si capacitatea atacatorului de a exploata defectul. Pentru a exploata o vulnerabilitate, un atacator trebuie sa aiba cel putin un instrument aplicabil sau tehnica care se poate conecta la slabiciunea unui sisitem. In acest cadru, vulnerabilitatea este cunoscuta ca suprafata de atac.  
Gestiunea vulnerabilitatii consta in actiunea ciclica de identificare, clasificare, remediere, si de atenuare a vulnerabilitatilor. Aceasta practica ,in general, se refera la vulnerabilitati software in sisteme de calcul.  
Un risc de securitate poate fi clasificat ca o vulnerabilitate. Utilizarea vulnerabilitatii cu aceeasi semnificatie de risc poate duce la confuzii. Riscul este legat de potentialul unei pierderi semnificative. Apoi exista vulnerabilitati fara risc: de exemplu, atunci cand bunul afectat nu are nici o valoare . O vulnerabilitate cu unul sau mai multe exemple de atacuri functionale puse in aplicare este clasificat ca fiind o vulnerabilitate de exploatat- o vulnerabilitate pentru care exista o exploatare . Fereastra de vulnerabilitate este timpul de cand gaura din securitate a fost sesizata sau manifestata in software-ul vizat, pana cand accesul a fost eliminat, sau o remediere a securitatii a fost implementata.  
**Bugul de securitate** ( defect de securitate ) este un concept mai restrans : exista vulnerabilitati care nu sunt legate de software: hardware-ul , site-ul , vulnerabilitatile de personal sunt exemple care nu sunt bug-uri de securitate software .  
Constructiile din limbajele de programare, care sunt dificil de utilizat in mod corespunzator pot fi o sursa mare de vulnerabilitati.

**Interceptarea** este actul discret de a asculta conversatia privata a altora fara consimtamantul lor, astfel cum sunt definite prin **Black’s Law Dictionary**.Acest lucru este de obicei considerat a fi lipsit de etica si exista o veche zicalaca"interceptorii aud rareori ceva bun de ei insisi. ..interceptorii incerca intotdeauna sa asculte problemele care ii privesc. "

1.3.2.Exploatarea

O **exploatare** (de la verbul englezesc **to exploit**, care Inseamna "folosind ceva pentru avantajul propriu") este o portiune de software, un cadru de date, sau o secventa de comenzi care profita de un bug sau o vulnerabilitate, in scopul de a provoca un comportament neintentionat sau neprevazut care sa apara pe software, hardware, sau ceva electronic (de obicei computerizate). Un astfel de comportament include frecvent lucruri, cum ar fi obtinerea de control a unui sistem informatic, care permite escaladarea privilegiilor, sau un atac de intrerupere a serviciilor.Exista mai multe metode de **clasificare** a exploatarilor . Cea mai comuna este cum intra in contact software-ul cucu vulnerabilitatea.Prin exploatare controlata se intelege accesul intr-o retea si exploatarea unei vulnerabilitati de securitate, fara niciun drept de acces anterior la sistemul vulnerabil. O exploatare locala necesita acces in sistemul vulnerabil si, de obicei, creste privilegiile persoanei exploatante care profita de drepturile acordate de catre administratorul de sistem persoanelor avizate. Exploatarea impotriva aplicatiilor client , de asemenea, exista , de obicei, formata din servere modificate, care trimit un **„exploit”**daca sunt accesate cu o aplicatie client. Exploatarile impotriva aplicatiilor client pot necesita , de asemenea, interactiunea cu utilizatorul si, astfel, pot fi utilizate In combinatie cu metode de inginerie sociala . O alta clasificare este in functie de actiunea impotriva sistemului vulnerabil ; accesul neautorizat de date sau executarea de cod arbitrar fiind cateva dintre exemple. Multe **„exploit”** sunt concepute pentru a oferi acces la nivel de **superuser** la un sistem informatic. Cu toate acestea , este de asemenea posibil sa se utilizeze mai multe **exploit-uri** , mai intai pentru a avea acces de nivel scazut , apoi sa escaladeze privilegii repetat pana cand se ajunge la radacina . In mod normal, un singur **exploit** poate doar profita de o vulnerabilitate de software specific .

1.3.3.Trojan horse(cal troian)

Un cal troian sau troian , in calcul este un tip de non-auto-replicare de program malware care contine cod malitios, care, atunci cand esteexecutat, desfasoara actiuni determinate de natura „**troian”** , de obicei cauzand pierderea sau furtul de date, si posibil deteriorarea sistemului.Un troian de multe ori actioneaza ca un**backdoor**, de a contacta un operator, care poate avea apoi acces neautorizat la computer-ul afectat. **Troienii si backdoorii** nu sunt insa usor de detectat, dar in cazul in care efectueaza calcul semnificativ sau executa activitati de omunicatii poate provoca o rulare mult mai lenta a calculatorului . Programele malware sunt clasificate ca troieni daca nu Incearca sa se infiltreze in alte fisiere ( virus de calculator ) sau sa se propage singuri ( vierme ) in sistem. Un computer poate gazdui un troian prin intermediul unui program de malware iar utilizator este pacalit in timpul executiei ( de multe ori un atasament de e-mail deghizat pentru a nu fi suspicios , de exemplu , un formular de rutina pentru a fi completat ).

1.3.4.Virusi si „viermi”

Un virus de calculator este un program de malware care , atunci cand este executat, se reproduce prin inserarea unor copii ( posibil modificate ) in alte programe de pe calculator, fisiere de date, sau sectorul de boot de pe hard disk; atunci cand aceasta replicare reuseste, se spune ca zonele afectate sunt „infectat” . Virusii efectueaza de multe ori un anumit tip de activitati daunatoare pe gazde infectate, cum ar fi: fura spatiu pe hard disk sau timp de procesare al CPU sau acceseaza informatii private. Cu toate acestea , nu toate virusurile transporta o informatie distructiva sau incearca sa se ascunda -caracteristica definitorie a virusurilor este ca ele sunt programe de auto-replicare de calculator care se instaleaza fara consimtamantul utilizatorului.  
Creatorii de virusi se folosesc de metode socialesi exploateaza cunostinte detaliate de vulnerabilitati de securitate pentru a obtine accesul la resurse de calcul a gazdelor lor. Marea majoritate a virusurilor tintesc sistemele care ruleaza Microsoft Windows ,folosind o varietate de mecanisme pentru a infecta noi gazde ,si de multe ori folosind strategii **anti-detection/stealth** complexe pentru a se sustrage din calea antivirusurilor . Motivele pentru crearea de virusi pot include profitul dorit , dorinta de a transmite un mesaj public, amuzament personal, pentru a demonstra ca exista o vulnerabilitate in software, pentru sabotaj, sau pur si simplu pentru ca se doreste explorarea vietii artificialesi a algoritmilor evolutivi .  
Virusurile de calculator produc in prezent daune economice de miliarde de dolari in fiecare an, din lipsa sistemelor de securitate cauzand irosirea resurselor de calculator, coruperea datelor, cresterea costurilor de intretinere, etc.Ca raspuns, au fost dezvoltate,instrumente antivirus gratuite, dar a si aparut o industrie de multe miliarde de doralri de furnizori de software antivirusi. Din pacate, software-ul antivirus nu estein prezent capabil de a captura toate virusurile de calculator ( mai ales cele noi ) ; cercetatorii in securitate sunt in cautare de noi modalitati de a permite solutii antivirus pentru a detecta mai eficient virusi in curs de dezvoltare , inainte ca acestia sa devina deja distribuiti pe scara larga .

Un **vierme informatic** este un program malware independent de calculator care se reproduce, in scopul de a se raspandi la alte computere . De multe ori, se foloseste o retea de calculatoare pentru a se raspandi, bazandu-se pe bresele de securitate de pe computerul tinta pentru a-l accesa. Spre deosebire de un virus de calculator, acesta nu are nevoie de a se atasa la un program existent . Viermii provoca aproape intotdeauna cel putin unele daune retelei, chiar daca acestea constau in reducerea latimii de banda , in timp ce virusii aproape intotdeauna corup sau modifica fisiere de pe un calculator tinta.  
Multi viermi sunt proiectati doar sa se raspandeasca, si nu incearca sa schimbe sistemele prin care trec. Totusi, asa cum viermii **Morris** si **MyDoom** expuneau chiar si aceste "sarcini utile libere" viermii pot provoca perturbari majoreprin cresterea traficului in retea si alte efecte nedorite. O "sarcina utila" este codul conceput pentru a face mai mult decat se raspandeasca-ar putea sterge fisiere pe un sistem gazda ( de exemplu , viermele **ExploreZip** ), cripta fisiere intr-un **atac cryptoviral**de fonduri , sau trimite documente prin e-mail. O sarcina utila foarte comuna pentru viermi este de a instala un backdoor in computerul infectat, pentru a permite crearea unui "**zombie**" computer sub controlul viermelui autor. Retele de astfel de masini sunt adesea mentionate ca **botnet-uri**si sunt foarte frecvent utilizate de catre expeditorii de spam pentru trimiterea de junk e-mail sau pentru mascarea adresei site-ului lor.**Spammerii** sunt , prin urmare, considerati a fi o sursa de finantare pentru crearea de astfel de viermi ,„scriitorii” de viermi fiind prinsi vanzand liste de IP ale calculatoarelor infectate.

1.3.5. Malware

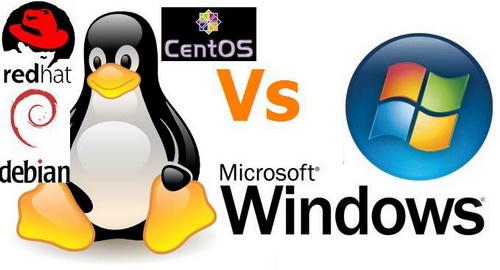
Malware, prescurtarea de la software rau intentionat, este un software folosit pentru a perturba functionarea computerului, adunand informatii sensibile, sau obtinand acces la sistemele informatice private. Acesta poate aparea sub forma de cod ,script-uri, continut activ, si alte software-uri . "Malware" este un termen general utilizat pentru a se referi la o varietate de forme de software ostil sau deranjante .   
Malware-ul include virusi de calculator ( inclusiv viermi , cai troieni ) , **ransomware , spyware , adware , scareware** , si alte programe malware . Majoritatea amenintarilor malware active sunt, de obicei, viermi sau troieni, mai degraba decat virusuri . In lege, malware-ul este uneori cunoscut ca un contaminant al calculatorului. Malware-ul este diferit de software-ul defect , care este un software legitim dar contine bug-uri nocive care nu au fost corectate inainte de lansare. Cu toate acestea, unele malware-uri sunt deghizate ca software-ul autentic, si poate veni de la un site oficial al companiei, sub forma unui program util sau atractiv, care are malware-ul nociv incorporat in el,impreuna cu software-ul de urmarire suplimentar care aduna statistici de marketing .  
Software-ul , cum ar fi anti-virus , anti-malware sau firewall-urii sunt utilizate de catre utilizatorii casnici si organizatii din intreaga lume pentru a incerca sa se protejeze impotriva acestor atacuri.

1.3.6.Payloads  
  
In securitatea calculatoarelor, sarcina utilase refera la partea de malware care efectueaza o actiune cu potential periculos.

Un **rootkit** este un tip ascuns de software , de obicei, rau intentionat, proiectat pentru a ascunde existenta unor procese sau programe de metodele uzuale de detectare si permite in continuare un acces privilegiat la un calculator.Termenul de „ROOTKIT” este o inlantuire de "ROOT=radacina" ( numele traditional al contului privilegiat pe sistemele de operare Unix ) si cuvantul "kit" ( care se refera la componentele software care pun in aplicare instrumentul ). Termenul de "rootkit" are conotatii negative, prin asocierea sa cu malware .  
Instalarea **rootkit-urilor** pot fi automatizate, sau un atacator il poate instala odata ce a obtinut radacina sau accesul de administrator . Obtinerea acestui acces este un rezultat de atac direct la un sistem ( de exemplu, exploatarea unei vulnerabilitati cunoscute, parola ( fie prin cracare ,fie prin escaladarea privilegiilor) ) . Odata instalat, devine posibila ascunderea patrunderii precum si mentinerea unui acces privilegiat. Cheia este accesul root / administrator. Control deplin asupra unui sistem inseamna ca software-ul existent poate fi modificat, care altfel ar putea fi folosit pentru a detecta patrunderea respectiva.  
Detectarea rootkit-urilor este dificila deoarece un rootkit poate fi capabil de a submina software-ul, care are ca scop destinat sa-l gaseasca. Metodele de detectare includ utilizarea unei alternative de sistem de operare de incredere, scanarea semnaturii, diferenta de scanaresi analiza de imagine de memorie .Stergerea poate fi complicata sau practic imposibila, in special in cazurile in care rootkit-ul are ca gazda**kernel-ul** ; reinstalarea sistemului de operare poate fi singura solutie a acestei probleme. Atunci cand se ocupa cu rootkit-uri de firmware stergerea poate solicita inlocuirea hardware-ului sau a echipamentelor specializate .

**1.4.Comparatie Linux/Windows**

1.4.1.Caracteristici de comparatie



**1. Pret**

Tinand cont de starea actuala a economiei pretul joaca un rol major in alegerea sistemului de operare. Din ce in ce mai multe persoane acorda o atentie bine meritata sistemului de operare Linux. In aceasta zona suprematia Linux-ului nu poate fi contestata deoarece majoritatea distributiilor sunt oferite gratuit. Poate va trece prin minte intrebarea “De ce este gratuit?”, raspunsul e simplu: Linux este creat si dezvoltat de o comunitate de programatori care nu lucreaza pentru aceeasi companie, Linux a fost gratuit inca de la inceputurile sale.Majoritatea programelor care au fost create pentru Linux sunt gratuite. Exista alternative la toate programele comerciale care ruleaza doar pe Windows iar faptul ca sunt gratuite nu le face mai putin calitative. In unele cazuri aceste programe gratuite si open source sunt mai bune decat alternativele comerciale.

**2. Libertate**

Folosind sistemul de operare Linux aveti libertatea de a alege, nu putem spune acelasi lucru despre Windows care va blocheaza la modul in care compania Microsoft considera ca ar trebui sa functioneze un sistem de operare. Microsoft considera ca daca pune la dispozitia utilizatorilor o bara de activitati, un buton Start, icoane si un system tray este suficient. Pentru unii poate asa este, dar majoritatea utilizatorilor vor sa aiba ceva diferit, personalizat sau cu mai multe functionalitati. Folosind Linux poti face sistemul de operare sa arate exact cum iti doresti, singurele limite sunt timpul si imaginatia.

