O abordare Data Mining pentru detectarea accesului neautorizat la baza de date.

1. Introducere

Deşi multe abordări diferite sunt folosite pentru a proteja datele importante in mediul de astăzi, aceste metode de multe ori nu reuşesc. O modalitate de a face datele mai puţin vulnerabile este de a implementa Intrusion Detection System (IDS) în sistemele informatice critice. În cazul în care un sistem informatic este compromis, o detectare timpurie este cheia pentru recuperarea datelor pierdute sau deteriorate, fără complexitate mare. În ultimii ani, cercetătorii au propus o varietate de abordări pentru creşterea eficienţei de detectare a intruziunilor şi precizie. Dar cele mai multe dintre aceste eforturi s-au concentrat pe detectarea intruziunilor la nivel de reţea sau la nivel de sistem de operare. Ele nu sunt capabile să detecteze exact ce date au fost mai exact corupte,care sunt pagubele şi care tranzacţii au realizat aceea corupţie de date. Fără aceste informaţii, evaluarea rapidă a daunelor şi recuperarea in timp optim nu pot fi atinse.

În această lucrare, ne propunem un model pentru identificarea tranzacţiilor rău intenţionate, care sunt orientate la coruperea datelor. Atunci când un atacator sau un utilizator rău intenţionat actualizează baza de date, prejudiciul rezultat se poate răspândi foarte rapid în alte părţi ale bazei de date prin intermediul utilizatorilor valabil. Detectarea rapidă şi precisă a unui atac cibernetic pe un sistem de baze de date este o condiţie prealabilă pentru evaluarea şi recuperarea rapidă a gradului de distrugere. Tranzacţiile rău intenţionate identificate în

această lucrare pot fi folosite mai târziu pentru evaluarea daunelor în baza de date şi procedurile de recuperare.

Abordarea noastră se concentrează pe dependenţele data mining printre elementele de date din baza de date. Prin dependenţă de date ne referim la corelaţii de acces la date între două sau mai multe elementele de date. Tehnicile prezentate utilizează abordarea de data mining pentru a genera dependenţele de date între elementele de date. Aceste dependenţe generate sunt sub formă de regulilor de clasificare, şi anume, înainte ca un element de date să fie actualizat în baza de date, ce alte elemente trebuie să fie citite şi după ce acest element de date este actualizat ce alte elemente sunt cele mai susceptibile de a fi actualizate de către aceeaşi tranzacţie.

1. Lucrări asemănătoare

O intruziune este definită ca orice set de acţiuni care încearcă să compromită integritatea, securitatea sau disponibilitatea unei resurse.

Inteligenţa artificială şi aplicaţii de date în exploatare de detectare a intruziunilor sunt angajaţi de către unii cercetători pentru a reduce efortul uman pentru construirea IDS şi pentru a creşte acurateţea de detectare.

De exemplu abordări data mining au fost realizate pentru a detecta comportamente frauduloase intr-o reţea de telecomunicaţii de către Fawcett şi Provost.

S-a realizat o cercetare destul de limitată în domeniul de detectare a intruziunilor in baze de date. De exemplu modelul Hidden Markov a fost propus să detecteze coruperea datelor in mod dăunător. Lee a utilizat semnăturile in timp pentru a descoperi accese nedorite la baza de date. Abordarea este de a eticheta semnătura timp la elementele de date.O alarmă de securitate este ridicată în momentul în care o tranzacţie incearcă să scrie date temporale, date care au fost deja actualizate într-o perioadă de timp.

2. Modelul de clasificare

Ne propunem să folosim abordarea data mining pentru determinarea dependenţelor de date în sistemul de baze de date. Regulile de clasificare care reflectă dependenţele de date sunt deduse direct din log-ul de baza de date. Aceste norme reprezintă de fapt identificarea datelor care probabil ar trebui citite înainte de efectuarea unei operaţii de actualizare şi care date ar trebui scrise după această operaţie de actualizare.

Tranzacţii care nu sunt conforme cu dependenţele de date generate sunt marcate ca operaţiuni anormale.

