Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
—
Институт прикладной математики и механики
Кафедра «Информационная безопасность компьютерных систем»

Л а б о р а т о р н а я р а б о т а № 2

«Модель Китайская стена»

по дисциплине «Модели безопасности компьютерных систем»

Выполнил
студент гр. 33656/1 Крамсаков Е.Ю.

<*подпись*>

Руководитель,
преподаватель <*подпись*> Овасапян Т.Д.

Санкт-Петербург
2019

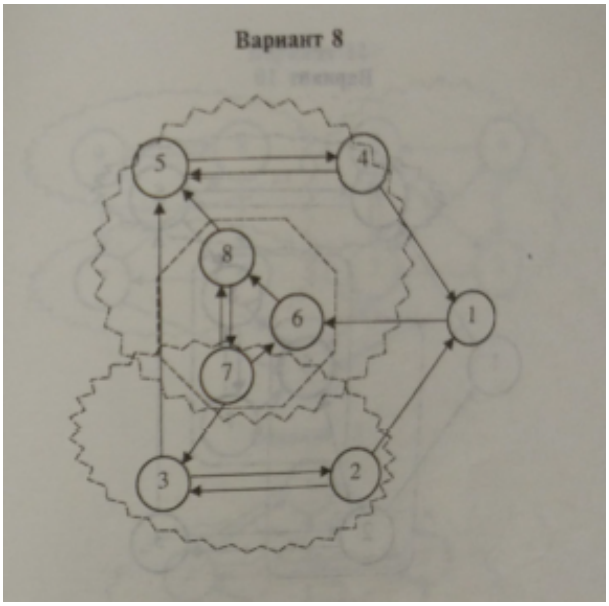
Цель работы

Исследование возможностей и защитных характеристик решетчатых моделей на примере модели Китайская стена.

Задачи работы

1) На языке программирования Пролог описать решетчатую модель Китайская стена.
2) Убедиться в защите, предоставляемой моделью Китайская стена в рамках одного и нескольких классов конфликтов интересов.

Выполнение работы



На языке программирования Prolog была создана система, удовлетворяющая заданию варианта 8.

В качестве субъектов используются вершины приведенного графа, остаток от деления номера которых на три равен 1 (1, 4, 7). Остальные вершины — объекты. Для отсутствия связей объект-объект граф связей был немного изменен.

Кроме того, были введены понятия классов конфликтов интересов: банк, it-компания, ресурсодобывающая компания. В каждом классе интересов определены несколько компаний, которые просто пронумерованы начиная с 1.

Таблица соответствия объектов, субъектов и компаний из классов интересов на начало работы модели:

s1	s4	s7	o2	o3	o5	o6	o8	o9
bank 1			bank 1	bank 2	bank 3, it 1	resources 1	bank 4, resources 2	

Был реализован доступ read и write от объектов к субъектам в соответствии с уровнями доступа и правилами NRU и NWD. То есть, субъект, которому присвоен один или несколько классов интересов, он сможет читать только из тех объектов, классов интересов которых еще нет в списке классов интересов субъектов, или, если есть, то именно этот объект и является интересом субъекта. При этом при чтении субъект помечается соответствующей компанией и классом интересов. Писать субъект может только в те объекты, которые не противоречат его меткам и не являются общедоступными.

```
?- try_access(s1, o2, r).
Reading done
true .
?- print_matrix.
o2: bank #1
o3: bank #2
o5: bank #3
o5: it #1
o6: resources #1
o8: bank #4
o8: resources #2
s1: bank #1
true.
?- try_access(s1, o3, r).
conflict of interest: s1 have bank #1 as well as o3 have bank #2
true .
?- try_access(s1, o5, r).
conflict of interest: s1 have bank #1 as well as o5 have bank #3
true .
?- try_access(s1, o6, r).
Reading done
true .
?- print_matrix.
o2: bank #1
o3: bank #2
o5: bank #3
o5: it #1
o6: resources #1
o8: bank #4
o8: resources #2
s1: bank #1
s1: resources #1
true.
?- try_access(s1, o8, r).
conflict of interest: s1 have bank #1 as well as o8 have bank #4
true .
?- try_access(s4, o5, r).
Reading done
true .
?- print_matrix.
o2: bank #1
o3: bank #2
o5: bank #3
o5: it #1
o6: resources #1
o8: bank #4
o8: resources #2
s1: bank #1
s1: resources #1
s4: bank #3
s4: it #1
true.
```

Тестирование с несколькими классами конфликтов интересов:

```
?- try_access(s7, o5, r).
Reading done
true .

?- print_matrix.
o2: bank #1
o3: bank #2
o5: bank #3
o5: it #1
o6: resources #1
o8: bank #4
o8: resources #2
s1: bank #1
s7: bank #3
s7: it #1
true.

?- try_access(s7, o5, r).
Reading done
true .

?- try_access(s7, o3, r).
conflict of interest: s7 have bank #3 as well as o3 have bank #2
conflict of interest: s7 have bank #3 as well as o3 have bank #2
true .

?- try_access(s7, o3, w).
Can not write to object with same interest, but other company
Can not write to object with same interest, but other company
true .
```

Результат работы

Была создана программа на языке программирования Пролог, которая представляет собой реализацию модели Китайская стена. В данной программе была исследована защищенность моделируемой системы.