**3. Ierarhia fisierelor de sistem**

In Linux se foloseste un sigur sistem ierarhic, totul incepe in directorul root “/”. Unitatile de stocare fiind etichetate /dev/sda, /dev/sdb etc. In Windows sistemul ierarhic este multiplu si depinde de numarul unitatilor de stocare, se foloseste un root pentru fiecare unitate de stocare. Sub Linux doar o unitate de stocare contine directorul root, celelalte unitati de stocare prezente vor fi montate in directorul /media/.

**4. Suport Hardware**

Aici lucrurile sunt un pic complicate deoarece sistemul de operare Windows are un segment de piata mult mai mare (chiar urias) iar majoritatea producatorilor de componente hardware vor ca produsul lor sa fie compatibil 100% cu Windows. Sub Linux suportul hardware depinde de modul in care producatorul este convins de catre dezvoltatori sa predea specificatiile. Se pot intalni cazuri in care specificatiile nu sunt eliberate de producator iar respectivele componente hardware nu vor functiona corespunzator sub Linux. Totusi in ultimii ani a fost acordata o atentie din ce in ce mai mare de producatorii de hardware sistemului de operare Linux iar cazurile in care o componenta hardware nu functioneaza pe Linux sunt destul de izolate.

**5. Securitate**

Acest subiect este foarte dezbatut de abele parti. Poate din cauza cotei de piata uriase, a vulnerabilitatilor si a atentiei acordate sistemului de operare Windows il fac mult mai slab la acest capitol decat Linux-ul. Principala vulnerabilitate a Windows-ului o reprezinta accesul la root. Pentru a face pagube pe un sistem Linux trebuie neaparat sa stii parola de acces la root. Asta nu inseamna ca Linux-ul este sigur 100%, sunt multe gauri de securitate si in Linux. In momentul cand este descoperita o vulnerabilitate in Linux aceasta este rezolvata de catre comunitate foarte repede pe cand Microsoft a demonstrat de multe ori ca au nevoie de prea mult timp pentru a rezolva o problema.

**1.4.2.Linux/Unix**

**a)UID/GID**

**UID**

Sistemele de operare „**Unix-like**” identifica utilizatorii in kernel cu o valoare de tip **integer** fara semn numit identificator de utilizator, de multe ori abreviat la **UID** sau ID utilizator. Gama de valori UID variazaintre diferite sisteme; cel putin, un UID reprezinta un intreg de 15 biti variind intre 0 si 32767, cu urmatoarele restrictii :  
    ->Superutilizatorul trebuie sa aibaintotdeauna un UID de la zero ( 0 ) .  
    ->Utilizatorul „Nimeni”in mod traditional va primi cel mai mare UID posibil (ca opus al superuser): 32767 .  
    -> Conventia isi rezerva UID 1-100 pentru utilizarea sistemului; unele manuale recomanda UID rezervea de la 101 pana la 499 ( Red Hat Enterprise Linux ) sau chiar de pana la 999 ( Debian ).  
Valoarea UID poate face referire la parolele,fisierele,documentele utilizatorului. Fisierele cu parole si Serviciul Informatiilor Retelei se referade asemenea la UID-uri numerice. Identificatorul de utilizator a devenit o componenta necesara a sistemelor de fisiere si a proceselor Unix . Unele sisteme de operare suporta UID-uri de 16 biti, ceea ce face 65536 de ID-uri unice posibile; un sistem modern cu UID-uri pe 32 de biti va crea 4294967296 ( 232) de valori distincte disponibile.  
In mediile compatibile POSIX linia de comanda va da numarul de identificator al utilizatorului, precum si mai multe informatii cu privire la contul de utilizator, cum ar fi numele de utilizator, grupul utilizator principal si identificatorul de grup ( GID ).

**GID**

In sistemele Unix, mai multi utilizatori pot fi clasatiin grupe. Sistemele de fisiere POSIX(Portable Operating System Interface) si Unix conventionale sunt organizate in trei clase: de utilizator, grup, si altele. Utilizarea grupurilor permite abilitati suplimentare care urmeaza sa fie delegate intr-un mod organizat, cum ar fi accesul la discuri, imprimante si alte periferice. Aceasta metoda permite, de asemenea, superutilizatorului de a delega anumite sarcini administrative pentru utilizatorii normali.  
  
Un identificator de grup , de multe ori abreviat la GID , este o valoare numerica folosita pentru a reprezenta un grup specific . Intervalul de valori pentru o GID variazaintre diferite sisteme ; cel putin , o GID poate fi intre 0 si 32767 , cu o restrictie : grupul de conectare pentru superuser trebuie sa aiba GID 0. Fisierele cu „**shadow password**”si Serviciul de Informatii de Retea se refera de asemenea la valori GID numerice . Identificatorul de grup este o componenta necesara pentru sistemele de fisiere si procese Unix.  
Limitele gameide identificatori de grup depind de spatiul de memorie utilizat pentru a le stoca. Initial, a fost folosit un „**signed integer**” de 16 - biti . Avand in vedere ca semnul nu a fost neaparat necesar numerele negative nu desemneaza grup valabil ID; un intreg fara semn este folosit in prezent, fiind permise doar grup ID-urile intre 0 si 65.535 . Sistemele de operare moderne folosesc, de obicei, „**unsigned integers**” pe 32 de biti , care permit ID-uri de grup intre 0 si 4294967295.Exista uncomutator de 16-32 biticare nu a fost initial necesar, o masina sau chiar o retea nu a servit mai mult de 65.536 de utilizatori la momentul dat , dar a fost facut pentru a elimina nevoia de a face acest lucru in viitor, cand va fi mult mai dificil de utilizat.

**Efective UID/GID**

UID efectiv ( EUID ) al unui proces este proprietatea atribuita fiserelor create de acel proces. GID efectiv ( EGID ) a unui proces poate afecta, de asemenea, crearea de fisiere in functie de semantica de punere in aplicare a kernel-ului specific utilizat si eventual optiunile de montare utilizate. In conformitate cu semantica BSD Unix , proprietatea grupului data unui fisier nou creat este neconditionat mostenita de la proprietatea grupului din directorul in care acesta este creat. In conformitate cu semantica sistemului AT & T 5 Unix (de asemenea adoptata de mai multe variante de Linux ) fisierelor nou create le va fi dat in mod normal, dreptul de proprietate de grup a EGID-ului procesului care le creeaza .  
Unele sisteme de fisiere pot implementa optiuni pentru selectarea in timp real care dintre semantica BSD sau AT & T ar trebui sa fie utilizate in ceea ce priveste proprietatea grup de fisiere nou create.  
In timpul accesului la fisiere , kernel-ul foloseste EUID si EGID eficienta a procesului pentru a stabili daca se poate accesa fisierul

**FSUID(File system User ID)**

Linux are, de asemenea, un ID utilizator al fisierelor de sistem(fsuid), care este folosit in mod explicit pentru controlul accesului la sistemul de fisiere. De fiecare data cand EUID este schimbat, schimbarea se propaga si la FUID.  
Intenia cu care FSUID a fost creat este de a permite programelor (de exemplu, serverul de NFS-Network File System) sa se limiteze la drepturile de sistem de fisiere a unor UID-uri fara a mai da permisiunea UID pentru a le trimite semnale. Incepand cu kernel 2.0, existenta FSUIDnu mai este necesara, deoarece Linux adera la regulile SUSv3 (Single UNIX Specification) pentru a trimite semnale, dar FSUIDramane, din motive de compatibilitate.

**RUID(Real User ID)**

RealUID(ruid) si RealGID(rgid) identifica proprietarul real al procesului si influenteaza permisiunile pentru a trimite semnale. Un proces fara privilegiul de superutilizator poate semnala un alt proces numai daca UID-ul real sau efectiv al expeditorului se potriveste cu UID-ulreal sau salvat(SUID)de la receptor.Procesele de tip„copil”mostenes cacreditarile de la procesele de tip „parinte”,apoise pot semnala reciproc.

**b)SETUID si SETGID**

SETUID si SETGID(prescurtarea de la "ID-ul setat de utilizator la executie" si"ID de grup setat la executare") sunt **steaguri** Unix pentru drepturi de acces care permit utilizatorilor sa ruleze un executabil cu permisiunile de proprietar al executabilului sau grup de proprietari al executabilului respectiv si de a schimba comportamentul indirector. Ele sunt adesea folosite pentru a permite utilizatorilorde pe un sistem de calcul sa ruleze programe cu privilegii ridicate temporar, in scopul de a indeplini o sarcina specifica in timp ce privilegiile de UID sau GID asumate oferite nu sunt intotdeauna ridicate,respectandu-se un minim de privilegii pentru acestea doua.   
SETUID si SETGID sunt necesare pentru sarcini care necesita privilegii mai mari decat cele pe care utilizatorii comuni le au, cum ar fi schimbarea parolei de autentificare. Unele dintre sarcinile care necesita privilegii ridicate pot sa nu fie imediat evidente, cum ar fi comanda ping care trebuie sa trimita si sa asculte de pachete de control de pe o interfata de retea.Atunci cand unui fisier executabil i-a fost dat atributul setuid, utilizatorii normali asupra sistemului care au permisiunea de a executa acest fisier obtin privilegiile utilizatorului care detine fisierul (de obicei root) in cadrul procesului creat . Cand privilegii de root au fost acumulate in cadrul procesului, aplicatia poate efectua apoi sarcini pe sistemul pe care utilizatorii obisnuiti in mod normal s-ar fi limitat a le face. Utilizatorului ii va fi interzisa de catre sistem modificarea noului proces, in orice mod, cum ar fi utilizarea ptrace, LD\_LIBRARY\_PATH sau trimiterea de semnale. ( semnale de la terminal vor fi inca acceptate, cu toate acestea ) .  
In timp ce functia de setuid este foarte utila in multe cazuri, utilizarea necorespunzatoare poate reprezenta un risc de securitate in cazul in care atributul setuid este atribuit programelor executabile care nu sunt atent proiectate . Din cauza unor probleme potentiale de securitate, mai multe sisteme de operare ignora atributul setuid atunci cand sunt aplicate la scripturi shell executabile .  
Atributul setgid va permite schimbul privilegiilor pe baza de grup in cadrul unui proces , asa cum face steagul setuid pentru privilegiile de tip utilizator de baza .  
Prezenta executabilelor setuid explica de ce apelul de sistem **chroot** nu este disponibil pentru utilizatorii non-root pe Unix .  
Bitii setuid si bit-ul setgid sunt in mod normal stabiliti cu **chmod** prin setarea cifrei 4 ( pentru setuid ) sau 2 ( pentru setgid ) . " **chmod 6711 file** " va stabili atat bit-ul setuid si bit-ul setgid ( 2 +4 = 6 ) , va seta fisierul pentru citire / scriere / executabil pentru proprietar, executabil pentru grup ( primul 1 ) si altii ( al doilea 1 ) . Atunci cand o alta persoanadecat proprietarul executa fisierul, procesul va rula cu permisiuni de utilizator si de grup stabilite de catre proprietarul sau.  
Steagurile setuid si bit-ul setgid, cand sunt setate pe un director, au un sens cu totul diferit.   
Setarea permisiunii setgid pe un director(chmod g+e) creeaza noi fisiere si subdirectoarele create in cadrul acestuia mostenind ID-ul de grup, mai degraba decat ID-ul grupului primar al utilizatorului care a creat fisierul(ID-ul proprietar nu este afectat, doarID-ul de grup). Subdirectoarele nou create vor mosteni bitul setgid. Astfel, acesta permite un spatiu de lucru comun pentru un grup fara inconvenientul de a cere membrilor grupului sa schimbe in mod explicit grupul lor actual, inainte de crearea de fisiere noi sau directoare. De retinut ca setarea permisiunii setgid pe un director afecteaza numai ID-ul de grupa noilor fisiere si subdirectoare create dupa ce bitulsetgid este setat, si nu se aplica entitatilor existente. Setarea bitului setgid pe subdirectoare existente trebuie sa fie facuta manual, cu o comanda cum ar fi urmatoarele:

**[root@foo]# find /path/to/directory -type d -exec chmod g+s '{}' \;**

Permisiunea setuid stabilita pe un director este ignorata pe sistemele UNIX si Linux. FreeBSD(sistem de operare Unix-like,gratuit provenit de la AT&T via BSD)poate fi configurat pentru a interpreta in mod analog setgid pentru a forta toate fisierele si sub-directoare sa fie detinute de catre proprietarul directorul de sus.   
Pe sisteme derivate din BSD, directoarele se comporta ca si cum bitul lor setgid este intotdeauna setat, indiferent de valoarea sa reala.

Dezvoltatorii ar trebui sa elaboreze si sa implementeze programe care folosesc cu atentie acest bit pe executabile pentru a se evita vulnerabilitati de securitate, inclusiv depasiri tampon. Atacuri de succes asupra aplicatiilor vulnerabile permit atacatorului sa execute cod arbitrar sub umbreladrepturilor procesului de exploatat. In cazul in care un proces vulnerabil foloseste bitul setuid pentru a rula ca root, codul se va executa cu privilegii de root, in vigoare oferind atacatorului acces de tip root la sistemul pe care procesul vulnerabil este in functiune.   
De o importanta deosebitain cazul unui proces setuid este mediul. In cazul in care mediul nu este „dezinfectat”in mod corespunzator de un proces privilegiat, comportamentul sau poate fi schimbat de procesul neprivilegiat care a pornit pe mediul respectiv. De exemplu, GNU libc a fost la un moment vulnerabil la o exploatare folosind setuid si o variabila de mediu, care a permis executarea codului de biblioteci partajate nesigur.