Se poate observa de la aplicaţii de baze de date din lumea reală că, deşi programul de tranzacţionare se schimbă , adesea, structura întregii baze de date şi corelaţiile esenţiale de date se schimbă foarte rar.

Modelul propus este conceput pentru a identifica tranzacţiile dăunătoare executate în SGBD de un intrus care a ocolit mecanismul de control al accesului al unui sistem de baze de date.

De exemplu, intrusul poate avea acces la o bază de date prin utilizarea atacului de injectare SQL la o porţiune a aplicatiei mai vulnerabilă sau de a fura parola unui utilizator legitim. Astfel, un intrus poate accesa baza de date de pe un site la distanţă prin executarea tranzacţiilor manual sau printr-o altă aplicaţie.

O tranzacţie este o unitate logică de prelucrare a bazei de date care include una sau mai multe operaţiuni de access la baza de date. Modelul nostru prevede ca log-urile bazei de date să înregistreze atât operaţiunile de citire cât şi de scriere pentru fiecare tranzacţie.

3. Dependenţele între date

Dependenţa de date efectuează analiza de dependenţele de date între elementele de date din baza de date. Următoarele definiţii ajută la înţelegerea conceptului.

4.1 Terminologia dependenţei de date

Deoarece scopul nostru general este de a descoperi dependenţele de date care sunt legate de succesiune de operaţii efectuate de tranzacţii, vom defini prima secvenţă în contextul nostru.

Definitia 1: O secvenţă este o lista ordonată de citire şi / sau scriere de operaţiuni. Vom indica o secvenţă **s** prin <o1(d1), o2(d2), …, on(dn)>, unde oi ∈ {r, w} şi dk este un element de date, $1 \leq k \leq n$. D(s) reprezintă setul de elemente de date conţinute în secvenţă, D(s) = {d1, d2, …, dn}.

Sprijinul pentru o secvenţă este definit ca fracţiunea din totalul tranzacţiilor care conţine această secvenţă.

Citirea şi scrierea secvenţei sunt folosite pentru a defini citirea şi scrierea dependenţei respective.

Definiţia 2: Secvenţa Citeşte al elementului x este o secvenţă cu formatul <r(d1), r(d2), …, r(dn), w(x)> care reprezintă faptul că tranzacţia ar putea avea nevoie pentru ca să citească toate datele d1, d2, …, dn în această ordine înainte de tranzacţia să actualizeze elementele de date x. Trebuie să se noteze că fiecare element de date poate avea mai multe secvenţe de citire de lungimi variabile. Toate aceste secvenţe sunt numite împreună Citeşte Secvenţa Set al acelui element de date.

Notaţia rs (x) este folosită pentru a indica citirea secvenţei de element de date x. De exemplu, luăm în considerare declaraţia următoarea de actualizare într-o tranzacţie.

Update set Tabelul 1 x = a + b + c în cazul în care d = 90;

În această declaraţie, înainte de a actualiza x, valorile a, b, c şi d trebuie să fie citite şi apoi noua valoare a lui x se calculează. Deci, <r(a), r(b), r(c), r(d), w(x)> ∈ rs(x). Trebuie să se constate că jurnalul de baza de date conţine imagini ale lui x numai înainte şi după tranzacţie,în loc de operaţia matematică utilizată pentru a calcula x, adică, x = a + b + c. Exemplul de mai sus este doar pentru a ilustra conceptul de secvenţă citită. Jurnal de date care conţine tranzacţia de mai sus poate arăta de fapt ca:

T1: r(m), r(n), w(y), r(u), r(v), r(a), r(b), r(c), r(d) w(x), r(a), w(c), commit.

Înainte de a scrie operaţia w(x), 8 elemente de date au fost citite. Unele dintre ele nu pot avea dependenţele de date cu x, de exemplu, ele sunt citite de către o altă declaraţie SQL în aceeaşi tranzacţie.