Контрольные вопросы

Какая математическая структура положена в основу модели «Китайская стена»? Каков ее характеристики?

В основу модели Китайская стена положена математическая решетка. Решетки — структур, содержащие конечное множество частично упорядоченных элементов с определенными операторами, задающими наибольшую нижнюю и наименьшую верхнюю границу.

Какие проблемы моделей мандатного и дискреционного доступа решаются с помощью решеточных моделей?

Проблемы утечки информации в потоках информации от субъектов к объектам и наоборот. Решетчатые модели следят за безопасностью потоков данных, а не сущностей.

С какой целью введена метка SYSHIGH?

Это метка, помеченная всеми компаниями во всех классах конфликтов интересов. Она представляет собой наименьшую верхнюю границу, так как в понятиях модели Китайская стена невозможно задать ее явно.

Каковы условия доминирования между метками l_1 и l_2?

Если $l[i_k]$ — i_k-й элемент метки l, то $l_1 \geq l_2$, если $l_1[i_k] = l_2[i_k] \vee l_2[i_k] = \perp$ для $\forall k = 1,2,\ldots,n$

Сколько уровней решетки будет в системе с пятью ККИ?

Шесть.

Итоги работы

В итоге работы была изучена решеточные модели на примере модели Китайская стена. Были изучены особенности данных моделей.

Выводы

Решеточные системы являются достаточно специфичными, так как следят исключительно за потоками информации. Я считаю, что их необходимо использовать совместно с дискреционными и/или мандатными моделями доступа.

Приложение

Исходный код программы:

```
:-dynamic subject/1.
:-dynamic object/1.
:-dynamic interest/3.
:-dynamic writing_flag/1.
:-dynamic not_reading_flag/1.
:-dynamic not_writing_flag/2.

/*субъекты -- если остаток от деления на 3 у числа равен 1*/

object(O):-subject(O).

object(o2).
object(o3).
object(o5).
object(o6).
object(o8).
object(o9).

subject(s1).
subject(s4).
subject(s7).

right(r).
right(w).
```

```
interest(o2, bank, 1).
interest(o3, bank, 2).
interest(o5, bank, 3).
interest(o5, it, 1).
interest(o6, resources, 1).
interest(o8, bank, 4).
interest(o8, resources, 2).


%%DEBUG

interest(s1, bank, 1).

%%DEBUG END


try_access(S, O, R):-


        not(subject(S)), write("there is no subject "), write(S);
        not(object(O)), write("there is no subject/object "), write(O);



        ( %% если нет вообще пересечения интересов
                interest(S, _, _),
                (
                        forall(interest(S, _si, _sn), %% для всех компаний (_sn) во всех сферах
интересов (_si) субъекта S
                                (
                                        interest(O, _, _),
                                        (
                                                forall(interest(O, _oi, _on), %%  взять все компании
(_on) во всех сферах интересов (_oi) объекта O
                                                        (
                                                                (
                                                                        _si == _oi, not(_sn == _on),

                                                                                (

                                                                                        (
                                                                                                R == r,

write("conflict of interest: "),

                                                                                                write(S),

write(" have "), write(_si), write(" #"), write(_sn), write(" as well as "),
                                                                                                write(O),

write(" have "), write(_oi), write(" #"), write(_on), nl, assert(not_reading_flag(t))
                                                                                        );
                                                                                        (
                                                                                                R == w,
                                                                                                write("Can
not write to object with same interest, but other company"), nl,

assert(not_writing_flag(S, O)), writing_flag(t), retract(writing_flag(t))
                                                                                        ); true
                                                                                )
                                                                );

                                                                (
                                                                        (R == w, not(not_writing_flag(S,
O)), assert(writing_flag(t))), false
                                                                )
                                                        )
                                                )
                                        )

                                )
                        )
                )
        );

        (
                writing_flag(t),
                        (
                                write("Writing done"), retract(writing_flag(t))
                        )
        );

        (
```

```
                not(not_reading_flag(t)),
                (
                        R == r,
                        (
                                forall(interest(O, _oi, _on),
                                        (
                                                (not(interest(S, _oi, _on)), assert(interest(S, _oi,
_on))); true
                                        )
                                )
                        ),
                        write("Reading done")
                );
                retract(not_reading_flag(t))
        );

        R == w, not(interest(O, _, _)), write("Can not write to public objects").



print_matrix:-
        forall(interest(_o, _i, _n),
                (
                        write(_o), write(": "), write(_i), write(" #"), write(_n), nl
                )
        ), true.
```