**c)MANAGEMENTUL ACCESULUI LA LINUX**

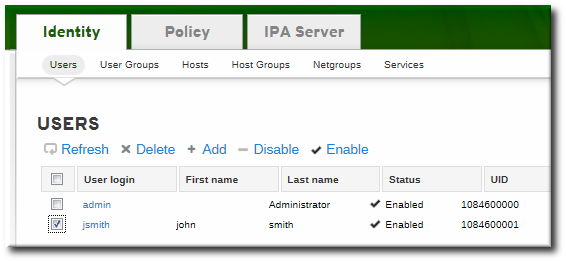
**1)Managementul Identitatii(Introducere)**

Informatiile de securitate se refera la identitatea utilizatorilor, masiniisi serviciilor . Odata ce identitatea este verificata , atunci accesul la servicii si resurse poate fi controlat .  
Pentru eficienta gestionarii riscurilor si a usurintei administrarii, administratorii IT incearca sa gestioneze identitatile cat mai centralizat posibil si sa se uneasca managementul identitatii cu politicile de autentificare si autorizare .Din de vedere istoric, mediile Linux au avut o perioada foarte dificilain stabilirea acestui management centralizat. Exista un numar de diferite protocoale (cum ar fi NIS si Kerberos ) care definesc domenii, in timp ce alte aplicatii stocheaza date ( cum ar fi LDAP ) si inca altele gestioneaza accesul ( cum ar fi sudo ). Nici unul dintre aceste aplicatii comunica unele cu altele sau folosesc aceleasi instrumente de management. Fiecare cerere a trebuit sa fie administrata separat si a trebuit sa fie administrata local . Singura modalitate de a obtine o politica de identitate consistenta a fost de a copia fisiere de configurare manual sau a incerca sa dezvolte o aplicatie de pentru a gestiona proprietatea, identitatile si politicile .  
Scopul Managementului Identitatii este de a simplifica cheltuielile administrative. Utilizatorii, masinile, serviciile, si politicile sunt toate configurate intr-un singur loc , utilizand aceleasi instrumente . Deoarece **IdM** creeaza un domeniu , mai multe masini pot folosi toate aceeasi configuratie si aceleasi resurse pur si simplu prin aderarea la domeniu . Utilizatorii trebuie doar sa se autentifice o singura data, si administratorii trebuie doar sa administreze un singur cont de utilizator.

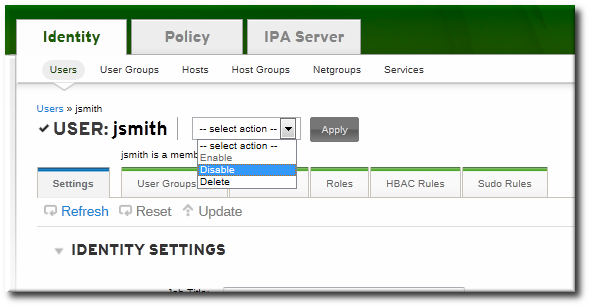
**2)Activarea si dezactivarea conturilor**

Conturile de utilizator pot fiactivate sau dezactivate. Un utilizator dezactivat nu se poate conecta in IdM sau la serviciile sale conexe (cum ar fi Kerberos) iar acesta nu poate efectua nicio sarcina. Cu toate acestea, contul de utilizator existaincain Identity Management si toate informatiile asociate raman neschimbate.

Orice conexiuneexistenta ramanevalabila pana ce Kerberos TGT si alte „bilete” expira. Odata ce biletul expira, utilizatorul nu poate reinnoi biletul.

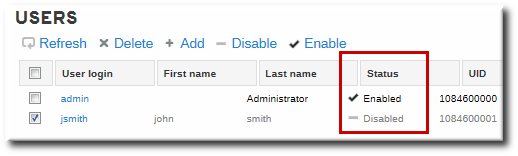
Utilizatorii potfidezactivatidin lista de utilizatori complet prin selectarea casetelor de utilizatori dorite si apoi debifand caseta dorita.

Un cont de utilizator poate fi dezactivat din pagina de intrare individuala a utilizatorului.   
1. Se deschide fila de Identitate si apoi se selecteaza subfila utilizatori.   
2. Se face click pe numele utilizatorului pentru a dezactiva sau a activa.   
3. In meniul vertical de actiuni se face click pe butonul Disable.



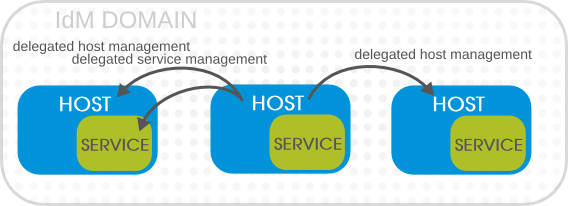
4.Click pe butonul Accept.

Atunci cand un cont de utilizator este dezactivat, aceasta este semnificat de un minus (-) pentru statutul de utilizator in lista de utilizatori si prin numele de utilizator pe pagina de start. In plus, textul pentru utilizator este de culoare gri (pentru a arata ca este inactiv) in loc de negru.



**3)Delegarea accesului Hosturilor si Serviciilor**

Un key tab este un fisier care contine perechi de directori Kerberos si chei criptate(acestea sunt derivate din parola Kerberos).Gestiunea inseamna capabilitatea de a prelua un keytab si o certificare pentru o alta gazda sau serviciu. Fiecare gazda si servici are o intrare **managedby** care enumera ceea ce gazduieste sau servicii ce pot gestiona. In mod implicit, o gazda in sine si toate serviciile sale poate gestiona. De asemenea, este posibil sa i se permita unei gazde sa gestioneze alte gazde, sau servicii pe alte gazde, prin actualizarea delegatiilor adecvate sau furnizand o intrare **managedby** adecvata.   
Un serviciu de IdM poate fi gestionat de la orice gazda IDM, atata timp cat gazda a fost de acord, sau i-a fost delegata, permisiunea de a accesa acest serviciu. De asemenea, gazdelor le pot fi delegate permisiuni pentru alte gazde in cadrul domeniului.



**4)Delegarea managementului serviciilor**

Unei gazde ii este delegat controlul asupra mai multor servicii folosind Comanda service-add-Host. EXISTA DOUA parti la delegarea serviciului: specificarea principalului si identificarea gazdelor(intr-o lista separata prin virgula) cu control:

**# ipa service-add-host *principal* --hosts=*hostnames***

Exemplu:

**# ipa service-add-host http/web.example.com --hosts=client1.example.com**

Odata ce gazdei ii este delegata o autoritate,gazda principala poate fi folosita pentru a gestiona serviciul:

**# kinit -kt /etc/krb5.keytab host/`hostname`**

**# ipa-getkeytab -s `hostname` -k /tmp/test.keytab -p http/web.example.com**

**Keytab successfully retrieved and stored in: /tmp/test.keytab**

Pentru acrea un bilet pentru acest serviciu, se creeaza o cerere de certificat pe gazda cu autoritatea delegata si se foloseste comanda CERT-request pentru a crea o intrare de serviciu apoi se incarca informatia de certificare:

**# ipa cert-request --add --principal=http/web.example.com web.csr**

**Certificate: MIICETCCAXqgA...[snip]**

**Subject: CN=web.example.com,O=EXAMPLE.COM**

**Issuer: CN=EXAMPLE.COM Certificate Authority**

**Not Before: Tue Feb 08 18:51:51 2011 UTC**

**Not After: Mon Feb 08 18:51:51 2016 UTC**

**Fingerprint (MD5): c1:46:8b:29:51:a6:4c:11:cd:81:cb:9d:7c:5e:84:d5**

**Fingerprint (SHA1):**

**01:43:bc:fa:b9:d8:30:35:ee:b6:54:dd:a4:e7:d2:11:b1:9d:bc:38**

**Serial number: 1005**

**5)Delegarea managementului gazdelor**

Gazdele sunt autoritati delegate asupra altor gazde prin comanda **host-add-managedby**. Acest lucru creeaza o intrare **managedby**. Odata ce intrarea **managedby** este creata, gazda poate prelua un keytab pentru gazda a carei autoritate trebuie delegata.

1. Utilizatorul se autentifica ca administrator

**# kinit admin**

1. Apoi este adaugata intrarea the *managedby* . Spre exemplu in exemplu este delegate autoritatea clientului 2 peste clientul1.

**# ipa host-add-managedby client2.example.com --hosts=client1.example.com**

1. Obtine un bilet ca client1 gazda si apoi preia un keytab pentru Client2:
2. **# kinit -kt /etc/krb5.keytab host/`hostname**`
3. **# ipa-getkeytab -s `hostname` -k /tmp/client2.keytab -p host/client2.example.com**

**Keytab successfully retrieved and stored in: /tmp/client2.keytab**

**6)Accesarea serviciilor delegate**

Atat pentru servicii cat si pentru gazde, in cazul in care un client a delegat autoritatea, se poate obtine un key tab pentru principalul de pe masina locala. Pentru servicii, acesta are formatul:service/hostname@REALM. Pentru gazde, serviciul este gazda.   
Cukinit, se foloseste optiunea-k pentru a incarca un keytab si optiunea-t pentru a specifica keytab.

Spre exemplu,pentru a accesa o gazda:

**# kinit -kt /etc/krb5.keytab host/ipa.example.com@EXAMPLE.COM**

Sau, pentru a accesa un serviciu:

**# kinit -kt /etc/httpd/conf/krb5.keytab** [**http/ipa.example.com@EXAMPLE.COM**](mailto:http/ipa.example.com@EXAMPLE.COM)

**1.4.3.Windows**

**a)SID(Security Identifier)**

Un **principal** in securitatea informatica este o entitate care poate fi„autentificata”de catre un sistem informatic sau de retea. Acesta este mentionat ca un principal de securitate in Java si in literatura Microsoft.

In cadrul liniei de Microsoft Windows NT de sisteme de operare, un identificator de securitate (SID frecvent abreviat) este un identificator unic, invariabill al unui utilizator, a unui grup de utilizatori,sau a unui **principal** de securitate. Un principal de securitate are un singur SID pe durata intregii vieti si toate proprietatile in calitate de **principal**, inclusiv numele sau, sunt asociate cu SID. Acest design permite unui principal sa fie redenumit (de exemplu, de la "John" la "Jane"), fara a afecta atributele de securitate ale obiectelor care se refera la director.

Windows garanteaza sau refuza accesul si privilegiile la resurse bazate pe listele de control al accesului ( ACL-uri ) , care folosesc un AIN pentru a identifica in mod unic utilizatorii si adeziunea lor de grup . Atunci cand un utilizator se utentifica la un calculator, un jeton de acces este generat, care contine SID de utilizator si de grup si nivelul de privilegii de utilizator . Atunci cand un utilizator cere accesul la o resursa, token-ul de acces este verificat impotriva ACL pentru a permite sau pentru a refuza o actiune speciala pe un anumit obiect .  
SID sunt utile pentru depanarea problemelor cu audituri de securitate, servere Windows si migratii de/la domenii .  
Formatul unui SID poate fi ilustrat folosind urmatorul exemplu : " S -1-5-21--3623811015 3361044348-30300820-1013 " ;  
S 1 5 21-3623811015-3361044348-30300820 1013  
Sirul este un SID . Domeniul sau identificatorul de calculatoarelor locale,un ID relativ ( RID ) . Orice grup sau utilizator care nu este creat in mod implicit va avea o identitate relativa de 1000 sau mai mare .  
Posibile valori de identificare a autoritatii sunt :  
    0 - Autoritatea Null  
    1 - Autoritatea Mondial  
    2 - Autoritatea locala  
    3 - Autoritatea Creator  
    4 - Autoritatea Non - unic  
    5 - Autoritatea NT  
    9 - Resurse Autoritatea de Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S** | **1** | **5** | **21-3623811015-3361044348-30300820** | **1013** |
| The string is a SID. | The revision level (the version of the SID specification). | The identifier authority value. | Domain or local computer identifier | A [Relative ID](https://en.wikipedia.org/wiki/Relative_ID) (RID). Any group or user that is not created by default will have a Relative ID of 1000 or greater. |

**b)JETONUL DE ACCES LA WINDOWS**

Un token de acces este un obiect care incapsuleaza identitatea de securitate a unui proces sau fir de executie(thread) . Un token este folosit pentru a lua decizii de securitate si pentru a stoca informatii falsificate despre o entitate de sistem. In timp ce un token este folosit in general pentru a reprezenta informatii numai de securitate, acesta este capabil de a detine date suplimentare de forma care pot fi atasate in timp ce token-ul este creat . Token-urile pot fi reproduse fara privilegii speciale, de exemplu pentru a crea un nou token, cu niveluri mai scazute de drepturi de acces pentru a restrictiona accesul unei cereri lansate. Un token de acces este folosit de Windows atunci cand un proces sau fir incearca sa interactioneze cu obiecte care au descriptori de securitate ( obiecte securizabile ) . Un token de acces este reprezentat de obiectul de sistem de tip Token .  
Un token de acces este generat de serviciul de conectare atunci cand un utilizator se conecteaza la sistemsi acreditarile furnizate de utilizator sunt autentificate in baza de date de autentificare . Baza de date de autentificare contine informatii de acreditare necesare pentru a construi token-ul initial pentru sesiunea de conectare, inclusiv id-ul de utilizare al acestuia, id-ul de grup primar, toate celelalte grupuri din care face parte, si alte informatii . Jetonul este atasat la procesul initial creatinsesiunea de utilizator si mostenit de procesele ulterioare create de procesul initial. Ori de cate ori un astfel de proces se deschide p cale catre orice resursa care are controlul accesului activat, Windows mediaza legatura intre datele din descriptorul de securitate al obiectului tinta cu continutul actual al token-ul de acces efectiv. Rezultatul acestei evaluari de verificare a accesului este o indicatie cu privire la ce tip de acces este permis si, daca accesul este permis, ce operatiuni (citire/ scriere / modificare, etc ) aplicatiei in asteptare ii este permis sa efectueze.

**Capitolul II:Functiile de securitate ale Win32API(Otelea Stefan)**

**2.1.Notiuni generale de Win32API**

Acronimul API este o abreviere a Application Programming Interface. Asadar Windows API (sau Win32 API) este un set de functii oferite de sistemul de operare Windows pentru manipularea resurselor calculatorului. Orice sistem de operare ofera (sau exporta) un set de astfel de functii, pentru a fi utilizate de programatori in dezvoltarea de aplicatii specifice acelui sistem de operare. Denumirea de Win32 API mai este folosita uneori pentru a marca diferenta dintre sistemele de operare Windows pe 16 biti (Windows 3.X) si sistemele de operare Windows pe 32 de biti (Windows 9X, Windows NT, Windows 2000, Windows XP).

Functiile din interfata Win32 API sunt implementate in urmatoarele trei biblioteci: **user32.dll**, **kernel32.dll** si **gdi32.dll**. Ele sunt exportate si se pot folosi de catre orice program, cu conditia includerii fisierului header corespunzator (care este**Windows.h**) si a editarii legaturilor programului cu fisierul de exporturi corespunzator (**user32.lib**, **kernel32.lib** sau/si **gdi32.lib**). Nu va speriati, aceste trei biblioteci sunt trecute in mod automat in lista de module a editorului de legaturi, in orice proiect generat de mediul Visual C++.

Descrierea interfetei Win32 API poate fi gasita in help-ul on-line la indexul Platform SDK. Functiile din interfata Win32 API sunt impartite in mai multe sectiuni. Dintre acestea, cele mai importante sunt User Interface Services si Windows Base Services.