Aceasta înseamnă că noua valoare a lui x nu este direct dependentă de valorile acestor 8 elemente de date. Scopul este de a arăta că pentru actualizarea lui x, elementele a,b,c şi d trebuie mai mult ca sigur citiţi şi sunt relevanţi pentru a calcula noua valoare a lui x.Trebuie să se noteze că rezultatul poate doar ilustra că a şi b au dependenţe de date cu x. Acest lucru se poate întâmpla atunci când unele tranzacţii doar citesc valorile lui a şi b înainte de actualizarea lui x.

Definiţia 3: Secvenţa de scriere de elemente de date x este o secvenţă cu formatul < w(x), w(d1), w(d2), …, w(dn) > care reprezintă faptul că tranzacţia ar putea avea nevoie pentru a scrie toate datele d1, d2, …, dn în această ordine după ce tranzacţia actualizează elementele de date x. Trebuie să se constate că fiecare element de date poate avea mai multe

secvenţe de scriere de lungimi diferite. Toate aceste secvenţe sunt numite împreună Scrie Secvenţa Set al acelui element de date.

De exemplu considerăm următoarea declaratie de actualizare într-o singură tranzacţie:

Update Table1 set x = a + b + c where …

Update Table1 set y = x + u where …

Update Table1 set z = x + w + v where …

Folosind exemplul de mai sus, se poate nota că <w(x), w(y),w(z)> este o secvenţă de scriere a elementului de date x , <w(x), w(y),w(z)> ∈ ws(x), unde ws(x) arată secvenţa set de scriere a lui x.

Definiţia 4: Greutatea dependenţei datelor indică în ce măsură un element de date x depinde de alte elemente de date, de exemplu, D (s) - x, în secvenţa de scriere sau citire s. Este definită de posibilitatea de citire (scriere),a acestor elemente de date înainte sau după actualizarea lui x. Noţiunile rweight(x, D(s) - x) şi wweight(x, D(s) - x) denotă greutatea de dependenţei de citire respectiv scriere. Un prag pre-set este utilizat pentru a identifica dacă o dependenţă este slabă sau puternică.

Figura 1 ilustrează un exemplu de dependenţă de date. Element de date x relaţii de dependenţă de citire cu {a, b, c, d}, {c, d}, şi {x, e, f}. De asemenea are relaţii de dependenţă de scriere cu {y, z} şi {u, v}. Să presupunem că pragul predefinit de greutate de dependenţă de date este de 40%. Apoi, pentru dependenţa de citire doar {a, b c, d} are o dependenţă de date puternică cu x. În mod similar pentru dependenţa de scriere numai {u, v}are o dependentă de date puternică cu x.

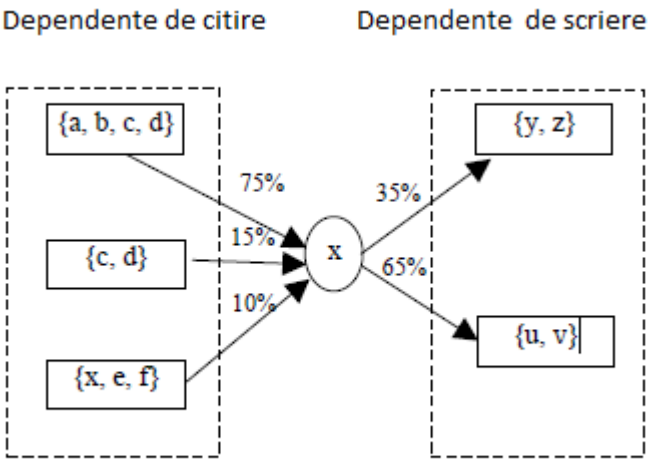

Figura 1.Exemplu de dependente de date

3.2 Metodologia

Pentru că metoda noastră utilizează secvenţe de operaţii, de exemplu, ce succesiune de operaţii de citire trebuie să fie efectuate înainte de o operaţie de actualizare şi ce succesiune de operatii de scriere trebuie să se facă după acceaşi operaţie de actualizare, este în mod intuitiv similar cu modelul secvential.

Dar , doar prin utilizarea algoritmului modelului secvenţial pe jurnalul sau logul bazei de date,putem obţine câteva modele secvenţiale consistente de operaţii mixte de citire şi scriere şi aceste modele secvenţiale nu reflectă neapărat corelările de date esenţiale în sistemul bazei de date. În plus, este dificil să se aplice direct aceste secvenţe pentru detectarea tranzacţiilor rău intenţionate. Prin analiza atentă problemele întâmpinate, am descoperit că printr-un algoritm de generare de reguli, algoritmul modelului secvenţial poate fi utilizat să genereze clasificarea regulilor dorite pentru scopul propriu.

Problema descoperirii dependenţelor de date este împărţită în trei paşi: faza descoperirii modelului secvenţial,faza generării secvenţelor setate de citire şi scriere şi faza de generare a regulilor de dependenţă a datelor.

3.2.1 Faza descoperirii modelului secvenţial

Se consideră 10 exemple de tranzacţii ca in tabelul 1. R(x) si w(x) reprezintă operaţii de citire şi scriere fără pierderea generalităţii, întregii sunt folosiţi pentru a reprezenta fiecare element de date din bază.Cu un suport minim setat la 25%, de exemplu un suport minim de 3 tranzacţii.

Tabel 1. Exemplu de tranzactii pentru modelul secvential

trans. ID	Operatiile tranzactiei
1	r(7),r(1),r(6),w(5),r(1),w(4);
2	r(1),r(5),w(1),r(4),r(5),w(4);
3	r(7),r(6),r(2),w(4),r(7),r(3),w(6),r(1),r(6),w(2), r(3),r(5),r(2),w(5);
4	r(2),w(2),r(4),r(7),w(3), r(6),w(5), r(1), w(4);
5	r(5),r(3),r(6),w(7);
6	r(6),r(1),w(3),r(1),w(6),r(2),r(7),r(4),w(2);
7	r(2),r(5),w(6);
8	r(4),w(6);
9	r(6),r(5),w(5),r(3),r(4),w(4),r(3),w(7);
10	r(5),r(6),w(5),r(5),w(4);

Tabelul 2 ilustrează 13 modele secvenţiale dorite care satisfac suportul impus.De exemplu ,modelul secvenţial <r(6), w(5), w(4)> este suportat de tranzacţiile 1,4,9 si 10.Un exemplu de secvenţă care nu satisface suportul minim este secvenţa <r(2), w(2), r(7)> care este suportată doar de tranzacţia 4. Unele secvenţe ,de exemplu <r(1)> şi <r(7),r(6)> nu sunt in setul de răspuns deoarece nu sunt maxime cu toate că au suport minim.

Pentru a descoperi modelul secvenţial din log-urile bazei de date,putem folosi algoritmii de model secvenţial :AprioriAll,PrefixSpan si GSP. În mediul de testare folosim AprioriALL pentru generarea modelului secvenţial în aceasta fază cu toate că performanţa acestuia nu este la fel de bună ca PrefixSPan şi GSP. Această fază preliminară oferă baza in descoperirea dependenţelor datelor printre elementele de date.

Model secvential cu suport>25%	Suport
r(3)	30%
w(6)	40%
r(1), w(4)	30%
r(2), w(2)	30%
r(2), r(7)	30%
r(4), w(4)	30%
r(5), w(4)	30%
r(5), w(5)	30%
r(6), r(5)	30%
r(6), w(5), w(4)	40%
r(7), r(6), r(1)	30%
r(7), r(6), w(4)	30%
r(7), r(6), w(5)	30%

Tabel 2. Modelul Secvential

3.2.2 Faza generării setului de secvenţe de citire şi scriere

Observand modelul secvenţial din tabelul 2,este clar că anumite modele pot fi folosite pentru descoperirea de dependenţe între date în timp ce alte modele nu ar trebui luate în considerare.

În primul rand, unele modele secvenţiale conţin doar o operaţie.De exemplu, modelul secvenţial <r(3)> sau <w(6)> conţin operaţii doar asupra uneia dintre date, aşa că nici o dependenţă de date nu poate fi generată şi drept urmare nu ar trebui luate în considerare.