**2.2.Protectie si Siguranta in functionare**

**2.2.1.Accesibilitate**

Win32API ofera un set de caracteristici de accesibilitate, care fac mai usor accesul inclusiv pentru persoanele cu handicap. Aceasta privire de ansamblu descrie caracteristicile de accesibilitate furnizate, functiile si structurile Win32 pentru a controla aceste caracteristici.

Exista doua categorii de caracteristici de accesibilitate, care corespund setarilor disponibile pentru utilizatori prin Ease of Acces Centerin Panoul de control :  
  
    ->Parametrii de accesibilitate. Cand sunt setati, acesti parametri indica faptul ca cererile ar trebui sa -si schimbe comportamentul implicit. Cererile pot verifica starea unui parametru de accesibilitate pentru a determina daca utilizatorul doreste un comportament special, care poate fi furnizatintr-un mod specific aplicatiei. De exemplu , parametrul ShowSounds indica faptul ca o aplicatie care utilizeaza in mod obisnuit sunet pentru a transmite informatii importante ar trebui sa ofere, de asemenea, informatiile vizuale .  
    ->Built - in caracteristici de accesibilitate. Aceste caracteristici sunt construite in sistem sau furnizate ca o extensie a sistemului. Acestea modifica modul in care utilizatorul asigura tastatura si mouse-ul la intrarea in computer. Cand este activata , functionalitatea lor este disponibil indiferent de aplicatii sunt inchise . Un exemplu este un filtru de tastatura care face mai usor pentru utilizatorii cu deficiente de miscare pentru a introduce combinatii de taste , cum ar fi CTRL + ALT + DEL .

**Considerente de securitate pentru Tehnologii de Asistenta**

Tehnologiile de asistenta sunt aplicatii care ruleaza pe desktop-ul Windows-ului si ajuta utilizatorii sainteractioneze cu sistemul de operare si alte aplicatii care ruleaza pe calculator, inclusiv aplicatii in noul Windows UI(User Interface). Aplicatiile de tehnologie de asistare lucreaza prin preluarea de informatii de la sistemul de operare si de la alte aplicatii, apoi prezinta informatiile intr -un mod care este accesibil pentru utilizator. O cerere de tehnologii de asistenta poate, de asemenea, "conduce" sistemul de operare si alte aplicatii bazate pe intrarea de la utilizator .  
Natura aplicatiilor tehnologiei de asistenta impune ca acestea sa treaca granitele procesului , si ca acestea sa aiba acces la procese care se executa la un nivel mai ridicat de integritate ( IL ) decat ei insisi . ( O cerere de tehnologie de asistenta ruleaza la mediu IL ). De exemplu , atunci cand utilizatorul incearca sa efectueze o sarcina care necesita privilegii administrative , Windows prezinta o caseta de dialog in cere utilizatorului acordul de a continua . Aceasta caseta de dialog se executa la o IL(Integration Level) mai mare pentru a se proteja de comunicatiile inter- proces, astfel ca software-ul rau intentionat nu poate simula intrarile de utilizator. In mod similar, ecranul de conectare la desktop ruleaza la o IL mai mare pentru a preveni accesarea de catre alte procese .  
Aplicatiile tehnologice de asistenta de obicei, necesita accesul la elementele UI protejate ale sistemului, sau la alte procese care ar putea rula la un nivel mai mare de privilegiu. Prin urmare , cererile de tehnologie de asistare trebuie sa fie de incredere, si trebuie saruleze cu privilegii speciale .  
Pentru a avea acces la procesele de IL mai mari , o cerere de tehnologie de asistenta trebuie sa stabileasca steagul UIAccess in manifestarea aplicatiei.  
In plus, pentru a avea acces la procesele de IL mai mari, o cerere de tehnologie de asistenta cu UIAccess poate rula ca cel mai de sus in orice moment, ceea ce inseamna ca o cerere de tehnologii de asistenta poate fi vizibila si disponibila ori de cate ori utilizatorul are nevoie de ea .

**2.2.2.Autentificare**

Paginile de referinta de autentificare contin descrieri detaliate ale functiilor de autentificare Microsoft, interfete, obiecte, structuri si alte elemente de programare. Aceste subiecte furnizeaza informatii despre elementele de programare utilizate pentru a stabili identitatea utilizatorilor si serviciile.Dupa ce identitatea este stabilita, acesta poate fi utilizat pentru a incerca accesul la un singur calculator sau a resurselor pe o retea de calculatoare.

**Functii de autentificare Win32**

1.SSPI Functions

Functiile SSPI(Security Support Provider Interface) se impart inurmatoarele categoriimajore:

**Package Management**  
Functiile de administrare a pachetelor SSPI initiaza un pachet de securitate, enumera pachete disponibile si organizaeaza atributele unui pachet de securitate. Urmatoarele functii SSPI furnizeze servicii de administrare pentru pachetele de securitate.

|  |  |
| --- | --- |
| **Functia** | **Descriere** |
| [**EnumerateSecurityPackages**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375397%28v=vs.85%29.aspx) | Listseaza pachetele de securitate si atributiile lor. |
| [**InitSecurityInterface**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa376103%28v=vs.85%29.aspx) | Recupereaza un pointer la unul din furnizorii de suport al securitatii (SSP) |
| [**QuerySecurityPackageInfo**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa379359%28v=vs.85%29.aspx) | Extrage informatii despre un pachet de securitate specificat. Aceste informatii includ limitele privind dimensiunile de informatii de autentificare, scrisorile de acreditare, si contextele. |

**Managementul acreditarii**

Functiile de gestiune a acreditarii SSPI ofera o referinta la un obiect de securitate opac, pentru accesarea unui director. Obiectul de securitate este opac, deoarece aplicatia are acces doar la „manerul”si nu la continutul real al structurii.   
  
Toate trimiterile la continutul unui context de acreditare sunt prin intermediul „manerului” obiectului iar pachetul de securitate face referire la„maner” pentru a accesa specificul de acreditare. Un „maner” de acreditare este o valoare pe 64 de biti intre {0x00000000, 0x00000000} si {0xFFFFFFFF, 0xFFFFFFFE}.

Aplicatiile utilizeaza acreditarile cu functii de gestiune al contextului pentru a crea un context de securitate.   
  
Functii de management al acreditarii, de asemenea, genereaza„manere” de acreditare si solicita atributele de acreditare. In prezent, numele asociat cu o acreditare este singurul atribut care pot fi solicitat.   
  
Urmatoarele functii sunt utilizate cu management de acreditari:

|  |  |
| --- | --- |
| **Functie** | **Descriere** |
| [**AcquireCredentialsHandle (General)**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374712%28v=vs.85%29.aspx) | Adauga un “handle” la cele deja existente pentru a face referire la un director. |
| [**ExportSecurityContext**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375409%28v=vs.85%29.aspx) | “Exporta” un context de securitate intr-un anumit buffer. |
| [**FreeCredentialsHandle**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375417%28v=vs.85%29.aspx) | Elibereaza un “handle” de acreditare si resursele associate acestuia. |
| [**ImportSecurityContext**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa375502%28v=vs.85%29.aspx) | Importa un context de securitate exportat prin utilizarea ExportSecurityContext. |
| [**QueryCredentialsAttributes**](http://msdn.microsoft.com/en-us/library/windows/desktop/aa379342%28v=vs.85%29.aspx) | Preia atributele unei acreditari, cum ar fi numele asociat cu acreditarearespectiva |

**Context Management**

Functiile de management al contextului SSPI creeazasi de folosesc contexte de securitate.   
Intr-o legatura de comunicare, client-ul si server-ul coopereaza pentru a crea un context de securitate comuna.Atat clientul cat si server-ul,folosesc contextul de securitate cu functii de suport al mesajului pentru a asigura integritatea mesajului si a vietii private in timpul conexiunii.   
Contextele de securitate sunt obiecte de securitate opace. Informatiile in contextul de securitate nu sunt disponibile la cerere. Functiile de conducere context creeaza si folosesc „handle”-ul contextului si pachetul de securitate indica „handle”-ul contextului pentru a accesa continutul de securitate.   
  
Un context „handle” este o valoare pe 64 de biti intre {0x00000000, 0x00000000} si {0xFFFFFFFF, 0xFFFFFFFE}.   
  
Urmatoarele functii sunt utilizate cu management context.

|  |  |
| --- | --- |
| **Function** | **Description** |
| [**AcceptSecurityContext (General)**](http://msdn.microsoft.com/en-us/library/aa374703%28v=vs.85%29.aspx) | Folosita de server pentru a crea [*security context*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_security_context_gly) bazat pe un mesaj opac primit de la client. |
| [**ApplyControlToken**](http://msdn.microsoft.com/en-us/library/aa374724%28v=vs.85%29.aspx) | Aplica un mesaj de securitate suplimentar la un context de securitate deja existent. |
| [**CompleteAuthToken**](http://msdn.microsoft.com/en-us/library/aa374764%28v=vs.85%29.aspx) | Completeaza un token de autentificare. Aceasta functie este folosita de protocoale, cum ar fi DCE, care necesita revizuirea informatiei de securitate dupa ce aplicatia de transport a actualizat anumiti parametric a-i mesajelor. |
| [**DeleteSecurityContext**](http://msdn.microsoft.com/en-us/library/aa375354%28v=vs.85%29.aspx) | Elibereaza contextul de securitate si resursele asociate. |
| [**FreeContextBuffer**](http://msdn.microsoft.com/en-us/library/aa375416%28v=vs.85%29.aspx) | Elibereaza buffer-ul de memorie alocat de pachetul de securitate. |
| [**ImpersonateSecurityContext**](http://msdn.microsoft.com/en-us/library/aa375497%28v=vs.85%29.aspx) | Depersonalizaeaza contextul de securitate pentru ca sistemul sa il poata vedea ca pe un client. |
| [**InitializeSecurityContext (General)**](http://msdn.microsoft.com/en-us/library/aa375506%28v=vs.85%29.aspx) | Folosita de client pentru a initia contextul de securitate generand un mesaj opacce trebuie sa ajunga la server. |
| [**QueryContextAttributes (General)**](http://msdn.microsoft.com/en-us/library/aa379326%28v=vs.85%29.aspx) | Ofera dreptul unei cereri de transport sa interogheze un “*security package”*  in scopul obtierii unor atribute a unui context de securitate. |
| [**QuerySecurityContextToken**](http://msdn.microsoft.com/en-us/library/aa379355%28v=vs.85%29.aspx) | Obtine un “[*access token*](http://msdn.microsoft.com/en-us/library/ms721532%28v=vs.85%29.aspx#_security_access_token_gly)” pentru un “[*security context*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_security_context_gly)” al unui client pe care il utilizeaza direct. |
| [**SetContextAttributes**](http://msdn.microsoft.com/en-us/library/aa380137%28v=vs.85%29.aspx) | Ofera dreptul unei cereri de transport sa seteze *attribute*  pentru un “security [*context*](http://msdn.microsoft.com/en-us/library/ms721572%28v=vs.85%29.aspx#_security_context_gly)“ care la randul lui e folosit de un“[*security package*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_security_package_gly)”. Aceasta functie este folosita doar de pachetul de securitate Schannel. |
| [**RevertSecurityContext**](http://msdn.microsoft.com/en-us/library/aa379446%28v=vs.85%29.aspx) | Da voie unui [*security package*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_security_package_gly)sa intrerupa depersonificarea apelantului si ii restaureaza propriul “[*security context*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_security_context_gly)”. |

**Message support**

Functiile de suport al mesajelor SSPI permit unei cereri sa transmitasi primi mesaje pentru a cripta si decripta mesaje. Aceste functii lucreaza cu unul sau mai multe buffer-e care contin un mesaj si un context de securitate creat de catre functiile de administrare a contextului.Comportamentul functiilor difera in functie de conexiune, datagrame, sau fluxul ce este in uz.  
Depersonifica contextul de securitate pentru a aparea ca si client al sistemului .

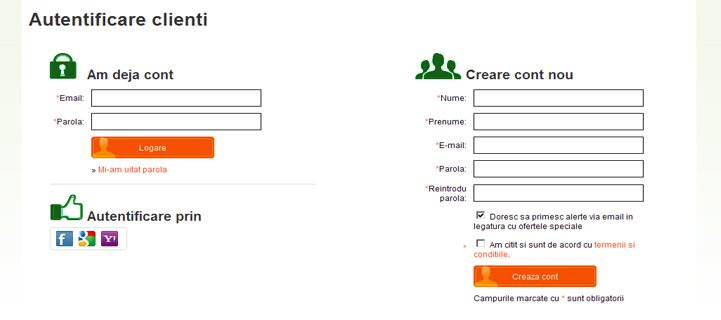
**InitializeSecurityContext ( General** )  
  
Folosite de un client de a initia un context de securitate prin generarea unui mesaj opac cu scopul de a ajunge la un server .  
**QueryContextAttributes ( General )**  
Permite o cerere de transport pentru a interoga un pachet de securitate pentru anumite atribute ale unui context de securitate .  
QuerySecurityContextToken

|  |  |
| --- | --- |
| **Function** | **Description** |
| [**DecryptMessage (General)**](http://msdn.microsoft.com/en-us/library/aa375211%28v=vs.85%29.aspx) | Decripteaza si cripteaza mesaje folosind[*session key*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_session_key_gly) dintr-un[*security context*](http://msdn.microsoft.com/en-us/library/ms721625%28v=vs.85%29.aspx#_security_security_context_gly). |
| [**EncryptMessage (General)**](http://msdn.microsoft.com/en-us/library/aa375378%28v=vs.85%29.aspx) | Cripteaza mesaje folosind session key dintr-un security context. |
| [**MakeSignature**](http://msdn.microsoft.com/en-us/library/aa378736%28v=vs.85%29.aspx) | Generateaza mesaje de control, de asemenea include informatii secventiale pentru a preveni pierderea lor sau insertia altora. |
| [**VerifySignature**](http://msdn.microsoft.com/en-us/library/aa380540%28v=vs.85%29.aspx) | Verifica semnatura mesajelor primite folosind[**MakeSignature**](http://msdn.microsoft.com/en-us/library/aa378736%28v=vs.85%29.aspx). |

## LSA Logon Functions

Urmatoarele functii de autentificare[Local Security Authority](http://msdn.microsoft.com/en-us/library/ms721592%28v=vs.85%29.aspx#_security_local_security_authority_gly) (LSA) autentifica si utilizatorii iar ei furnizaeaza sesiune de informatii de tip logon.