În al doilea rând unele modele secvenţiale conţin doar operaţii de citire. Din moment ce suntem preocupaţi în special de modificări dăunătoare ale elementelor de date, realizate de tranzacţiile unui utilizator, vom acorda atenţie doar tranzacţiilor care conţin operaţii de scriere. Astfel modelele care conţin doar operaţii de citire vor fi de asemenea neglijate.

Pentru toate celelalte modele secvenţiale, următoarele proceduri sunt angajate pentru a genera seturile de secvenţe de citire şi scriere.Pentru fiecare operaţie de scriere w(di) în modelul secvenţial, adăugăm <r(di1), r(di2), r(di3),…, r(din), w(di)> la setul de secvenţe de citire a datelor di unde {r(di1), r(di2), r(di3),…, r(din)} este setul tuturor operaţiilor de citire înainte de w(di). În mod similar ,adăugăm <w(di), w(dj1), w(dj2), w(dj3),…, w(djk)> la setul de secvenţe de scriere a datelor di unde {w(dj1), w(dj2), w(dj3),…, w(djk)} este setul tuturor operaţiilor de scriere după w(di).

Tabelul 3 ilustrează seturile de secvenţe scriere şi citire generate utilizând metoda de deasupra, din modelul secvenţial din tabelul 2. De exemplu secvenţa <r(6),w(4)> denotă ca înainte ca elementul de date 4 să fie actualizat, elementul de date 6 ar trebui citit. În timp ce secvenţa <r(7),r(6),w(4)> reprezintă faptul că înainte ca elementul de date 4 să fie actualizat , elementele de date 7 şi 6 ar trebui citite în secvenţă. Din aceste două secvenţe, cea care reprezintă o dependentă mai exactă poate fi determinată prin analiza rweight(4,{6}) şi rweight(4,{7,6}) şi aceasta va fi ilustrată în următorul subcapitol.În setul de secvenţe de scriere există doar un element <w(5),w(4)> care denotă că după ce elementul de date 5 este actualizat, elementul de date 4 ar trebui actualizat.

Tabel 3. Seturi de Secvente Citire si Scriere

Setul de Secvente Citire	Setul de Secvente Scriere
r(1), w(4)	w(5), w(4)
r(2), w(2)	
r(4), w(4)	
r(5), w(4)	
r(5), w(5)	
r(6), w(5)	
r(6), w(4)	
r(7), r(6), w(4)	
r(7), r(6), w(5)	

3.2.3 Faza de generare a regulilor de dependenţă a datelor

Regulile de dependenţă a datelor sunt categorizate ca reguli de citire şi reguli de scriere. Următoarea procedură este utilizată pentru a genera reguli de dependenţă a datelor. Pentru toate modelele secvenţiale <r(di1), r(di2), …, r(din), w(di) > în setul de reguli de citire, generează reguli de citire cu formatul w(di)→r(di1), r(di2), …, r(din). Dacă securitatea regulii este mai mare decat securitatea minimă, atunci este adăugată la setul de răspunsuri ale regulilor de citire care descrie ca înainte de a actualiza di, elementele de date di1,di2....din ,trebuie citire din aceeaşi tranzacţie.

Tabel 4. Reguli generate de dependenta de date

Regulile dependentelor de date	Siguranta
w(2)→r(2)	100%
w(5)→r(6)	100%
w(4)→r(6)	83%
w(5)→w(4)	80%

În mod similar pentru toate modelele secvenţiale w(di),w(dj1),w(dj2),...w(djk) în setul de secvenţe de scriere, generează reguli de scriere cu formatul w(di)→w(dj1), w(dj2), …, w(djk). Dacă securitatea regulii este mai mare decat securitatea minimă, atunci este adăugată la setul de răspunsuri ale regulilor de scriere care descrie ca după actualizarea di, elementele de date dj1,dj2....djk trebuie actualizate de aceeaşi tranzacţie.

De exemplu, atât <r(6),w(4)> cât şi <r(7),r(6),w(4)> aparţin setului de secvenţe ale elementului de date 4. Aşa că două reguli de citire: A: w(4) → r(6) şi B: w(4) → r(7), r(6) pot fi generate. Dar nu reflectă cât de puternică este corelarea datelor intre elementul de date 4 şi {6} sau între 4 şi {7,6} cu toate că ambele satisfac suportul minim specificat. Presupunem că securitatea minimă este setată la 70%.Datorită securităţii regulii B este 50%, de exemplu rweight(4, {7, 6}) = 50%, nu se găseşte în setul de reguli. În timp ce securitatea regulii A este de 83% ,rweight(4, {6}) = 83%, este selectată. Regulile de scriere şi citire generate din setul de secvenţe in Tabelul 3 sunt ilustrate în tabelul 4.

Aceste reguli funcţionează ca şi reguli de clasificare pentru a identifica tranzacţii rău intenţionate la sistemul de baze de date.

După generarea regulilor de citire şi scriere, acestea pot fi utilizate pentru a detecta tranzacţii dăunătoare prin verificarea jurnalului sau a logurilor bazei de date. Tranzacţiile care au făcut modificări in bază fără a urma regulile de dependenţă a datelor sunt marcate ca tranzacţii dăunătoare bazei.

Procedura este după cum urmează. Pentru toate tranzactiile bazei de date care au operatii de scriere , verifică dacă fiecare operaţie urmează regulile de dependenţă a datelor, de

exemplu dacă această tranzacţie a citit elementele de date corespunzătoare înainte de actualizare şi dacă a scris elementele de date corespunzătoare dupa actualizarea respectivă.

Iată un exemplu mai clar de a ilustra procedura de detectare.

Presupunem că avem tranzacţiile T1: r(2), r(5), w(2), r(6), r(1), w(7), r(4), r(3), r(5), w(5). In T1, elementele 2,7 şi 5 sunt actualizate.

Pentru elementul 2 regula w(2)→r(2) este satisfăcută deoarece înainte ca elementul 2 să fie actualizat , însuşi elementul 2 este citit de tranzacţie.

Pentru elementul 7, deoarece nu există nici o dependenţă de date pentru acesta , nu este nevoie să îl verificăm.

Pentru elementul 5,regula w(5)→r(6) este satisfăcută dar regula w(5)→w(4) nu este deoarece după ce elementul 5 este actualizat nici un alt element de date nu mai este actualizat de aceeaşi tranzacţie. Deci tranzacţia T1 este identificată ca şi tranzacţie dăunătoare .

4. Algoritm

Algoritmul formal pentru determinarea regulilor de dependenţă a datelor este prezentat după cum urmează:

Algoritm:

1. Iniţializarea setului de secvenţe de citire RS={} şi setului de secvenţe de scriere WS={};
2. Iniţializarea setului de reguli de citire RR={} şi setului de reguli de scriere WR={};
3. Generarea modelului secvenţial X={xi| suport(xi)>suportul minim} prin utilizarea algoritmului modelului secvenţial existent;
4. Pentru fiecare model secvenţial xi unde |xi|>1
   Dacă există o operaţie de scriere in el
   Pentru fiecare operaţie scrisă wi ∈ xi
   Dacă <$r(d_{i1})$, $r(d_{i2})$, $r(d_{i3})$,…, $r(d_{in})$, $w(d_i)$>∉ RS şi
   < $r(d_{i1})$, $r(d_{i2})$, $r(d_{i3})$,…, $r(d_{in})$> ≠ <∅>
   adaugă <$r(d_{i1})$, $r(d_{i2})$, $r(d_{i3})$,…, $r(d_{in})$, $w(d_i)$>
   la RS unde $r(d_{i1})$, $r(d_{i2})$, $r(d_{i3})$,…, $r(d_{in})$
   toate sunt operaţii de citire înainte de w(di)
   Dacă <w(di), w(dj1), w(dj2), w(dj3),…, w(djk)> ∉ WS şi
   <wj1, wj2, wj3,…, wjk> ≠ <∅>
   adaugă <w(di), w(dj1), w(dj2), w(dj3),…, w(djk)>
   la WS unde w(dj1), w(dj2), w(dj3),…, w(djk)
   toate sunt operaţii de scriere după wi
5. Pentru fiecare secvenţă din RS
   Dacă suportă (<r(di1), r(di2), r(di3),…, r(din), w(di)>)
   /suportă(<wi(di)>) > securitatea minimă
   adaugă w(di)→r(di1), r(di2), …, r(din) la RR
   Pentru fiecare secvenţă din WS
   Dacă suportă (<w(di), w(dj1), w(dj2), …, w(djk)>)
   /suportă(<wi(di)>) > securitate minimă
   adaugă w(di)→w(dj1), w(dj2), …, w(djk) la WR