|  |  |
| --- | --- |
| **Functie** | **Descriere** |
| [**LsaCallAuthenticationPackage**](http://msdn.microsoft.com/en-us/library/aa378261%28v=vs.85%29.aspx) | Primeste un serviciu specific pachetului de la un pachet de autentificare. |
| [**LsaConnectUntrusted**](http://msdn.microsoft.com/en-us/library/aa378265%28v=vs.85%29.aspx) | Restabileste o conexiune nesigura la LSA. |
| [**LsaDeregisterLogonProcess**](http://msdn.microsoft.com/en-us/library/aa378269%28v=vs.85%29.aspx) | Deconecteaza utilizatorul de la LSA si elibereaza resursele allocate apelantului. |
| [**LsaEnumerateLogonSessions**](http://msdn.microsoft.com/en-us/library/aa378275%28v=vs.85%29.aspx) | Recupereaza[*locally unique identifiers*](http://msdn.microsoft.com/en-us/library/ms721592%28v=vs.85%29.aspx#_security_locally_unique_identifier_gly) (LUIDs) de la sesiunile existente de logon. |
| [**LsaFreeReturnBuffer**](http://msdn.microsoft.com/en-us/library/aa378279%28v=vs.85%29.aspx) | Elibereaza memoria alocata buffer-ului returnand-o apelantului. |
| [**LsaGetLogonSessionData**](http://msdn.microsoft.com/en-us/library/aa378290%28v=vs.85%29.aspx) | Recupereaza informatia despre o sesiune de logon specifica. |
| [**LsaLogonUser**](http://msdn.microsoft.com/en-us/library/aa378292%28v=vs.85%29.aspx) | Autentifica user logon data inlocuind informatia de acreditare deja existenta. Daca resueste, se creeaza o noua sesiune de logon ce returneaza token-ul utilizatorului. |
| [**LsaLookupAuthenticationPackage**](http://msdn.microsoft.com/en-us/library/aa378297%28v=vs.85%29.aspx) | Obtine identificatorul unic dintr-un pachet de autenticare. |
| [**LsaQueryDomainInformationPolicy**](http://msdn.microsoft.com/en-us/library/aa378303%28v=vs.85%29.aspx) | Recupereaza informatii despre un anumit domeniu de la obiectul[**Policy**](http://msdn.microsoft.com/en-us/library/ms721912%28v=vs.85%29.aspx). |
| [**LsaQueryForestTrustInformation**](http://msdn.microsoft.com/en-us/library/aa378308%28v=vs.85%29.aspx) | Recupereaza“forest trust information” de la[*Local Authority*](http://msdn.microsoft.com/en-us/library/ms721592%28v=vs.85%29.aspx#_security_local_security_authority_gly) [**TrustedDomain**](http://msdn.microsoft.com/en-us/library/ms722464%28v=vs.85%29.aspx). |
| [**LsaRegisterLogonProcess**](http://msdn.microsoft.com/en-us/library/aa378318%28v=vs.85%29.aspx) | Stabileste o conexiune la server-ul LSA si verifica daca apelantul foloseste o cerere logon. |
| [**LsaSetDomainInformationPolicy**](http://msdn.microsoft.com/en-us/library/aa378322%28v=vs.85%29.aspx) | Sets domain information to the [**Policy**](http://msdn.microsoft.com/en-us/library/ms721912%28v=vs.85%29.aspx) object. |
| [**LsaSetForestTrustInformation**](http://msdn.microsoft.com/en-us/library/aa378324%28v=vs.85%29.aspx) | sets the forest trust information for a specified [*Local Security Authority*](http://msdn.microsoft.com/en-us/library/ms721592%28v=vs.85%29.aspx#_security_local_security_authority_gly) [**TrustedDomain**](http://msdn.microsoft.com/en-us/library/ms722464%28v=vs.85%29.aspx) object. |

****

**2.2.3.Autorizatie**

Paginile de referinta de autorizare contin descrieri detaliate ale Microsoft, functiile de autorizare, interfetele, obiecte, structuri, si alte elemente de programare.Aceste pagini includ descrieri de referinta ale API pentru a lucra cu controale de acces, inclusiv editorii de control al accesului.

Dupa ce o persoana, un program sau un calculator a fost identificat si autentificat cu succes, atunci trebuie sa se stabileasca ce resurse informationale li se permite sa acceseze si ce actiuni le va fi permis sa efectueze (run , vedere, creare, stergere, schimbare ) . Aceasta se numeste **autorizatie** . Autorizatia de a accesa informatii si alte servicii de calcul incepe cu politicile si procedurile administrative. Politicile prescriu ceea ce serviciile de informare si de calcul pot accesa, de catre cine si in ce conditii. Mecanismele de control al accesului sunt configurate pentru a pune in aplicare aceste politici . Sistemele de calcul diferite, sunt echipate cu diferite tipuri de mecanisme , unele de control al accesului ce pot oferi chiar posibilitatea de a alege diferite mecanisme de control al accesului . Mecanismul de control al accesului se va baza pe una dintre cele trei abordari asupra controlului accesului sau poate fi derivat dintr- o combinatie a celor trei abordari .  
Abordarea **non- discretionar** consolideaza toate controalele de acces sub o administrare centralizata. Accesul la informatii si alte resurse se bazeaza de obicei pe functia persoanelor fizice ( rol ) in cadrul organizatiei sau sarcinile pe care individul trebuie sa le efectueze . Abordarea discretionar ofera creatorului sau proprietarului resursei de informatii capacitatea de a controla accesul la aceste resurse .In abordarea controlului de acces obligatorie , accesul este acordat sau refuzat bazandu-se pe clasificarea de securitate atribuita de resursele de informatii .  
Exemple de mecanisme de control al accesului comun utilizate in prezent includ controlul accesului bazat pe roluri disponibil in mai multe sisteme avansate de gestiune a bazelor de date prevazute in sistemele de operare UNIX si Windows , Group Policy Objects prevazute in sistemele de retea Windows , Kerberos , RADIUS , TACACS , si simple liste de acces folosite in multe firewall-uri si routere .  
Pentru a fi eficiente, politicile si alte controale de securitate trebuie sa fie executorii si incurajatoare . Politici eficiente se asigura ca oamenii sunt trasi la raspundere pentru actiunile lor. Toate esuarile si incercarile de autentificare de succes trebuie sa fie conectate, si toate caile de acces la informatie trebuie sa tina cont de un anumit tip de audit .  
De asemenea, principiul necesitatii de a cunoaste trebuie sa fie valabil atunci cand vorbim despre de controlul accesului. Principiul necesitatii de a cunoaste confera drepturi de acces unei persoane pentru a indeplini functiile specifice locului de munca .Chiar daca doi angajati din diferite departamente au un profil secret , ei trebuie beneficieze de informatii comune pentru a le putea interschimba. In cadrul principiului nevoii de a cunoaste , administratorii de retea trebuie sa acorde salariatului anumite privilegii pentru a preveni accesul angajatilor si de a accesa mai mult decat ceea ce ar trebui sa acceseze .

**2.2.4.Criptografie**

Securitatea informatiilor utilizeaza criptografie pentru a transforma informatii utilizabile intr-o forma care le face inutilizabile de catre oricine altcineva decat un utilizator autorizat; acest proces este numit **criptare.**Informatiile care au fost criptate (inutilizabile) pot fi transformate inapoi in forma sa originala utilizabila de catre un utilizator autorizat , care are cheia de criptare , prin procesul de **decriptare**. Criptografia este folosita in securitatea informatiilor pentru a proteja informatiile impotriva divulgarii neautorizate sau accidentale, in timp ce informatiile se aflain tranzit ( electronic sau fizic ), si in timp ce informatiile se aflain depozit .  
Criptografia ofera securitate informatiilordar si alte aplicatii utile cum ar fi metode imbunatatite de autentificare , semnaturi digitale, dar si managementul retelelor de comunicatii criptate Aplicatii mai putin sigure si mai vechi, cum ar fi Telnet si FTP sunt incet inlocuite cu aplicatii mai sigure, cum ar fi **ssh** care foloseste retele de comunicatii criptate . Comunicatiile fara fir pot fi criptate folosind protocoale cum ar fi WPA/WPA2 sau mai vechi ( si mai putin sigure ) WEP. Comunicatii cu fir ( cum ar fi ITU - T G.hn ) sunt asigurate folosind AES pentru criptare si X.1035 pentru autentificarea si schimbul de chei. Aplicatii software, cum ar fi GnuPG sau PGP poate fi folosite pentru a cripta fisierele de date si e-mail .  
Criptografia poate introduce probleme de securitate atunci cand nu este pusain aplicare in mod corect. Solutii criptografice trebuie sa fie puse in aplicare prin solutii acceptate care au fost supuse riguros evaluarii de catre expertii in criptografie. Lungimea si puterea de cheia de criptare este, de asemenea, un aspect important . O cheie care este slaba sau prea scurt va produce criptare slaba. Cheile folosite pentru criptarea si decriptarea trebuie sa fie protejate cu acelasi grad de rigoare ca si orice alte informatii confidentiale . Ele trebuie sa fie protejate impotriva divulgarii si distrugerii neautorizatesi trebuie sa fie disponibile atunci cand este nevoie. Infrastructurile de chei publice ( PKI )sunt solutii de abordare a multe dintre problemele cu care se confrunta gestionarea cheilor .

**Functii de baza de criptografie**

Functiile criptografice de baza asigura mijloacele cele mai flexibile de dezvoltare a aplicatiilor de criptografie . Toate comunicatiile cu un furnizor de servicii criptografice ( CSP ) au loc prin intermediul acestor functii .  
Un CSP este un modul independent care efectueaza toate operatiunile criptografice . Cel putin un CSP este necesar cu fiecare aplicatie care utilizeaza functii criptografice. O singura cerere poate folosi, ocazional, mai mult de un CSP.  
  
Daca se utilizeaza mai mult de un CSP , cea folosita la momentul curent poate fi specificata in apelurile de functii criptografice CryptoAPI. Un CSP, furnizorulcriptografic de baza al Microsoft, este la pachet cu CryptoAPI. Acest CSP este folosit ca un furnizor implicit de multe dintre functiile CryptoAPI daca nicio alta CSP nu este specificata .  
Fiecare CSP ofera o punere in aplicare diferita a sprijinului criptografic oferit de CryptoAPI. Unele ofera algoritmi de criptare puternici ; altele contin componente hardware , cum ar fi smart card-uri . In plus , unele CSP-uri , ocazional, pot comunica direct cu utilizatorii, cum ar fi atunci cand semnaturile digitale sunt efectuate prin utilizarea unei cheie privata cu semnatura utilizatorului .  
  
Functii criptografice de baza sunt in urmatoarele grupe mari :  
  
   -> Functii furnizor de servicii:**CryptAquireContext,CryptContextAddREf,CryptEnumProviders,CrypGetProvParam**  
    ->Functii de generare a cheii si de schimb:**CryptDEriveKey,CryptDEstroyKey,CryptDuplicateKey,CryptGenKey**  
    ->Functii de codare si decodare a obiecteleor:**CryptDecodeObject,CryptEncodeObject**  
    ->Functiide criptare si decriptare a datelor:**CryptDecrypt,CryptEncrypt,CryptProtectData**

    ->Functii pentru semnatura digitala:**CryptCreateHash,CryptDestroyHash,CryptDuplicateHash**

****

**2.2.5.Managementul parolei prin MS-CHAP(Microsoft Challenge-Handshake Authentication Protocol)**

**Functiile MS-CHAP**

1)MS-CHAPSrvChangePassword-schimba parola curenta a contului unui utilizator.

**Sintaxa**

**C++**

**extern WINADVAPI DWORD WINAPI MSChapSrvChangePassword(**

**\_In\_  PWSTR ServerName,**

**\_In\_  PWSTR UserName,**

**\_In\_  BOOLEAN LmOldPresent,**

**\_In\_  PLM\_OWF\_PASSWORD LmOldOwfPassword,**

**\_In\_  PLM\_OWF\_PASSWORD LmNewOwfPassword,**

**\_In\_  PNT\_OWF\_PASSWORD NtOldOwfPassword,**

**\_In\_  PNT\_OWF\_PASSWORD NtNewOwfPassword**

**);**

**Parametri**

ServerName [in]

Un pointer la un sir “null-terminated” Unicode indica prin “ Universal Naming Convention (UNC)” numele serverului pe care sa opereze. Daca acest parametru este**NULL**, functia opereaza pe computer-ul local.

UserName [in]

Un pointer la un sir”null-terminated” Unicode ce specifica numele user-ului a carui parola este schimbata.

LmOldPresent [in]

Parametru de tip**BOOLEAN** ce specifica daca parola alocata deLmOldOwfPassword este valida. LmOldPresent este**FALSE** dacaLmOldOwfPassword este mai mare de 128-b in lungime si nu poate fi reprezentata de ncicio functie one-way Lan Manager (LM) (OWF) .Altfel acesta este**TRUE**.

LmOldOwfPassword [in]

Un pointer la structura[**LM\_OWF\_PASSWORD**](http://msdn.microsoft.com/en-us/library/ms697862%28v=vs.85%29.aspx) ce contine OWF-ul praolei utilizatorului curent.Acest parametru este ignorant daca LmOldPassword[in] este **FALSE**.

LmNewOwfPassword [in]

Un poiter la structura[**LM\_OWF\_PASSWORD**](http://msdn.microsoft.com/en-us/library/ms697862%28v=vs.85%29.aspx) ce contine OWF-ul utilizatorului si noua parola LM.

NtOldOwfPassword [in]

Un pointer la structura[**NT\_OWF\_PASSWORD**](http://msdn.microsoft.com/en-us/library/cc325731%28v=vs.85%29.aspx) ce contine OWF-ul praolei NT curenta a utilizatorului.

NtNewOwfPassword [in]

Un pointer la structura[**NT\_OWF\_PASSWORD**](http://msdn.microsoft.com/en-us/library/cc325731%28v=vs.85%29.aspx) ce contine the OWF-ul noii parole a utilizatorului.

De asemnea, exista si o versiune mai noua a functie MS-CHAP numita [MSChapSrvChangePassword2](http://msdn.microsoft.com/en-us/library/ms697869%28v=vs.85%29.aspx)ce contine sintaxa:

C++

extern WINADVAPI DWORD WINAPI MSChapSrvChangePassword2(

\_In\_  PWSTR ServerName,

\_In\_  PWSTR UserName,

\_In\_  PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldNt,

\_In\_  PENCRYPTED\_NT\_OWF\_PASSWORD OldNtOwfPasswordEncryptedWithNewNt,

\_In\_  BOOLEAN LmPresent,

\_In\_  PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldLm,

\_In\_  PENCRYPTED\_LM\_OWF\_PASSWORD OldLmOwfPasswordEncryptedWithNewLmOrNt

);

**2.3.Securitatea informatiei in Romania**

**Asociatia Nationala pentru Securitatea Sistemelor Informatice (ANSSI)** este constituita ca persoana juridica romana, fiind o organizatie de drept privat, nonprofit, fara scop patrimonial, fara scop lucrativ, neguvernamentala, profesionala, independenta, formata in scopul coagularii eforturilor si experientelor relevante in sensul promovarii standardelor si bunelor practici in domeniul securitatii informatiilor.

De asemenea ANSSI a definit urmatoarele  directii de actiune si activitati strategice:

* Implicarea in definitivarea Strategiei Nationale Cibernetice ce va solicita eforturi coordonate din partea autoritatilor cu responsabilitati specifice si a unei comunitati largi ce va include industria IT, beneficiarii ei, consultanti, organizatii profesionale.
* Pregatirea lansarii unei platforme de comunicare a subiectelor de interes major, ce va permite incurajarea schimbului de informatii pentru securitatea sistemelor informatice si facilitarea interactiunii la nivelul autoritatilor (de exemplu cu implicarea CERT), comunitatii de afaceri si publicului larg pentru a raspunde intr-un mod eficient si coordonat la amenintarile cibernetice
* Sustinerea operationalizarii unor elemente ale Strategiei Nationale Cibernetice
* Implicarea  in reglementarea unor zone de activitate considerand necesitatile specifice (ca de exemplu reglementarea traficului pe internet, responsabilitatea sociala a ISP din punctul de vedere al protectiei si securitatii utilizatorilor, diseminarea si reglementarea corespunzatoare a semnaturii electronice).
* Adoptarea unui cod de bune practici la nivelul comunitatii de afaceri  care sa permita consolidarea culturii si aptitudinilor in domeniul securitatii informatiei, precum si definirea rolului Information Security Officer in cadrul companiilor si a unei bune guvernante  corporatiste in materie.
* Organizarea de training-uri, cursuri pentru formarea profesionala in domeniul securitatii sistemelor informatice si initierea de programe pentru constientizarea riscurilor la care sunt supuse sistemele informatice

****

**Capitolul III:Apelurile de sistem legate de securitate la Linux(Chirigiu Corin)**

**3.1.Anatomia unui apel de sistem**

In domeniul IT un apel de sistem este modalitatea prin care un program cere un serviciu kernelul-ui sistemului de operare. Acesta poate consta intr-un serviciu ce se leaga de hardware (ex: accesul unui hard disk), crearea si executarea de procese noi precum si comunicarea cu servicii integrate in kernel (cum ar fi scheduling). Apelurile de sistem ofera o interfata esentiala intre proces si sistemul de operare.

In general sistemele ofera o librarie sau un API care este “amplasat” intre programele normale si sistemul de operare. Pe sisteme de tip Unix acel API este deobicei o parte a implementarii librariei C (libc), cum ar fi glibc, ce ofera functii wrapper pentru apelurile de sistem.

Apelul la functiile librariei nu produce o comutare la modul kernel automata ( daca functia nu a fost deja in modul kernel) , acesta fiind deobicei un apel de subrutina obisnuit (ex: metoda de assembler “CALL” in unele arhitecturi de instructiuni). Apelul de sistem propriu-zis transfera controlul la kernel (este mai dependent de implementare si de platforma decat apelul la librarie). In cazul sistemelor de operare Unix , functiile “fork” si “execve” sunt functii ale librariei C care executa la randul lor instructiuni ce invoca apelurile de sistem “fork” si “execve”. Efectuarea apelului de sistem in interiorul aplicatiei este dificil de implementat si necesita cod de assembler integrat (in C si C++) precum si cunostinte de interfata binara low-level pentru operatia de apel de sistem, care pot fi susceptibile la schimbari in timp astfel nefacand parte a aplicatiei de interfata binara.

Implementarea apelurilor de sistem necesita un transfer de control ce necesita un anume tip de arhitectura. O metoda tipica de a implementa acest lucru este utilizarea unui trap sau interrupt software. Intreruperile cedeaza controlul kernel-ului sistemului deci software-ul trebuie doar sa seteze anumite registre cu numarul apelului de sistem si sa execute intreruperea de tip software.

Aceasta utilizare a API-urilor ofera si anumite avantaje:

-**Portabilitate**: atata timp cat API-ul este suportat de un sistem, orice sistem utilizeaza acel API poate compila si rula.

-**Facilitate in functionare**: utilizarea API-ului este mult mai usor decat utilizarea propriu-zisa a apelului de sistem.

Apelurile de sistem pot fi impartite in mai multe categorii: control de proces, manipularea fisierelor, manipularea echipamentelor, mentenanta informatiei si comunicare. Kernel-ul pastreaza o lista a tuturor apelurilor de sistem inregistrate intr-o tabela numita *tabela de apeluri de sistem.* Aceasta tabela asociaza fiecarui apel de sistem valid un numar unic de apel de sistem care nu poate fi modificat sau reciclat. Referinta la apelurile de sistem de catre procese nu se efectueaza prin numele apelurilor ci prin aceste numere asociate.

Orice apel de sistem are un set de parametrii. Exista 3 metode de transmitere a parametrilor catre sistemul de operare:

1. Transmiterea parametrilor prin registre.
2. Cand sunt mai multi parametrii decat registre , parametrii pot fi stocati in blocuri si adresa blocului este transmisa ca parametru catre un registru.
3. Parametrii pot fi extrasi/introdusi cu ajutorul instructiunilor push/pop in stiva de catre sistemul de operare.

In continuare vom studia cateva apeluri de sistem.

**3.2.Exemple de apeluri de sistem**

Asa cum am spus in subcapitolul 3.1 apelurile de sistem pot fi impartite in mai multe categorii.In acest subcapitol ne vom ocupa de analiza acestor apeluri de sistem.

***Apelurile de sistem ce se ocupa cu controlul proceselor***

In sistemele de tip Unix exemple de apeluri de sistem ce se ocupa cu controlul proceselor sunt: fork(), exit(), wait().

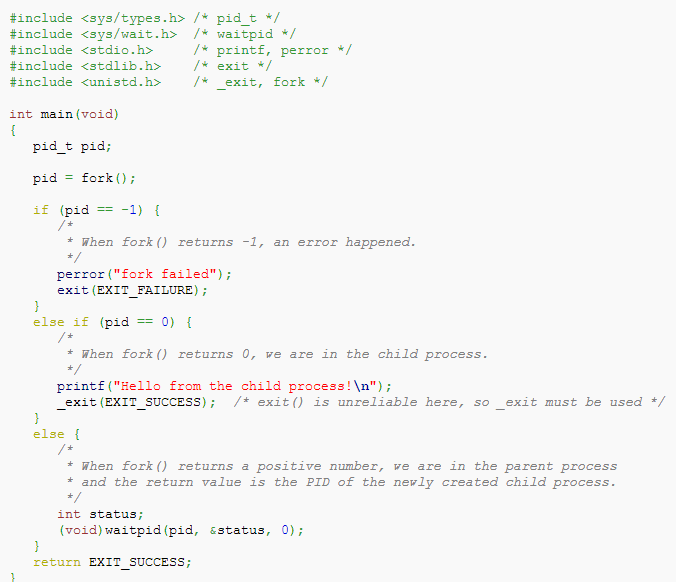
**Fork()**  este un apel de sistem prin care un proces creeaza o copie a sa.**Fork**() este principala (si istoric vorbind, unica) metoda de a crea procese in sistemele de tip Unix. Intrebarea fireasca este de ce ne trebuie **fork()**?

In sistemele de operare de tip multitasking procesele (programe ce ruleaza ) au nevoie de o metoda prin care pot crea noi procese (de exemplu pentru a rula alte programe). **Fork()** si variantele acestui apel de sistem este singura metoda prin care putem realiza acest lucru in sistemele de operare de tip Unix. Pentru ca un proces sa porneasca executia unui alt program prima oara apeleaza**fork()** pentru a crea o copie a sa. Atunci copia, numita “proces copil”, apeleaza apelul de sistem **exec** pentru a se suprapune cu celalalt program, stopandu-si executia in favoarea noului program.

In sistemele de operare de tip Unix cu suport de memorie virtuala se creeaza un spatiu de de adrese separat pentru procesul “copil” ce contine toate segmentele de memorie a procesului parinte.

Mai exista cateva variante ale functiei **fork()** cum ar fi: vfork (a carei utilitate a devenit nefolositoare in anul 2004), rfork (introdus in sistemul de operare plan 9), clone (un apel de sistem in kernel-ul Linux inspirat de rfork).

In figura de mai jos ilustram un exemplu de utilizare a functiei **fork** (sursa: <http://en.wikipedia.org/wiki/Fork_(system_call))>.



**Fig.1 Exemplu de utilizare a functiei fork() (sursa:** [**http://en.wikipedia.org/wiki/Fork\_(system\_call))**](http://en.wikipedia.org/wiki/Fork_(system_call)))**.**

Un alt exemplu de apel de sistem este functia **exit()**. In multe sisteme de operare un proces isi finalizeaza executia prin utilizarea acestui apel de sistem. Intr-un sistem de tip multithread un exit presupune oprirea unui fir de executie. Din punct de vedere al managementului resurselor, sistemul de operare elibereaza resursele folosite de procesul apeland al functiei **exit().**

Atunci cand procesul “copil” (creat in urma apelului de sistem **fork()**) este inchis, fie prin **exit**, fie din cauza unei erori sau semnal (ex: SIGTERM, SIGINT, SIGKILL), sistemului de operare ii este trimis un exit status si procesului “parinte” ii este trimis un semnal SIGCHLD. Statusul exit poate fi apoi recuperat de catre procesul “parinte” cu ajutorul unui apel de sistem **wait.** Marea majoritate a sistemelor de operare permite procesului ce se termina sa ofere un exit status specific (deobicei o variabila de tip intreg dar si char in unele sisteme de operare ex: Plan 9).

Sintaxa in Unix shell pentru apelul de exit este *exit 0*.

In paragraful anterior am introdus notiunea de **wait,** acesta este un apel de sistem utilizat de catre procesul “parinte” prin care isi suspenda executia, permitand astfel rularea procesului “copil”. Dupa terminarea acestuia procesul “parinte” isi reia executia.

O alta categorie de apeluri de sistem este aceea de manipulare a fisierelor, din aceasta categorie ne vom ocupa de: **open(), read(), write(), close().**

**Open()**  este apelul de sistem utilizat atunci cand se initializeaza accesul la fisier, acest apel aloca resursele asociate fisierului (descriptorul de fisier) si returneaza un pointer (ce devine descriptorul fisierului) care va fi utilizat de catre proces pentru a face referinta la fisier.

**Open()** are 3 argumente la apelare:

1. Calea catre fisier
2. Tipul de acces cerut (read,write,append etc.)
3. Nivelul de permisiune al fisierului initial este cerut prin intermediul celui de-al 3-lea argument numit *mode*.

In marea majoritate a sistemelor de operare moderne pentru accesarea datelor dintr-un fisier utilizam apelul **read().**Acest apel de sistem citeste datele (sub forma de bytes) din fisier si le stocheaza intr-un buffer oferit de catre procesul apelant.

**Read()** poate avea 3 argumente:

1. Descriptorul fisierului
2. Buffer-ul unde se stocheaza datele
3. Numarul de bytes ce trebuiesc cititi din fisier.

**Write()** este una dintre cele mai primare rutine oferite de kernel. Acesta permite scrierea de date (sub forma de bytes) dintr-un buffer declarat de catre utilizator in program, intr-un fisier indicat de catre procesul apelant.

**Write()** are tot 3 argumente:

1. Descriptorul de fisier
2. Bufferul de unde se preiau datele ce vor fi scrise
3. Numarul de bytes ce vor fi cititi din buffer.

**Close()** se utilizeaza pentru inchiderea accesului la fisier a unui program. Acest apel de sistem reimprospateaza metadata fisierului si tabela de fisiere in uz a sistemului de operare, elibereaza buffer-ele si dezaloca resursele alocate fisierului.

Continuam prezentarea apelulilor de sistem cu cele ce vizeaza manipularea dispozitivelor, acestea sunt: **ioctl(), read(), write().**

**Ioctl()** vizeaza operatiile de input/output ale dispozitivului, acest apel are un parametru prin care se specifica tipul de cerere (request code). Efectul apelului depinde in totalitate de acest request code, care de multe ori este specific dispozitivului (ex: deschiderea cd-rom-ului are un anumit request code). Acest apel de sistem poate fi utilizat pentru configurarea dispozitivelor hardware, pentru terminale, pentru extensia kernel-ului.

**Ioctl()** primeste ca parametrii:

1. Un descriptor al fisierului deschis
2. Un request code
3. Un integer sau un pointer catre data.

Apelurile de sistem **read()** si **write()**  pentru dispozitive au aceleasi functionalitati ca si cele pentru fisiere.

Apelurile de sistem sunt utilizate si pentru mentenanta informatiei, din aceasta categorie amintim: **getpid(), alarm(), sleep().**

**Getpid()**  este utilizat pentru returnarea id-ului procesului apelant. Acest id este unic si folositor pentru a crea nume temporare de fisier. **Getppid()** returneaza id-ul “parintelui” procesului apelant.

**Alarm()** primeste ca paramteru un intreg, acest apel de sistem genereaza un semnal SIGALRM dupa un numar de secunde reprezentat de parametru primit. Procesul apelant este cel caruia ii este destinat semnalul. Acest apel de sistem pregateste un proces pentru a primi semnale.

**Sleep()** este utilizat pentru a suspenda starea unui proces pentru un numar de secundeprimit ca parametru. Expirarea timpului sau primirea unui semnal/intrerupere provoaca stoparea starii de suspendare a procesului.

O alta categorie de apeluri de sistem este cea a apelurilor de sistem folosite pentru comunicare. Vorbim aici de **pipe(),shmget(),mmap().**

Functia **pipe()**  creeaza un *pipe*, un obiect ce permite flux de date bidirectional, de asemenea aloca o pereche de descriptori de fisier. Prin conventie primul descriptor este folosit drept capat de citire, al doilea capat de scriere. Pipe-ul exista pana cand toti descriptorii asociati sunt inchisi.Un pipe cu un capat inchis este denumita *widowed* scrierea pe un astfel de pipe returneaza un semnal SIGPIPE.

Functia **shmget()** aloca un segment de memorie comun. Ea returneaza identificatorul segmentului de memorie asociat valorii argumentului cheie.