Pasii 1 şi 2 iniţializează seturile de secvenţă citire/scriere şi seturile de reguli citire/scriere.

Pasul 3 angajează algoritmul de modelare secvenţială existent pentru a genera modele secvenţiale care consistă într-o secvenţă de operaţii de citire sau scriere care satisfac suportul minim.

Pasul 4 generează seturile de secvenţe citeste sau scrie din modelul secvenţial. Modelul secvenţial care nu conţine nici o operaţie de scriere nu sunt luate in considerare în aceasta fază.Pentru fiecare operaţie de scriere în toate celelalte modele, secvenţa care constă în toate secvenţele de citire înainte de aceste operaţii de scriere şi această operaţie de scriere sunt adăugate în setul de secvenţe de citire. În mod similar pentru secvenţa ce conţine operaţiia de scriere însăşi si celelalte operaţii de scriere dupa ce aceste operaţii de scriere sunt adăugate setului de secvenţă de scriere.

Pasul 5 generează reguli de scriere şi citire de la seturile de secvenţe de scriere şi citire bazate pe securitate minimă.Toate regulile care satisfac securitatea minimă se găsesc in seturile de reguli de scriere sau citire ca ieşiri,rezultate ale algoritmului.

6. Analiza experimentală

Mai multe experimente au fost efectuate pentru testarea performanţei metodei propuse.

Două log-uride baze de date diferite au fost generate pentru testarea atât a ratei adevărate pozitive cat şi a ratei fals pozitive ale modelului nostru. Primul log a constat din tranzacţii de baza de date sintetice, care au fost tratate ca tranzacţii dăunătoare. Aceste tranzacţii dăunătoare au fost generate aleator bazat pe ipoteza că atacatorul ar putea să aibe cunoştinţe de dependenţele de date din baza de date. Al doilea log sau jurnal al bazei de date a constat în tranzacţii normale ale unui utilizator.

Tabelul 5 ilustrează bază de stabilire a experimentului nostru. În scopul de a testa măsura în care abordarea noastră este sensibilă la dependenţele de date, tranzacţiile rău intenţionate şi de user normal, au fost generate bazat pe parametrii patru şi cinci din tabelul 5, de exemplu, numărul mediu de operaţiuni de citire imediat înainte de operaţiunea de scriere din tranzacţii şi numărul mediu de scriere în tranzacţii. Apoi, prin varierea unui parametru la un moment dat, am evaluat modul în care sistemul de detectare a intruziunii a răspuns la schimbarea parametrilor de performanţă şi dacă au fost sensibili la această schimbare.

Tabel 5. Situaţia de referinţă a experimentului

Suport	Siguranţa	# operaţii citite	# operaţii scrise	# tranzacţii
.15	.75	2	2	2000

Figura 2 prezintă ratele adevărate pozitive în detectarea tranzacţiilor dăunătoare. Jurnalul de baze de date cu tranzacţiile dăunătoare a fost folosit pentru acest experiment. Figura 2 arată că ratele adevărate pozitive cresc constant atunci când rata de dependenţă de date este mai puternică printre elementele de date. Este de remarcat faptul că atunci când numărul mediu de operatii de scriere variază de la 1 la 5, creste rata de adevărat pozitiv de la 41% la 91%. Întrucât, în cazul în care numărul mediu de operaţii de citire imediat înainte de o operaţie de scriere în tranzacţii variază de la 1 la 5, rata adevărat pozitiv creşte de la 65% la 86%. Comparând cele două grafice din figura 2, se observă că rata de detecţie adevărat pozitiv este mult mai sensibilă la numărul mediu de scriere intr-o tranzactie. Aceasta inseamnă că, în cazul în care există mai multe declaraţii actualizare în tranzacţii, rata de detecţie va creşte rapid.