Functia **mmap()**  creeaza o harta a fisierelor sau a dispozitivelor in memorie. Aceasta poate primi urmatorii parametrii:

* Dimensiunea in bytes a obiectului
* Protectia memoriei (execute, read, write, none)
* Tipul obiectului ce trebuie mapat
* Descriptorul de fisier de unde preluam informatia
* Offset ce indica pozitionarea bytes-ilor de informatie.

Unele apeluri de sistem tin de securitate de exemplu functiile: **chmod(), umask(), chown().**

**Chmod()** vine de la change mode, aceasta functie schimba nivelul de acces al unui fisier. Aceasta functie are urmatoarea sintaxa:

**chmod [OPTION] mode[,mode]FILE**

Parametrul mod poate lua atat valori octale cat si un grup de cuvinte cheie cu functii prestabilitie (de exemplu: ***chmod 666 file.text*** este echivalentul ***chmod a+rw file.text*** care schimba modul de acces la read-write pentru fisierul file.text). File reprezinta fisierul asupra caruia aducem modificari de acces.

**Umask()** este o functie utilizata pentru a controla ce permisiuni se seteaza pentru fisiere si directoare atunci cand sunt create. Atunci cand masca are bitul “1” permisiunea corespunzatoare este dezactivata, daca bitul are valoarea “0”, atunci permisiunea corespunzatoare va fi setata de program si sistem. **Umask()** aceleasi notatii si valori octale ca si **chmod().**

**Chown()** schimba proprietarul (owner) si grupul unui fisier atunci cand este utilizat. Se mai intalnesc in practica variantele: **fchown()** (este specificat fisierul prin intermediul descriptorului de fisier) si **lchown()** (asemanator **chown()** dar nu dereferentiaza legaturile simbolice). Aceasta functie primeste ca parametrii noul owner, noul grup si calea de acces/descriptorul de fisier.

Un exemplu de functionare a unui apel de sistem in cod Kernel (<https://www.student.cs.uwaterloo.ca/~cs350/F07/notes/kernel.pdf>):

mips\_syscall(struct trapframe \*tf) /\*functia care asociaza apelul de sistem de la nivelul user cu functionalitatea acesteia din Kernel. Parametrul dat functiei reprezinta adresa structurii de exceptii a sistemului (frame-ul), apelul de sistem fiind privit ca o exceptie.\*/

{

assert(curspl==0);

callno = tf->tf\_v0; /\* se identifica numarul apelului de sistem din structura de exceptii, asta inseamna ca apelul de sistem a fost stocat in registrul v0 inainte de executarea syscall\*/

retval = 0;

switch (callno) { /\*evalueaza sistemul de apel pentru a identifica functionalitatea acestuila\*/

case SYS\_reboot:/\* aici pot fi mai multe tipuri de apeluri de sistem, noi analizam SYS\_reboot\*/

is in kern/main/main.c

err = sys\_reboot(tf->tf\_a0);/\* eroarea (apelul de sistem la nivelul user) ii este atribuita functia din kernel corespunzatoare e.g. sys\_reboot()\*/

break;

default://in cazul in care nu se identifica apelul de sistem in kernel

kprintf("Unknown syscall %d\n", callno);//afiseaza un mesaj ca nu exista acest apel in kernel

err = ENOSYS;

break;

} if (err) {//ne pregatim pentru returnarea apelului de sistem, daca exista in kernel

tf->tf\_v0 = err;/\*incarcam in registrul v0 eroarea, aici nivelul user va identifica apelul de sistem corespunzator\*/

tf->tf\_a3 = 1; //semnaleaza o eroare , registru implicit

} else {

tf->tf\_v0 = retval;//daca nu exista apelul de sistem in v0 nu avem nimic (retval=0)

tf->tf\_a3 = 0;//nu semnaleaza eroare

}tf->tf\_epc += 4; //avanseaza PC pentru a evita o repetare a apelului

assert(curspl==0);

}

**Capitolul IV:Implementarea securitatii la Windows/Linux(Chirigiu Corin)**

**4.1 Caracteristici generale Windows/Linux**

Marea majoritate a sistemelor de operare moderne ofera posibilitatea de a rula simultan mai multe aplicatii intr-un singur sistem hardware. Cu un asemenea mediu de multitasking si time-sharing, aplicatiile utilizeaza aceleasi resurse ale sistemului (ex: CPU, memorie, dispozitive I/O) aflate sub controlul sistemului de operare. Pentru a impiedica atacul unui proces de catre altul concomitent, cele mai multe sisteme de operare implementeaza un tip de container abstract, cum ar fi un proces, TCB, memorie virtuala, fisier, port, IPC etc. O aplicatie este protejata astfel incat sa aibe acces la anumite resurse si sa execute anumite operatii.

Totusi aceasta distribuire in sistemele de operare comerciale presupune o siguranta bazata pe owner si identitatea user-ului fara a lua in considerare criterii ,relevante din punct de vedere al securitatii, cum ar fi nivelul de incredere in aplicatie si ce executa aceasta, rolul user-ului sau integritatea datelor. Atata timp cat userii sau aplicatia are acces intreg asupra obiectelor este dificil sa se controleze fluxul de date sau sa se implementeze un sistem global de securitate. Din aceste motive este destul de dificil de protejat un intreg sistem de operare, fiind relativ usor sa spargi securitate unui sistem odata ce o aplicatie a fost compromisa (ex: un buffer overflow attack).

Cateva dintre posibilele vulnerabilitati ale unei aplicatii:

-Utilizarea ilegitima a resurselor neprotejate ale sistemului. De exemplu un “vierme” trimite mail-uri tuturor contactelor din lista de email odata ce contul de mail a fost compromis.

-Obtinerea accesului asupra resurselor protejate ale unui sistem prin folosirea ilegitima a privilegiilor. De exemplu un “sendmail” pe un sistem Unix rezulta in obtinerea de privilegii super a atacatorului si acces necontrolat asupra resurselor sistemului.

-O multitudine de fisiere de securitate (cu rolul de a lua decizii) false.

Este imposibil de protejat prin metode comune un sistem de operare modern impotriva codului malitios intrucat o aplicatie ce ruleaza sub un anumit user are toate privilegiile asociate acelui user. Pe langa acest aspect trebuie sa ne gandim ca repartizarea nivelurilor de privilegii in sistemele de operare se efectueaza in 2 categorii (extreme si ineficiente), vorbim de super-useri (administratori) care au acces la orice si useri normali cu acces limitat. Ca rezultat aplicatiile privilegiate si serviciile de sistem trebuie sa functioneze cu nivel de privilegiu ce permite acces la root, ceea ce de multe ori nu este necesar.

“**The Orange Book**” imparte in categorii in functie de diferite niveluri de securitate si defineste nevoile fundamentale de securitate ale unui computer:

**D-Protectie minimala**: nici un nivel de securitate nu este necesar; sistemul nu se califica pentru nivelurile superioare.

**C1-Protectie discreta:** sistemul trebuie sa identifice diferiti useri ce ruleaza in sistem, si ofera mecanisme pentru autentificare si autorizare a user-ilor pentru a preveni interferenta reciproca a programelor ce ruleaza.

**C2-Protectie prin controlul accesului:** acest tip de sistem are in plus anumite elemente de securitate, de exemplu poate dezaloca resurse hard sau de memorie.

**B1-Protectie prin label:** sistemul implementeaza protectia prin MAC prin care fiecare subiect si obiect sunt caracterizate de un label, si fiecare subiect trebuie sa verifice label-urile si sa respecte anumite reguli clare de securitate.

**B2-Protectie structurata:** nu sunt introduse caracteristici noi de securitate, se axeaza mai mult pe structura pentru a asigura integritatea acestuia.

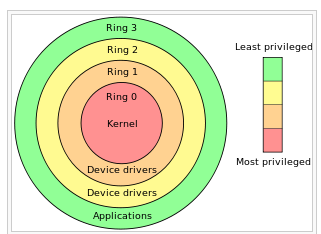
**B3-Domenii de securitate:** se refera la sisteme suficient de mici astfel incat nu pot avea erori care ar ocoli controlul obligatoriu de acces.

**A1-Design verificat:** fata de **B3** sunt anumite proceduri formale in plus pentru analiza design-ului de sistem si control mai riguros al implementarii.

Marea majoritate a sistemelor de operare comercial au un rating **C2** sau mai jos, principala metoda de protectie o reprezinta nivilele de privilegii.

Atunci cand discutam despre securitate, trebuie analizata si structura ierarhica a domeniilor de protectie, cunoscuta si ca structura bazata pe inele. Aceasta structura consta in limitarea accesului unui proces la anumite date, acest lucru, asa cum am amintit si mai sus, protejeaza sistemul ex: O aplicatie compromisa nu trebuie sa aiba acces la date din nivelul kernel-ului pentru a limita distrugerile provocate.

Arhitectura Intel x86 prezinta 4 astfel de inele, numerotate 0-3 (0-cel mai privilegiat, 3-cel mai putin privilegiat).



**Fig.2 Structura inelelor de protectie (sursa:** [**http://en.wikipedia.org/wiki/Protection\_ring**](http://en.wikipedia.org/wiki/Protection_ring)**)**

Impartirea aceasta are ca scop protejarea a 3 elemente cheie dintr-un sistem: memoria, porturile I/O si abilitatea de a executa anumite instructiuni masina. La nivelul cel mai privilegiat (0) regasim cum este de asteptat kernel-ul, reprezentand structura de baza a oricarui sistem. In nivelul 1 se regasesc portiuni neprivilegiate ale sistemului de operare. Nivelul 2 este rezervat utilitarelor pentru porturile I/O, operatii low-level si diferite utilitati in timp ce nivelul 3 este locul in care sunt executate aplicatiile si procesele- acesta este nivelul user. Marea majoritate a sistemelor moderne de operare utilizeaza doar nivelul 3 si nivelul 0.

Pentru a intelege cum functioneaza aceasta schema introducem notiunile de CPL (current privilege level), RPL (requested privilege level), DPL (descriptor privilege level). CPL este un camp in segmentul selector de cod care indica nivelul sau de privilegiu, RPL este un camp in segmentul selector de date ce permite kernel-ului sa acceseze date folosind un nivel de privilegiu redus, iar DPL este un camp din segmentul descriptor ce indica nivelul de privilegiu al segmentului de data/cod pe care dorim sa il accesam/apelam. Criteriul de baza pentru a decide daca se permite accesul la segment este urmatorul: max(RPL,CPL) <= DPL, daca maximul dintre acestea este mai mare decat DPL (mai putin privilegiat) atunci nu se permite accesul la segement. Pentru a se putea face schimburi intre nivelurile de privilegiu diferite utilizam porti. Acestor porti le sunt asociate descriptori si se impart in 4 categorii: descriptori de poarta-apel (call-gate), descriptor de poarta-intrerupere (interrupt-gate), descriptor de poarta-capcana (trap-gate) si descriptor de poarta-sarcina (task-gate). Portile apel ofera un punct de acces in interiorul kernel-ului si sunt utilizate cand folosim instructiuni de tip call sau jump. Portile de sarcina nu sunt atat de utilizate (in Linux sunt folosite cand apar probleme de hardware sau kernel). Portile capcana si cele de intreruperi sunt cele mai frecvent utilizate pentru tratarea intreruperilor hard si a exceptiilor (am vazut importanta exceptiilor si asocierea lor cu apelurile de sistem). Descriptorii acestor porti sunt stocati intr-o tabela numita tabela de intreruperi (IDT). Importanta acestor tipuri de porti este data de faptul ca toate functionalitatile intreruperilor/exceptiilor la Linux se executa in interiorul kernel-ului (ex: Codul si comentariul din capitolul anterior).

De retinut este faptul ca aceasta metoda de securizare nu tine cont de utilizatorii sistemului de operare. Administrator, user, guest, super user toti utilizeaza cod cu un nivel 3 de privilegiu, in timp ce kernel-ul are nivelul 0 de privilegiu comunicarea intre cele 2 fiind posibila datorita existentei portilor. Este foarta importanta aceasta distinctie intre securitatea microprocesorului (relatiile inter-segmente) si cea a sistemului de operare propriu-zisa.

**4.2 Accesul bazat pe niveluri de privilegii Windows/Linux**

Un privilegiu inseamna dreptul unui cont, cum ar fi un cont de grup sau user pentru a executa diferite operatii pe sistem cum ar fi stingerea calculatorului, modificare timpului sistemului, incarcarea driver-elor pentru dispozitive.

Privilegiul nu trebuie confundat cu dreptul de acces, privilegiile vizeaza operatiile asupra sistemului si este oferit de un administrator de sistem unui user sau unui grup.

Vom analiza in continuare metode de a separa nivelurile de privilegii la cele 2 sisteme de operare Windows si Linux.

Cea mai buna metoda de a efectua o impartire a nivelurilor de privilegii este pe grupuri si nu pe useri individuali, grupurile sunt mai usor de controlat. Windows (sa luam exemplu Windows 7) are cateva grupuri “built-in” ce pot fi utilizate.

**Administratorii**  au cel mai mare nivel de privilegii intr-un sistem Windows. Membrii acestui grup pot efectua marea majoritate a proceselor pe un sistem.

**Backup operators**membrii acestui grup pot sa nu aiba acces la fisierele unui sistem, dar pot citi acele fisiere cu scopul de a efectua un back-up.

**Event log reader** membrii acestui grup pot citi rezultatele evenimentelor petrecute in acest sistem, desi acestia nu sunt administratori. Ei pot citi de asemenea rezumatele evenimentelor ce tin de securitate.

**Network configuration operators**membrii acestui grup pot aduce modificari la configuratia retelei.

**Performance log users and performance monitor users** membrii acestui grup pot monitoriza performanta sistemului si pot urmari loggin-ul.

**Remote desktop users** sunt userii ce pot opera desktop-ul pe sistem.

Aceasta enumeratie este doar o mica parte a posibilitatii de repartizare a nivelurilor de privilegii, putem vorbi de diferite add-on-uri ce includ niveluri de privilegii asupra bazelor de date si multe alte operatii.

Active directory este un serviciu de directoare implementat de Microsoft pentru retelele Windows (Windows domain). Este inclus in marea majoritate a sistemelor de operare de tip Windows Server. Intreaga securitate a unei retele ce implementeaza acest serviciu este controlata de un controler de domeniu (domain controller), acesta autentifica si autorizeaza toti utilizatorii si toate computer-ele de pe o retea de tip Windows. De exemplu cand un utilizator se conecteaza la o retea de tip Windows, controller-ul verifica parola si numele pentru a determina daca acest utilizator este administrator sau utilizator normal. Active directory foloseste LDAP (lightweight directory access protocol), o versiune Microsoft pentru Kerberos si DNS.