Figura 2. Rate adevarat pozitive in detectarea tranzactiilor daunatoare

Figura 3 ilustrează ratele fals pozitive la testele de tranzacţii normale de utilizare. Jurnalul de baze de date cu tranzacţii normale de utilizare a fost folosit în aceste cazuri. Este de observat că rata fals pozitiv poate fi la fel de scăzuta ca 12,5% atunci când dependenţa de date nu este foarte puternică. Rata maximă de fals pozitiv este de 29% în cazul în care numărul mediu de operaţii de citire imediat înainte de o operaţie de scriere în tranzacţii este de 5. Comparativ cu Figura 2, se observă că odată cu creşterea de dependenţă de date, creşterea ratei adevărat pozitiv este mult mai mare decât cea a ratei de fals pozitiv.

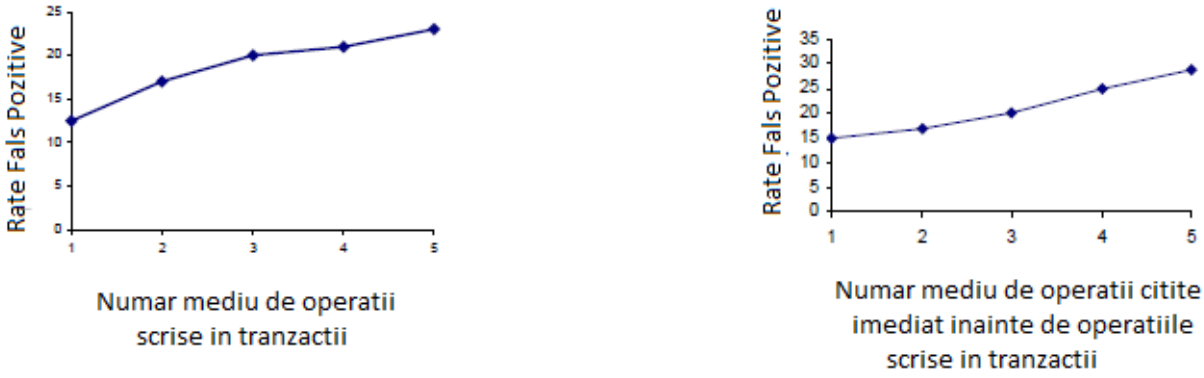

Figura 3. Ratele Fals Pozitive in Testarea tranzactiilor unui Utilizator Normal

7. Concluzii

În această lucrare am propus o abordare de data mining pentru a detecta tranzacţiile dăunătoare în sistemele de baze de date. Abordarea noastră se concentrează pe dependenţe data mining printre elementele de date din baza de date. Regulile dependenţelor de date

descoperite de dependentele de date miner sunt folosite ca regulile de clasificare pentru identificarea anomaliilor.

Experimentul pe tranzacţiile de baze de date sintetice ilustrează faptul că metoda propusă funcţionează în mod eficient pentru a detecta tranzacţiile dăunătoare în sistemele de baze de date furnizand anumite dependenţe de date existente.

Rezultatul arată în continuare că cu cât o dependenţă de date intre elementele de date este mai puternică cu atât este mai mare performanţa.

Bibliografie:

A Data Mining Approach for Database Intrusion Detection , Y. Hu, B. Panda, Computer Science and Computer Engineering Department, University of Arkansas.

(Text realizat de Beatrice Popa, grupa ABD master, ianuarie 2011, articol p711- hu, pentru cursul”Sisteme avansate de baze de date”).