LDAP este un protocol standard pentru aplicatii, pentru accesarea si intretinerea serviciilor de informatie de director utilizate intr-o retea ce implementeaza IP (internet protocol). Serviciile de directoare joaca un rol important in dezvoltarea aplicatiilor pentru intranet sau internet, deoarece ofera posibilitatea de a impartasi informatii despre utilizatori, sistem, retea, servicii in interiorul retelei. Una dintre cele mai intalnite utilizari ale LDAP este “single sign on” ce permite utilizarea unei singure parole de catre un utilizator pentru a accesa mai multe servicii (e.g. logare la internet, la mail, Windows cu aceeasi parola).

DNS (domain name system) este un sistem de numire ierarhic distribuit pentru calculatoare, servicii, sau orice alta resursa conectata la Internet sau o retea privata. Acest sistem permite asocierea unor nume de domenii cu adresa IP pentru a efectua o localizare a serviciilor si a dispozitivelor in toata lumea.

Active directory este alcatuit dintr-o baza de date si codul executabil corespondent responsabil pentru solutionarea cererilor si mentenanta bazei de date. Partea executabila, cunoscuta ca Directory System Agent, este o colectie de servicii si procese Windows ce ruleaza pe platforma Windows 2000 si versiunile mai noi de controler de domeniu. Obiectele din Active Directory pot fi accesate prin intermediul protocolului LDAP, ADSI, MAPI si managerul de securitate al conturilor.

Structura Active Directory ce contine obiectele poate fi vazuta ca o repartizare pe diferite niveluri. Aceste diviziuni logice sunt numite forest, tree si domain.

Intr-o implementare obiectele sunt grupate in domenii. Obiectele pentru un domeniu sunt stocate intr-o singura baza de date, care poate fi copiata. Domeniile sunt identificate prin structura lor de nume, DNS.Un domeniu este un grup logic de obiecte de retea (calculatoare, utilizatori, servicii). O grupare de domenii cu DNS invecinate sunt grupate intr-un tree (copac). O colectie de tree care impart un catalog global comun, o schema de director si o configuratie de director formeaza un forest (padure).

Obiectele din interiorul unui domeniu pot fi grupate in unitati organizationale (OU), acestea pot oferi ierarhie intr-un domeniu si poate usura administratia acestuia.

Pentru a permite accesul unui utilizator dintr-un domeniu la resurse aflate in alt domeniu Active Director foloseste un sistem de “incredere” (trust).

Aceste trust-uri sunt create automat cand sunt create domeniile, forest (structura din varful piramidei ierarhice din AD) seteaza limite implicite de incredere si implicit trust-ul tranzitiv se seteaza pentru toate domeniile din interiorul unui forest.

Din acest punct de vedere trust-urile se impart in:

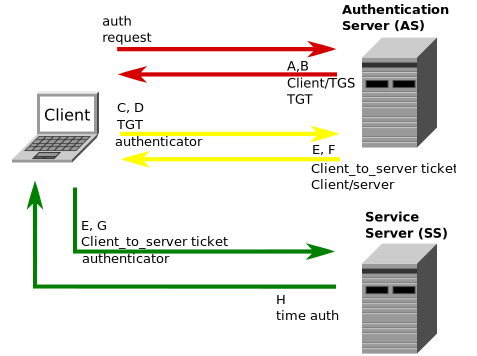
* One-way trust: Un domeniu permite accesul utilizatorului in alt domeniu, viceversa nefiind valabila.
* Two-way trust: Doua domenii permit accesul utilizatorilor intre ele.
* Trusting domain: Un domeniu permite accesul utilizatorului dintr-un domeniu in care are “incredere”.
* Trusted domain: Domeniul de “incredere” ai carui utilizatori au acces in trusting domain.
* Transitive trust: O “incredere” ce poate fi extinsa la mai mult de 2 domenii din interiorul forest-ului.
* Intransitive trust: O ”incredere” intr-un singur sens (one-way) ce nu poate fi extinsa la mai mult de 2 domenii.
* Explicit trust: Creata de admin, nu este tranzitiva si este intr-un singur sens.
* Cross-link trust: O “incredere” explicita intre domeniile din structuri de tree diferite, sau in acelasi tree daca nu exista relatie de mostenire intre cele 2 domenii.
* Shortcut: Uneste 2 domenii in structuri tree diferite. Tranzitiva singur sau dublu sens.
* Realm: Poate fi atat tranzitiva cat si intranzitiv, sens unic sau dublu.
* External: Conexiune la alte structuri forest sau domenii non-AD. Intranzitive, sens unic sau dublu.

Prima structura ierarhica construita in AD poarta denumirea de root forest. Administratorii enterprise si de scheme pot exista doar in domeniul root forest. Aceste grupuri de administratori sunt baza administratiei si securitatii unui sistem, avand acces nelimitat. Odata ce structura ierarhica a fost construita trebuie limitat accesul celorlalti utilizatori.

Protocolul Kerberos este un protocol de autentificare pentru o retea de calculatoare ce functioneaza pe baza de “ticket” (tichet) pentru a permite nodurilor, ce comunica intr-o retea nesigura, sa se identifice unul fata de celalalt intr-un mod sigur. Kerberos utilizeaza criptografia cu cheie simetrica ( o clasa de algoritmi ce folosesc aceeasi cheie pentru criptare si decriptare, cheia este cunoscuta de 2 sau mai multe parti si poate fi utilizata pentru a pastra informatia privata) si are nevoie de o a treia parte de incredere pentru efectuarea transferului de informatie.

Protocolul Kerberos a fost dezvoltat de MIT la sfarsitul anilor ’80 si se bazeaza pe protocolul cu cheie simetrica Needham-Schroeder. Windows 2000 (si versiunile noi) utilizeaza Kerberos ca protocolul implicit in autentificare. Windows utilizeaza acest protocol dar nu si software-ul MIT, cand server-ul si clientul nu fac parte din acelasi cerc de incredere Kerberos nu este utilizat, Windows utilizand NTLM (NT Lan Manager protocol de securitate ce ofera integritate, autentificare si confidentialitate utilizatorilor) pentru autentificare.

Modul de functionare al protocolului se bazeaza pe cunoasterea reciproca intre client si server. Clientul se autentifica server-ului de autentificare (Authentication Server-AS) ce transmite numele utilizatorului unui centru de distributie a cheii (Key Distribution Center-KDC). KDC elibereaza un Ticket Granting Ticket (TGT), caruia i se atribuie data si ora, acesta este criptat utilizand parola utilizatorului si transmite rezultatul sistemului clientului. Atunci cand un client doreste comunicarea cu un alt nod trimite TGT serviciului de oferire al tichetelor (Ticket Granting Service- TGS), care imparte de obicei aceeasi gazdaca si KDC. Dupa validarea TGT-ului utilizatorului i se permite accesul la serviciu cerut, TGS elibereaza un tichet (ticket) si chei de sesiune, clientului. Clientul transmite atunci tichetul si cererea de serviciu server-ului de servicii.



**Fig.3 Ilustrarea protocolului Kerberos (sursa: http://upload.wikimedia.org/wikipedia/commons/4/4e/Kerberos.svg)**

Linux utilizeaza software pentru autentificarea utilizatorilor si a serviciilor prin protocolul Kerberos.

Linux a fost creat la inceputul anilor ’90 ca fiind o clona a sistemului de operare Unix, mostenind astfel nucleul Unix de securitate- Discretionary Access Control (DAC). DAC permite “proprietarului” unui obiect sa seteze polita de securitate pentru acel obiect. Ca utilizator poti crea un fisier nou si poti decide cine are dreptul sa citeasca/scrie in acest fisier. Permisiunile pentru acces la fisier pot fi diferite pentru un utilizator, un grup, sau restul (e.g. utilizatorul x poate citi in fisier, grupul y poate scrie in fisier). Ca si la Windows exista un super user, un utilizator cu toate drepturile in sistem, libertate oferita pentru a gestiona sistemul.

DAC este o schema de securitate relativ simpla si nu ofera protectii impotriva software-ului malitios, in consecinta aplicatiile privilegiate (e.g. rulate in modul super user) sunt predispuse la acest tip de atac, putand oferi atacatorului acces total la sistem. Optiunea de a dezvolta un nou sistem de securitate nu este fezabila, ca urmare s-au adaugat diferite caracteristici noi la sistemul de securitate. Astfel practic securitatea Linux consta intr-o colectie de modele de securitate.

Una dintre extensiile de securitate Linux este DAC extins, una dintre primele extensii aplicate nucleului Unix.

Listele de control a accesului POSIX sunt bazate pe un standard POSIX (Portable Operating System Interface), acesta extinde schema listei de control a Unix DAC intr-o schema mult mai detaliata, permitand permisiuni separate pentru utilizatori individuali si diferite grupuri. Sunt gestionate cu ajutorul comenzilor setfacl si getfacl.

O alta extensie a securitatii adaugata o reprezinta Namespace (spatiu de numire), acesta permite o impartire a resurselor vazute de procese, permitandu-le sa aibe o proprie viziune asupra fisierelor de sistem sau a tabelei de procese. Un exemplu de implementare a acestei functionalitati o reprezinta securitatea multi-nivel in care fisierele sunt securizate si eventual ascunse in totalitate utilizatorilor fara drept de acces.

Securitatea retelei este un alt exemplu de extensie a securitatii pentru Linux, acesta putand avea un numar mare de protocoale si caracteristici. Netfilter este o structura de retea bazata pe protocolul IP care are acces la pachetele care trec prin, in sau de la sistem. Modulele de la nivelul kernel pot lua decizii de securitate ce vizeaza aceste pachete, iptables este un astfel de modul ce implementeaza un firewall IPv4.

Linux Security Modules (LSM) este o structura ce ofera kernel-ului Linux posibilitatea de a implementa diferite modele de securitate pentru computer. In kernel-ul oficial sunt acceptate urmatoarele module de securitate AppArmor, SELinux, Smack si TOMOYO Linux.

SELinux este un modul de securitate Linux, o implementare MAC (Mandatory Access Control) conceput pentru a satisface o gama larga de cerinte de securitate, de la uz general pana la uz guvernamental sau militar. Securitatea MAC difera de cea DAC prin faptul ca este administrata central, utilizatorii nu seteaza polite de securitate resurselor proprii.

In SELinux toate obiectele din sistem, cum ar fi fisierele si procesele, sunt asociate etichete de securitate. Astfel toate interactiunile intre entitati ce presupun validarea securitatii sunt administrate de catre modulul SELinux, care decide daca operatia ar trebui sa continue sau nu.

O varianta mai simpla a SELinux o reprezinta modulul Smack, acesta implementand MAC si scheme bazate pe etichete de securitate. Cel mai des este intalnit in sistemele de tip embededd.

AppArmor presupune o schema MAC de “izolare” a aplicatiei si a fost conceput pentru a fi usor de gestionat. Este diferit de SELinux si Smack intrucat acest modul presupune etichetarea nu a obiectelor propriu-zise ci a impune restrictii de securitate pe calea de acces a fisierului (pathname). AppArmor este special si datorita caracteristicii de “invatare” prin care analizeaza comportamentul de securitate a unei aplicatii si il transfoma in profil de securitate.

TOMOYO este un modul asemanator AppArmor avand aceleasi concepte de a eticheta calea de acces si are o caracteristica de “invatare” asemanatoare AppArmor. Difera fata de AppArmor prin faptul ca sunt inregistrati arbori de invocare a proceselor (fata de aplicatii la AppArmor), descrise ca domenii. De exemplu atunci cand are loc boot la sistemul, o serie de task-uri sunt invocate, un user se logheaza scrie in consola. Aceasta inlantuire este memorata si daca apare o abatere de la inlantuire TOMOYO o neaga.

**4.3 Concluzii**

Active Directory este o platforma flexibila si scalabila de administrare a resurselor si aplicatiilor intr-o retea distribuita. Planificarea cu atentie este necesara, setarea trust-urilor si a domeniilor este esentiala, intrucat o greseala poate fi catastrofica pentru integritatea sistemului.

Am prezentat evolutia securitatii la Linux, determinata atat de schimbarile din exterior (cresterea exponentiala continua a Internet-ului ) cat si de marirea numarului de utilizatori Linux (cu cerinte din ce in ce mai diversificate). Faptul ca cerintele determina o continua schimbare in modulele de securitate Linux, ii permite acestuia sa introduca noi metode de protectie.

Nu se poate confirma existenta unui sistem de securitate infailibil implementat de un sistem de operare, dar de obicei Linux prin caracteristicile sale (module diversificate de securitate, cod open-source, utilizatori mai putini deci malware mai putin) este considerat superior din punct de vedere al securitatii.

**Bibliografie**

<http://ihacks.info/linux-windows-diferente-fundamentale/>

<http://en.wikipedia.org/wiki/User_identifier>

<http://en.wikipedia.org/wiki/Group_identifier>

<http://en.wikipedia.org/wiki/Setuid>

<https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/>

<http://kb.iu.edu/data/aumh.html>

<http://msdn.microsoft.com/en-us/library/aa375728%28v=vs.85%29.aspx>

<http://msdn.microsoft.com/en-us/library/aa380252%28v=vs.85%29.aspx>

<http://msdn.microsoft.com/en-us/library/ms697868%28v=vs.85%29.aspx>

<http://www.anssi.ro/>

<http://standardizare.files.wordpress.com/2010/02/clip_image002.jpg>

[www.linuxuser.co.uk](http://www.linuxuser.co.uk)

[www.wikipedia.org](http://www.wikipedia.org)[www.google.com](http://www.google.com)

[www.linux.org](http://www.linux.org)

[www.windows.microsoft.com](http://www.windows.microsoft.com)

The challenge of Managing priviledged access on Windows Desktop and Servers,Darren Mar-Elia – Februarie 2011

[www.stst.elia.pub.ro](http://www.stst.elia.pub.ro)

Curs sisteme de operare,Stefan Stancescu

Principles of operating systems, CS 446/646

System calls and libraries, Vivek Pai, Lecture 4, COS 318, 25.09.2014

Operating system security and secure operating systems, Cui-Qing Yang, Version 1.4, Option 1 for GSEC, January 2003

<http://www.bottomupcs.com/system_calls.html>

<http://www.sans.org/reading-room/whitepapers/win2k/basic-security-issues-active-directory-191>

<http://www.pearsonitcertification.com/articles/article.aspx?p=1998558&seqNum=3>